MONTE CARLO SIMULATION STUDIES OF THE ISING MODEL

Project Report

by

SATHEESH D M (PH22C043)

Mentored by

PROF. BIRABAR RANJIT KUMAR NANDA

DEPARTMENT OF PHYSICS
INDIAN INSTITUTE OF TECHNOLOGY MADRAS
JAN 2023

ACKNOWLEDGEMENT

I would like to express my gratitude to Dr. Birabar Ranjit Kumar Nanda for suggesting this topic to explore. His guidance and discussions were of great help and I appreciate the time he spent with me amidst his tight schedule.

I would like to thank Mr. Jatin Kumar Bidika for his help through out this project. Atlast, I would like to expand my thanks to everyone who directly or indirectly helped me in completing this project.

Satheesh D M (PH22C043),
IIT - Madras,
Chennai - 600036.

ABSTRACT

This project work is done to understand the importance of Monte Calro simulation technique used in condensed matter systems. Ising model systems are studied in this work. Ising model with nearest neighbour interactions are simulated for a 1D chain of 100 lattice points and 2D (10 x 10) lattice. A second order phase transition is observed in the 2D model simulations. Then the simulation results are compared with the analytical solutions available. The effect of finite lattice size on the simulation is observed in this comparison. Further, non analytic systems like 3D (10 x 10 x10) model and 2D model with next nearest neighbour interactions are simulated. Their critical points are identified by calculating the Binder's cumulant.

This report is divided into five chapters. The first chapter is the introduction to Ising model. The second chapter covers the Metropolis algorithm. The third chapter includes the simulation results of the nearest neighbour models. 2D next nearest neighbour model results and the Binder's cumulant plots are included in the fourth chapter. The last chapter is the summary and suggestions for future work.

LIST OF FIGURES

TABLE OF CONTENTS

1 INTRODUCTION

The motivation for this work is that not every physical system is analytically solvable. This does not mean that the system is violating the underlying physical law. The issue is mathematical, given the initial conditions of the system and the constraints imposed on the system, the set of equations describing the systems do not have any solution in closed form (no analytic solution). But the system still follows the equations that are written from the physical laws. This happens to be the case in many problems in statistical physics.

In statistical mechanics the system studied has many particles in them ($\approx 10^{23}$), and these particles interacts and undergoes a complex dynamics when studied in detail. We can not and also we do not need to keep track of every particle in such systems (microstate). Knowing the average quantities is just enough for most practical purposes (macrostate). Even though the equilibrium statistical mechanics is a well established field, simple toy models like the NN Ising model in higher dimensions (3D), etc,. do not have analytic solutions. In such cases the only way to analyse and predict the solution is via simulations. The availability of computers become helpful. Algorithms are the simple step by step rules which can be used to simulate a problem in the computer. Even computer simulation takes more time to solve some problems, when only the brute force of the computer is used in simulating the system. In such cases, algorithms that are optimised for fast and efficient computing is put into use.

In this work we simulated and studied various Ising models using the **Markov Chain Monte Carlo** simulation technique, this is executed as a python code by using the famous **Metropolis algorithm.** The quantities like average energy per spin, magnetization per spin, specific heat and susceptibility are computed and the phase transition in 2D and 3D models are studied in detail. The necessary concepts from the statistical physics and the Ising model is discussed in the following sections of this chapter.

1.1 Concepts From Statistical Mechanics

Canonical Ensemble: A statistical ensemble which is in contact with the thermal reservoir maintained at the temperature T. The number of particles N and the volume which encloses them are kept constant. The macrostate of this system is represented as (N,V,T). The system is allowed to exchange the energy with the reservoir to maintain the thermal equilibrium. [6]

The probability of finding our system in one of the microstate s is given by

$$P(s) = \frac{e^{-\beta H(s)}}{Z(N,V,T)} \tag{1.1}$$

where, β=$\frac{1}{K_B T}$, K_B being Boltzmann constant and H(s) is the Hamiltonian of the microstate.

Z(N,V,T) is the **Canonical partition function**. It is a vital quantity which links the microscopic details with macroscopic quantities. The partition function normalizes the probability distribution P(s), hence it is given by

$$Z(N,V,T) = \sum_s e^{-\beta H(s)} \tag{1.2}$$

where is summation is through all possible microstates s.

The thermodynamic quantities are related to partition function Z, like average energy($< E >$), Helmholtz function (F)

$$< E > = -\frac{\partial}{\partial \beta} ln(Z) \tag{1.3}$$

$$F = -\frac{1}{\beta} ln(Z) \tag{1.4}$$

From the Helmholtz function F, quantities of interest such as Magnetization (M), Specific heat capacity(C) can be obtained,

$$M = -\frac{\partial F}{\partial H} \tag{1.5}$$

where H is the external magnetic field

$$C = -T\frac{\partial^2 F}{\partial T^2} \tag{1.6}$$

In the above equations, Boltzmann constant k_B is set to 1.

1.2 The General Ising Hamiltonian

The Ising Model is the one of the simplest but most important models in statistical mechanics that is used to study phase transitions. The Hamiltonian of this Ising model has two types of terms as of our considerations, **Internal interaction term**. This term accounts for the contributions from a spin interacting with its neighbouring spins within the lattice. Based on model there are many terms like Nearest neighbour(NN), Next Nearest Neighbour (NNN), etc.

$$E_{int} = -\sum_{i,j} J_{ij} S_i S_j \tag{1.7}$$

Where, S_i is randomly chosen spin, S_j is other interacting spin and J_{ij} represents the interaction strength between spins $S_i$and S_j. **External interaction term**. This term accounts for the system's interaction with the external magnetic field

$$E_{ext} = -\sum_{i} H_i S_i \tag{1.8}$$

Here, H_i is the magnetic field strength. In both energy terms, minus sign is used for conventional reasons. Hamiltonian of the general Ising model is the sum of the above mentioned two terms,

$$\boxed{\mathcal{H}(S_i) = -\sum_{i,j} J_{ij} S_i S_j - \sum_{i}^{N} H_i S_i} \tag{1.9}$$

2 MONTE CARLO SIMULATION

In this section the concept and algorithm required to perform the simulation used in this project work are discussed briefly.

2.1 Markov Chain Monte Carlo

This is also known as MCMC method. It is a random sampling technique. A Markov chain is a sequence of points or states of the system which are correlated and the next state is chosen from the memory extending to one previous state. Othervise the next state in a Markov chain is achieved by following a distribution $\pi(x)$ and the selection of next state has a transition probability function as below.

$$P(x_i \rightarrow x_{i+1}) \equiv P(x_i|x_{i+1}) \tag{2.1}$$

The above mentioned correlation leads to ergodicity. Ergodicity means, the Markov chain sequence will have access to all possible states in the state space as the time evolves. Otherwise, all the states in the state space are accessible for the Markov chain sequence. This is shown below as an equilibrium. It is called the detailed balance equation.

$$\boxed{\pi(x_1)P(x_2|x_1) = \pi(x_2)P(x_1|x_2)} \tag{2.2}$$

Markov chains exhibit the so-called Markov property or memoryless property. Memoryless property in words can be put as: “The future depends on the past only through the present.”

Here in MCMC method we are interested in finding a stationary distribution $\pi(x)$ by starting from a similar initial distribution $\mu(x_0)$. For a Markov chain to converge to a stationary distribution it must have the following properties of **irreducibility** and **aperiodicity**.[5] The Markov chain is generated in the simulation by following the metropolis algorithm. [8]

2.2 Metropolis Algorithm for Ising model

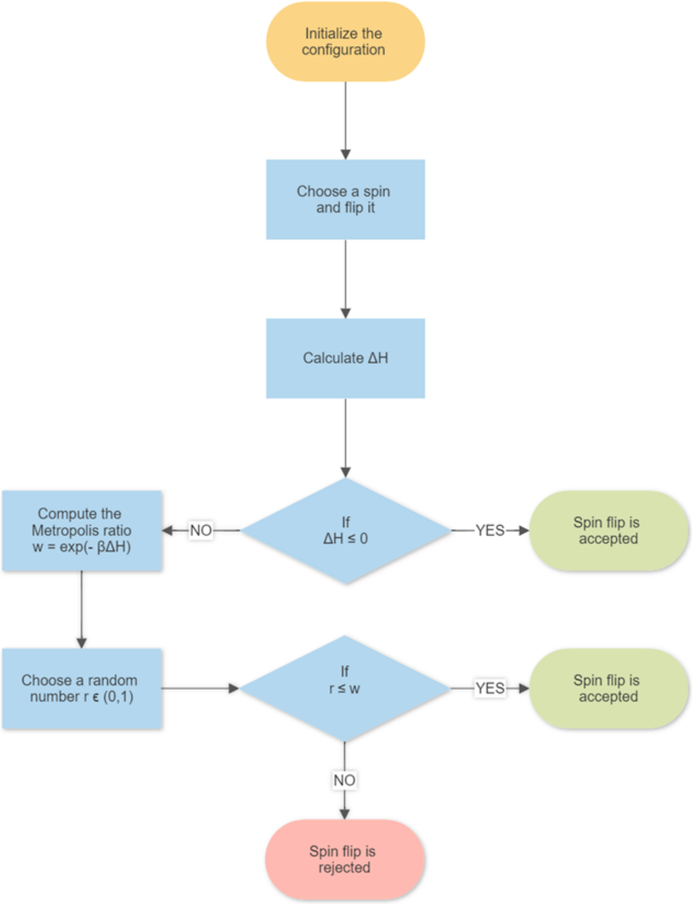
Initialize the configuration
Choose a spin and flip it
Calculate ΔH
If ΔH ≤ 0
YES
Spin flip is accepted
NO
Compute the Metropolis ratio w = exp(- βΔH)
Choose a random number r ϵ (0,1)
If r ≤ w
YES
Spin flip is accepted
NO
Spin flip is rejected

Figure 2.1: Metropolis Algorithm Flow Chart

The above flow chart shows the algorithm to flip a single spin in the initialized configuration. The above algorithm starting from the step of choosing a spin at random. Algorithm is repeated numerous times so that the Markov chain of states generated reaches the targeted stationary distribution. These steps are called the thermalization steps or the burn in steps. After this burn in steps the further steps simulated can be used for computing the quantities. A sample of the metropolis algorithm used in this project work is shown below [2].

```python
# Metropolis algorithm loop for updating configuration in 2D NN model:
def mc_loop(config, beta):
    for i in range(N):
        for j in range(N):
            # select a  spin
            spin =  config[i, j]
            # calculate delta_E due to spin flip
            nn = (config[(i+1)%N, j]\
                + config[(i-1)%N, j]\
                + config[i, (j+1)%N]\
                + config[i, (j-1)%N])    # nn = nearest neighbours
            delta_E = 2 * spin * nn
            r = rand()                   # random number generator
            # condition for acceptance of spin flip
            if (delta_E <= 0):   # accept
                spin *= -1
            elif (delta_E > 0 and r <= np.exp(-delta_E * beta)):# accept
                spin *= -1
            elif (delta_E >0 and r > np.exp(-delta_E * beta)):  # reject
                spin = spin
            # updating configuration
            config[i, j] = spin
    return config
```

2.3 Quantities Computed

These following quantities are derived according to formulae provided in section 1.1.

- **Energy per site**

$$< E >= \frac{-J}{2N}\sum_{ij} S_i S_j \tag{2.3}$$

It is divided by factor 2 to compensate for overcounting.

- **Magnetisation per site**

$$< M >= \frac{1}{N}\sum_i S_i \tag{2.4}$$

- **Specific heat capacity**

$$C = \frac{1}{T^2}(< E^2 > - < E >^2) \tag{2.5}$$

  where $< E^2 >$ indicates average over E^2.

- **Susceptibility**

$$\chi = \frac{1}{T}(< M^2 > - < M >^2) \tag{2.6}$$

  where $< M^2 >$ indicates average over M^2. Here, N is the total lattice points or total no. of spins in system considered.

3 NEAREST NEIGHBOUR MODELS

The simplest version of Ising model is the nearest neighbour interaction model. To further simplify the simulations, following assumptions are considered.

- The system is isotropic ($J_{ij} = J$)
- No external magnetic field ($H_i = H = 0$)
- Natural units is adopted ($k_B = 1$)

Hamiltonian of NN model

$$\boxed{\mathcal{H}(S_i) = -J\sum_{<ij>} S_iS_j} \tag{3.1}$$

here, $< ij >$ means nearest neighbour interactions.

3.1 1D Model - Analytical Solutions

1D model consists of a long chain of spins, in which a spin interacts only with two neighbouring spins. It can be solved analytically.

- **Energy per spin**

$$< E >= -Jtanh\left(\frac{J}{T}\right) \tag{3.2}$$

- **Magnetisation per spin**

$$< M >= tanh\left(\frac{H}{J}\right) \tag{3.3}$$

  since, H=0 in our case, $< M >$=0.

- **Specific heat capacity**

$$C = \left(\frac{J}{T}\right)^2 \frac{1}{cosh^2(J/T)} \tag{3.4}$$

- **Susceptibility**

$$\chi = \left(\frac{1}{T}\right) \frac{1 + tanh(J/T)}{1 - tanh(J/T)} \tag{3.5}$$

  We do not expect any phase transitions in 1D model.

3.2 1D model - Temperature Response

To study the behaviour of the system with temperature, interaction strength J is fixed to 1. The model is simulated using Markov Chain Monte Carlo technique. Lattice of 100 spins is taken and periodic boundary conditions are considered. 1000 Monte Carlo steps are taken for thermalization. The quantities are averaged over 1000 Monte Carlo steps after thermalization. 200 temperature points are considered in the range T$\in$ $[0, 5]$. It is clear from the following graphs, there is **no phase transition** observed and this is in accordance with the theory. The system is in **paramagnetic state** and this is observed in the $< M >$ Vs $< E >$ graph below, as zero and random $< M >$ values.

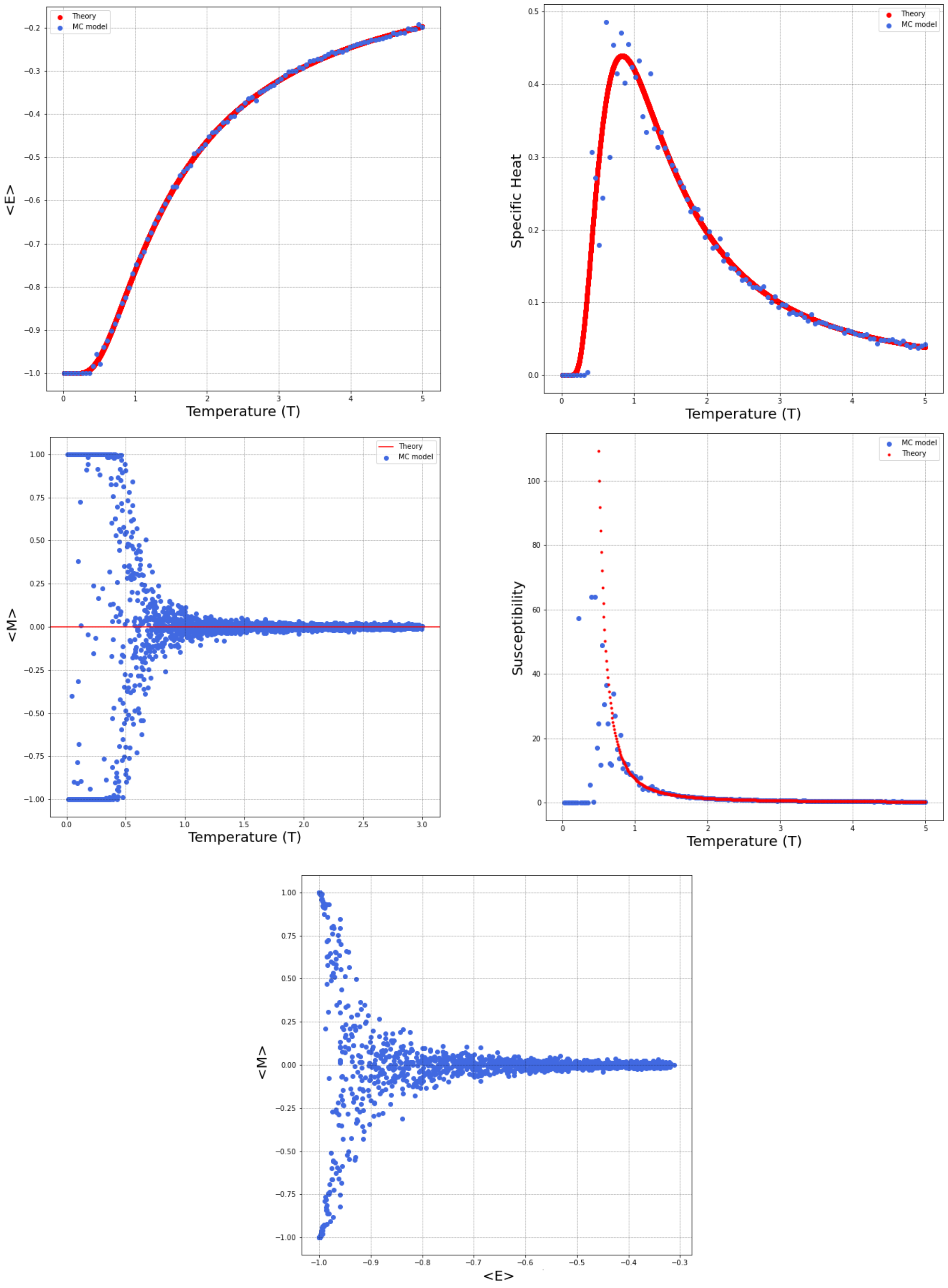

Figure 3.1: 1D NN Model - Temperature Response

3.3 1D model - Interaction Strength Response

To study the system's behaviour with the interaction strength J, temperature is fixed (T = 1K). Lattice of 100 spins is taken and periodic boundary is considered. 1000 Monte carlo steps are taken for thermalization. 200 data points are considered in the range, J $\in [-1, 1]$, the quantities computed are averages over 1000 MC time steps after thermalization. The simulation is in agreement with the theory (cf. Fig3.2). The system in the range J $\in$ [-1,0) is in **antiferromagnetic** phase, this is because, susceptibility is low and $< M >$ is almost zero with minor fluctuations indicating no

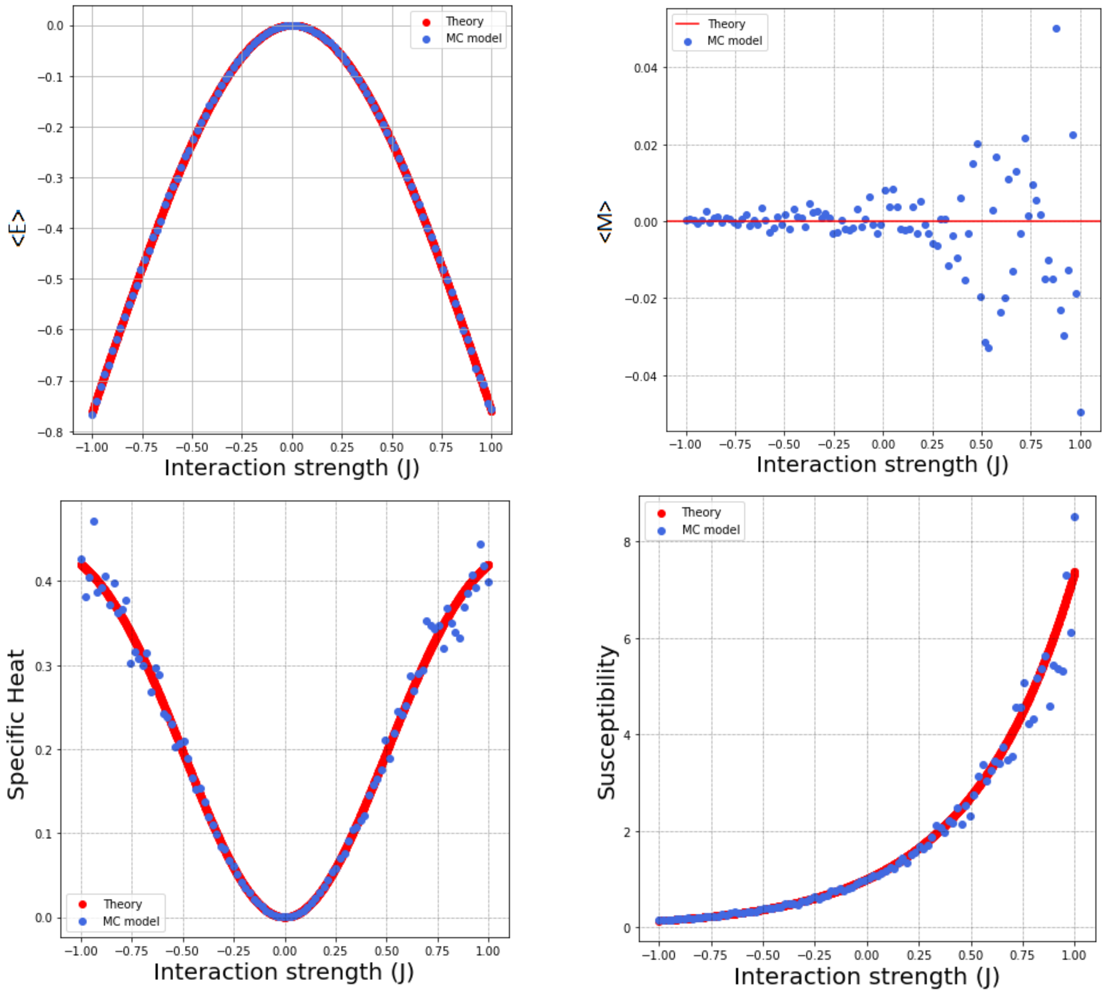

Figure 3.2: 1D NN Model - Interaction Strength Response

net spin. The system is in **paramagnetic** phase in the range J $\in$ (0,1], since the susceptibility is relatively greater and $< M >$ is fluctuating near zero indicating random orientation of spins. The Energy ($< E >$ Vs J) graph is symmetric about J = 0 which indicates that the system tends to be at lower energy state irrespective of the nature of interaction.

3.4 2D Model - Analytical solutions

This 2D Ising model models a square lattice of size (N x N). The solution for this model is given by Onsager in his paper in 1944. He used periodic boundary conditions to solve this model. The solutions mentioned below are adopted from the K. Huang's Statistical mechanics book.[4]

- **Energy per spin**

$$< E >= -J \coth\left(\frac{2J}{T}\right)\left[1 + \frac{2}{\pi}k^{'}K_1(k)\right] \tag{3.6}$$

- **Specific heat capacity**

$$C = \frac{2J^2}{\pi T^2}\coth^2\left(\frac{2J}{T}\right)\left\{2K_1(k) - 2E_1(k) - (1-k^{'})\left[\frac{\pi}{2} + k^{'}K_1(k)\right]\right\} \tag{3.7}$$

- **Magnetisation per spin**

$$< M >= \begin{cases} \pm\left\{1 - \left[\sinh\left(\frac{2J}{T}\right)\right]^{-4}\right\}^{\frac{1}{8}} & ;(T < T_c) \\ 0 & ;(T > T_c) \end{cases} \tag{3.8}$$

Here, the definitions considered are as below,

$$k \equiv \frac{2\sinh\left(\frac{2J}{T}\right)}{\cosh^2\left(\frac{2J}{T}\right)} \tag{3.9}$$

$$k^{'} \equiv 2\tanh^2\left(\frac{2J}{T}\right) - 1 \tag{3.10}$$

the complete elliptical integrals of first and second kind are shown below respectively,

$$K_1(k) \equiv \int_0^{\pi/2} \frac{d\phi}{\sqrt{1 - k^2 sin^2\phi}} \tag{3.11}$$

$$E_1(k) \equiv \int_0^{\pi/2} \sqrt{1 - k^2 sin^2\phi}\, d\phi \tag{3.12}$$

The critical point, it is reached when $k^{'} = 0$, from eqn. 3.10

$$2 \tanh^2 \left(\frac{2J}{T} \right) = 1 \tag{3.13}$$

for Temperature response, the above formula gives the T_c as a function of J,

$$\boxed{T_c = 2.269185J} \tag{3.14}$$

for Interaction sterngth response, the above formula gives the J_c as a function of T,

$$\boxed{J_c = 0.4406868T} \tag{3.15}$$

3.5 2D model - Temperature Response

Each spin in this model interacts with its 4 nearest neighbours. Periodic boundary conditions is assumed for the simulation. To study the behaviour of the system with temperature, J is fixed $(J = 1)$. A square lattice of size (50 x 50) is considered. 1000 MC steps are used for thermalization. 100 data points are considered in the range T ∈ [0, 5] and the quantities are averaged over 1000 MC steps. A **second order phase transition** from ferromagnetic to paramagnetic phase is observed as expected in the theory. The critical temperature in simulation ($T_c \approx$ 2.27) is also in agreement with eqn.3.14. The system is in **ferromagnetic** phase for T ∈ (0, 2.27) since, $< M >$ = ± 1 in this range. It is in **paramagnetic** phase for all T $> T_c$.

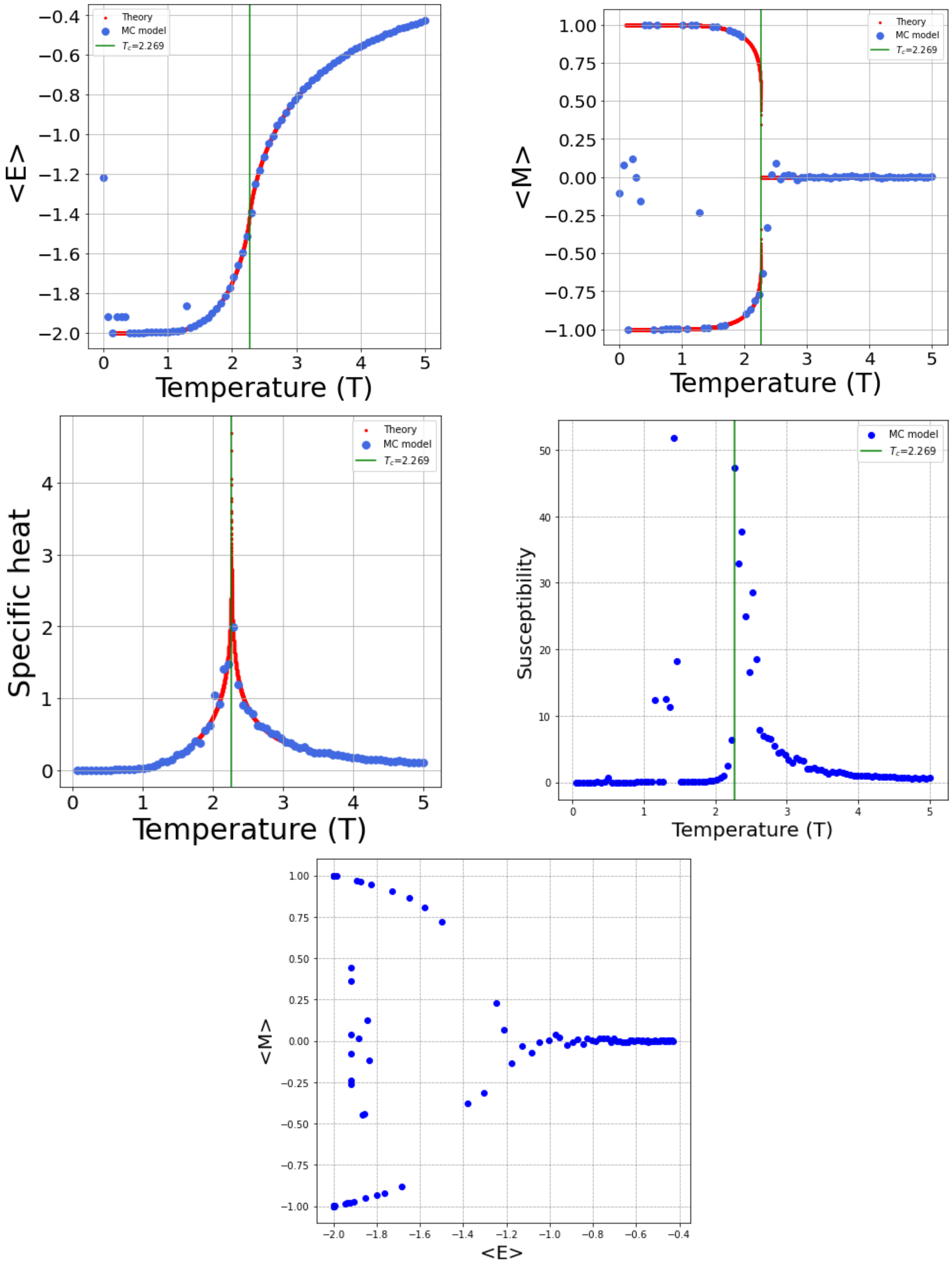

Figure 3.3: 2D NN Model - Temperature Response

3.6 2D Model - Interaction Strength Response

To study the interaction strength response the value of temperature is fixed. A (10 x 10) lattice is considered. 1000 Monte Carlo steps are used for thermalization and then another 1000 steps is used for computing the quantities. Since 2D model shows phase transition this model is simulated for three different temperature values (T $< T_c$, T $\approx T_c$ and $T > T_c$). The simulations are in agreement with the theory for all the temperature values.

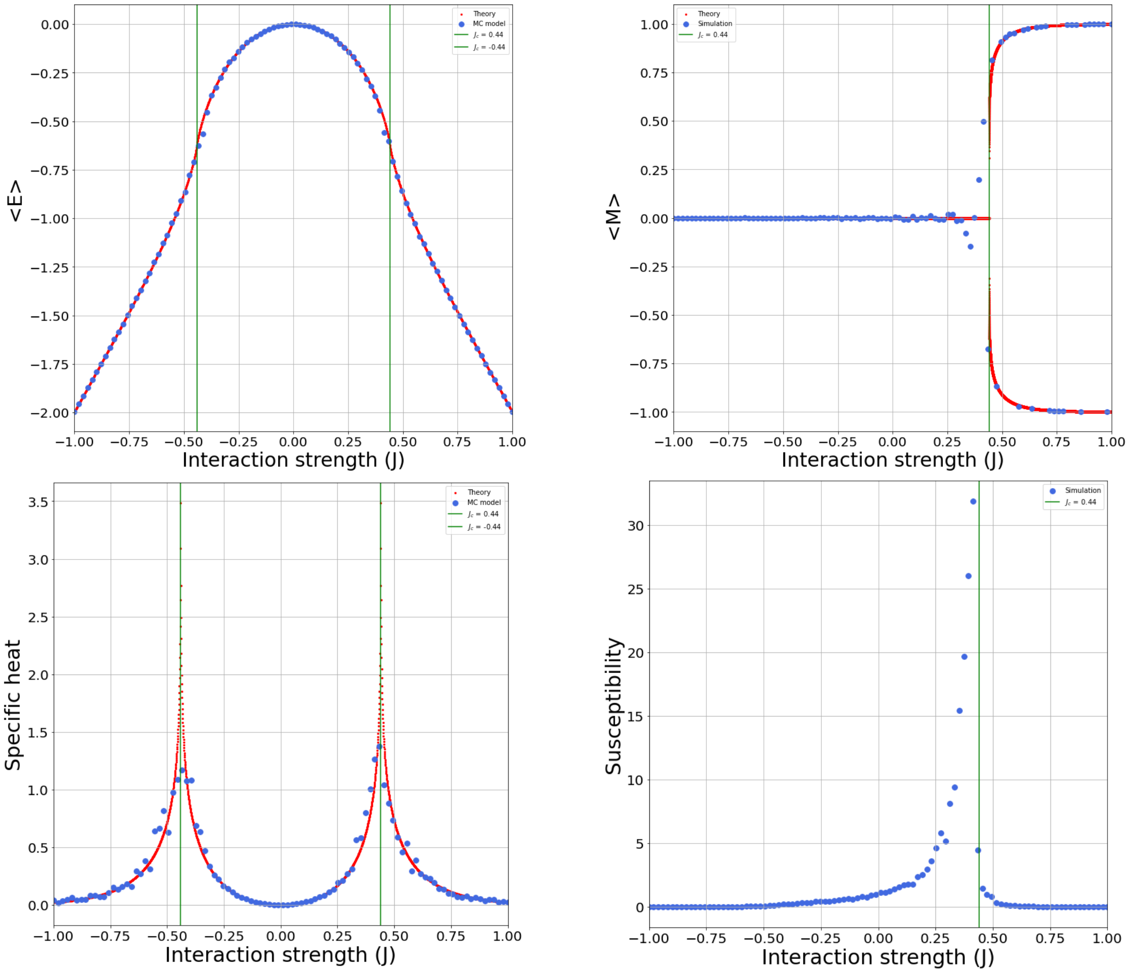

Figure 3.4: 2D NN Model - Interaction Strength Response - (T $= 1 < T_c$)

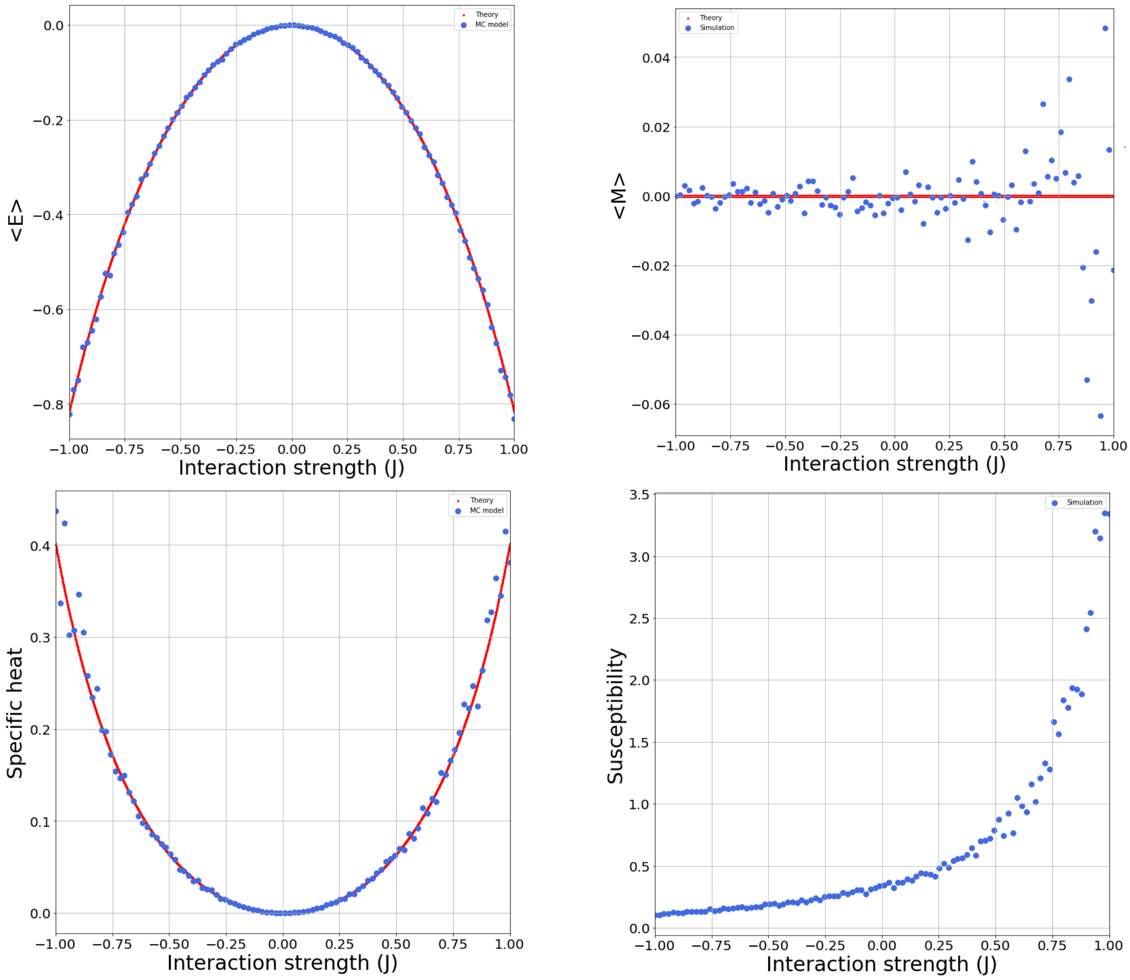

Figure 3.5: 2D NN Model - Interaction Strength Response - (T $= 3 > T_c$)

Here the phases of the system are described below,

- When $J < 0$, the system is **antiferromagnetic**, This is ture always and independent of temperature value for which the system is simulated. Here The $< M >$ is zero and susceptibility is very small positive value (nearly zero).

- $J \in [0, 1]$ and $T > T_c$ the system will be in **paramagnetic** phase throughout. No paramagnetic to ferromagnetic phase transition occurs. The nature of the graphs are similar to the 1D interaction strength response, Fig.3.2

- When, $J \in [0, 1]$ and $T < T_c$, then J_c can be computed from eqn.3.15, then,
  - For $J \in [0, J_c)$, the system is in **paramagnetic** phase. Here $< M >$ per spin is zero and susceptibility increases from zero and becomes maximum at J_c.
  - For $J > J_c$, the system is in **ferromagnetic** phase. Here $< M >$ per spin has two stable values $+1$ and -1. Susceptibility becomes zero.

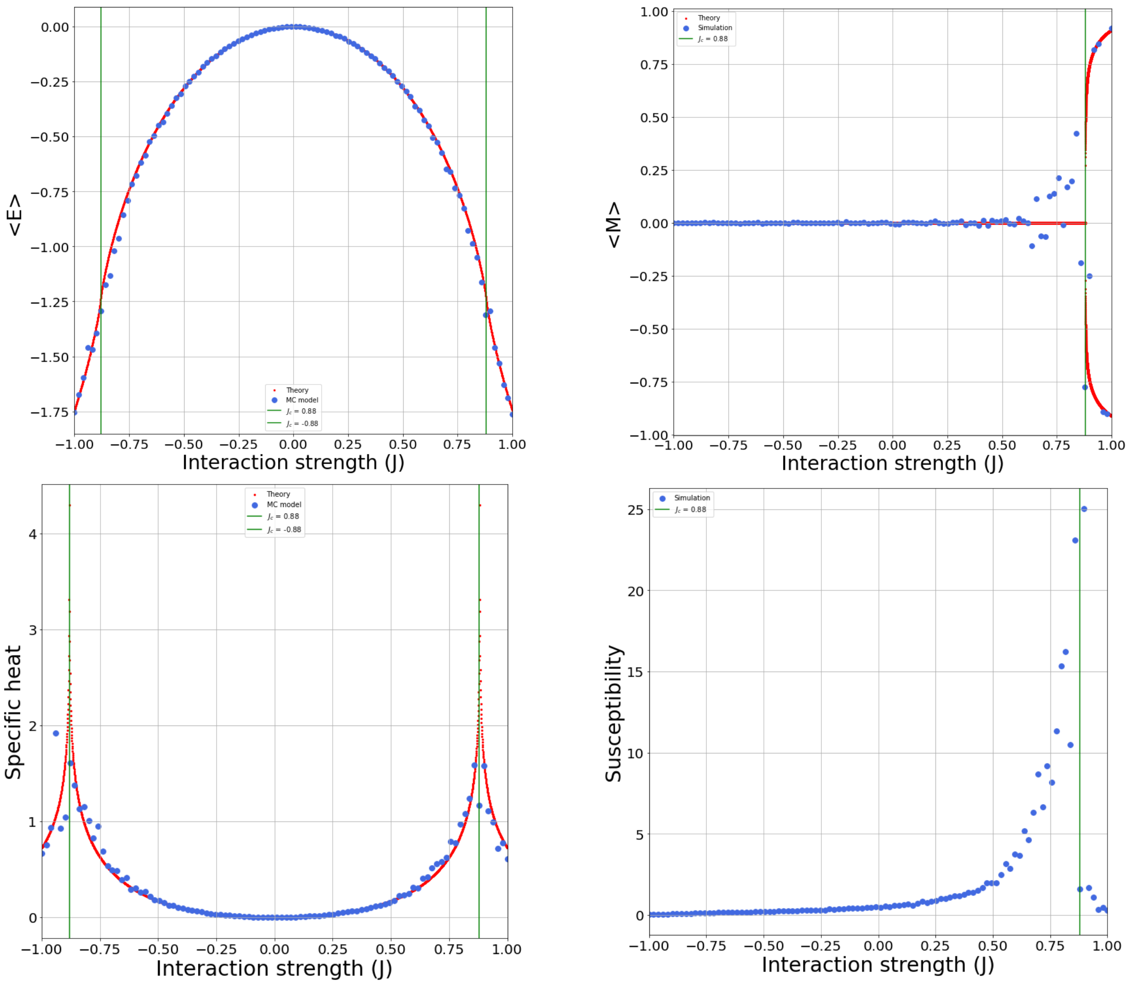

Figure 3.6: 2D NN Model - Interaction Strength Response - (T $= 2 \approx T_c$)

Here also there is symmetry noticed in the $< E >$ and specific heat plots at all the temperatures.

3.7 3D model - Temperature Response

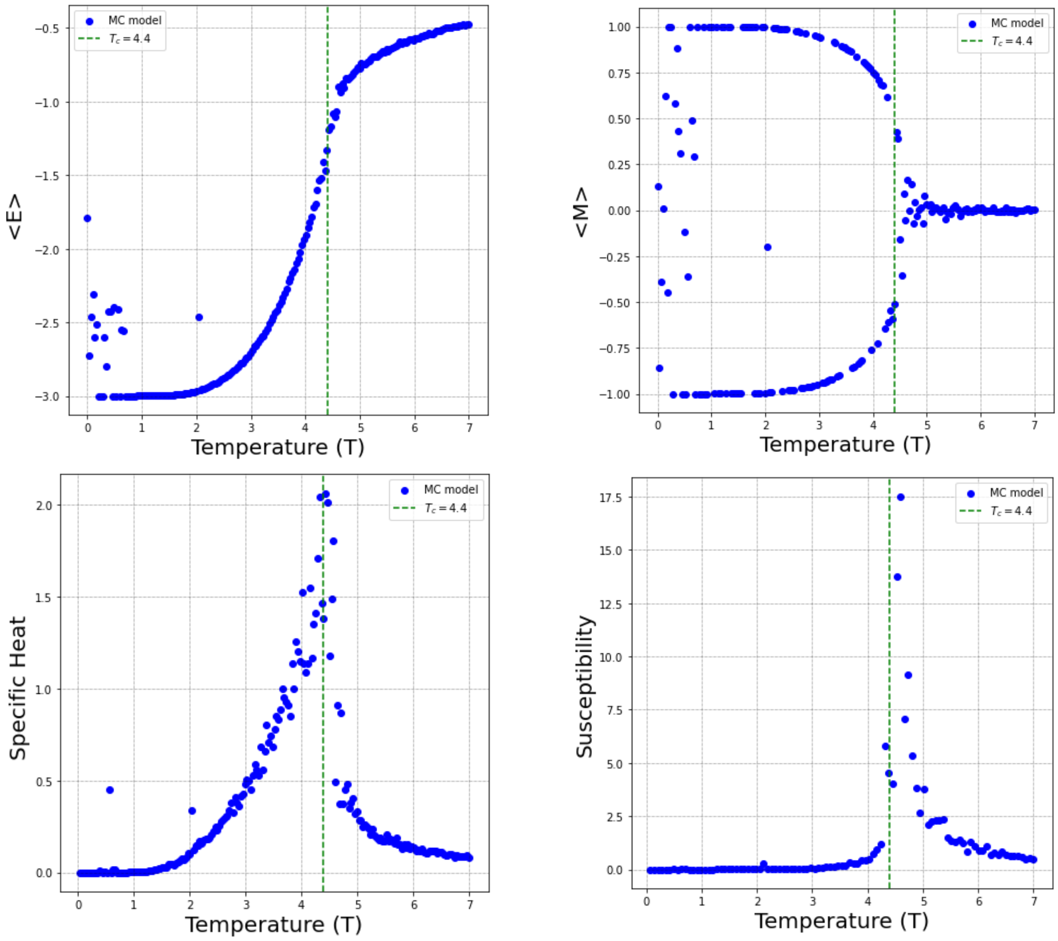

Figure 3.7: 3D NN Model - Temperature Response

3D model is a cubic lattice in which spins are placed on the corners of its unit cell, so each spin has 6 nearest neighbours. There is **no existing analytical solution** for 3D model. So, to study the system and to find the phase transitions, we have to rely entirely upon simulations. MCMC algorithm is used to simulate this model[3]. The relation between J and T is obtained for this model up to two decimal accuracy based on the simulation data using **Binder's cumulant** which shall be discussed in further section.

$$\boxed{T_c \approx 4.50J} \tag{3.16}$$

A lattice of size (10 x 10 x 10) is chosen and periodic boundary condition is considered. J is fixed, (J=1). 1000 MC steps are taken for thermalization. The quantities are averaged over 2000 MC steps after thermalization. We observed the phase transition at the temperature $T_c \approx 4.4$ (cf Fig3.7) which is close to the value in eqn 3.16. The error is statistical in nature since we considered a smaller lattice size and the number of MC steps to equilibrate is not enough. These considerations are due to limited computational resources.

3.8 3D model - Interaction Strength Response

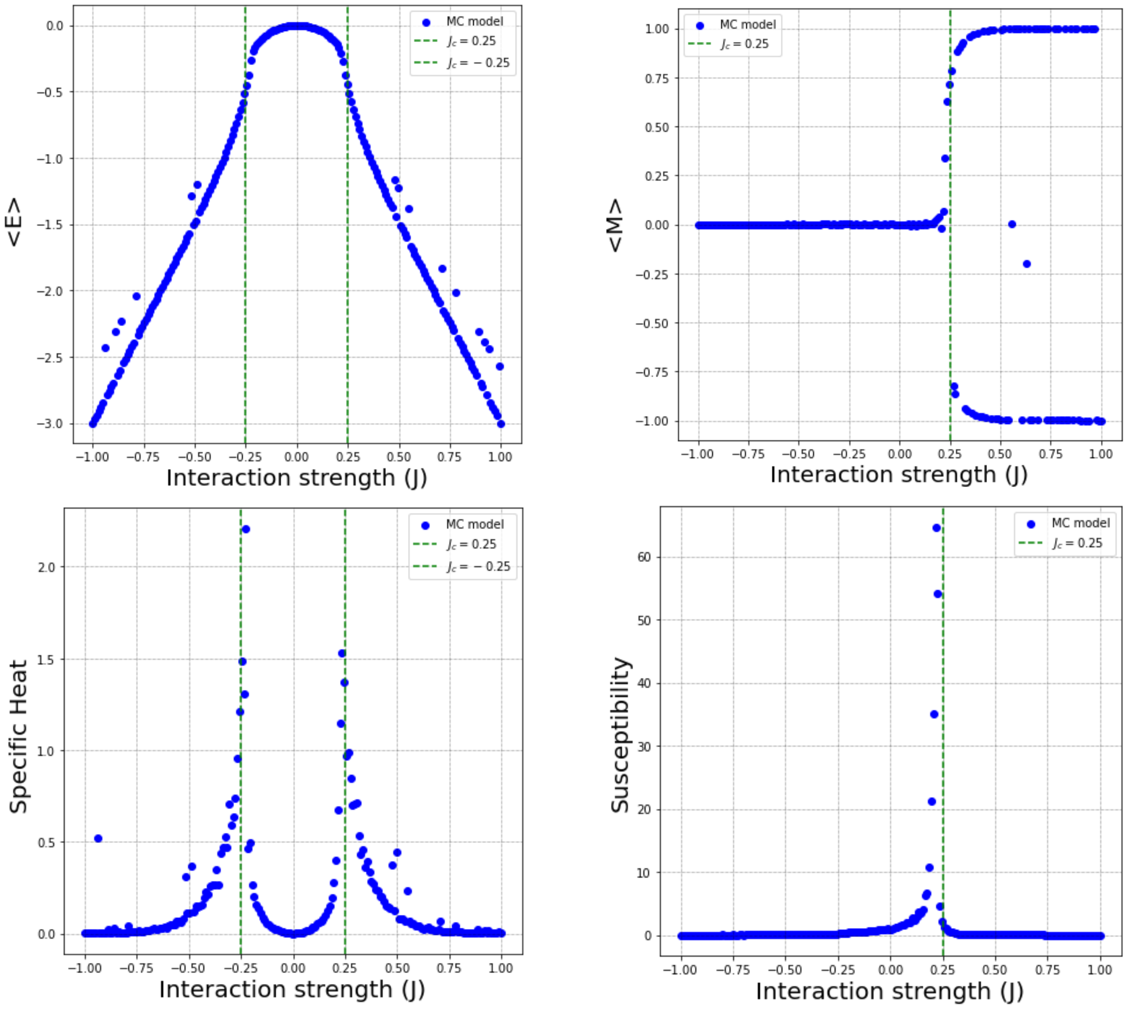

Figure 3.8: 3D NN Model - Interaction Strength Response - (T $= 1 < T_c$)

A lattice size (10 x 10 x 10) is chosen and periodic boundary condition is assumed. 1000 MC steps are taken for thermalization. The quantities are averaged over 2000 MC steps after the thermalization. Since 3D model shows phase transition, this model is simulated for three different temperature values (T $< T_c$, T $\approx T_c$ and $T > T_c$).

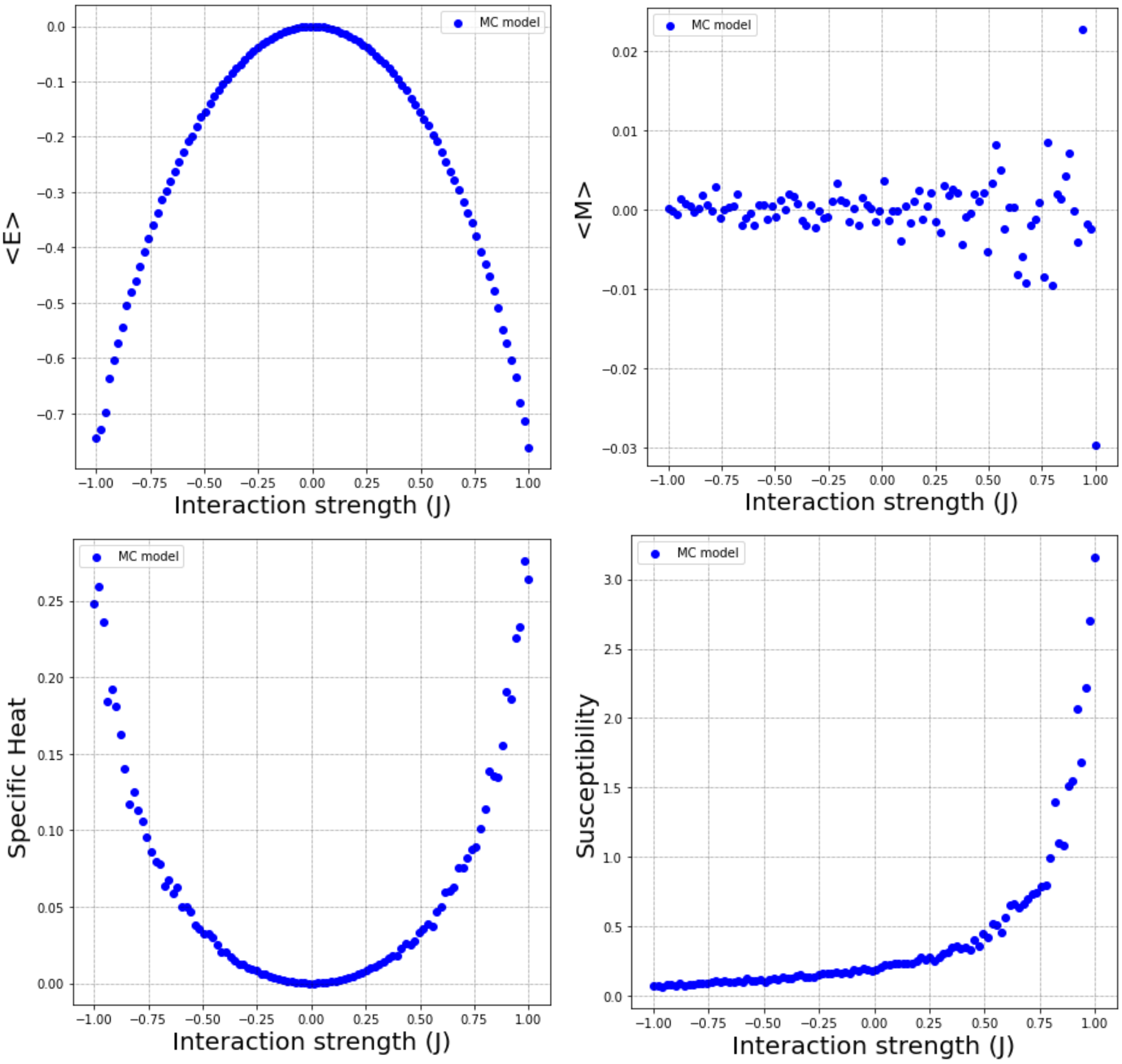

Figure 3.9: 3D NN Model - Interaction Strength Response - (T $= 5 > T_c$)

Though having no analytic solutions, the behaviour of the 3D NN model is similar to that of 2D NN model in all the qualitative aspects. The only difference is that the phase transition occurs at a higher temperature. The relationship between J_c and T is given by,

$$\boxed{J_c \approx 0.22T} \tag{3.17}$$

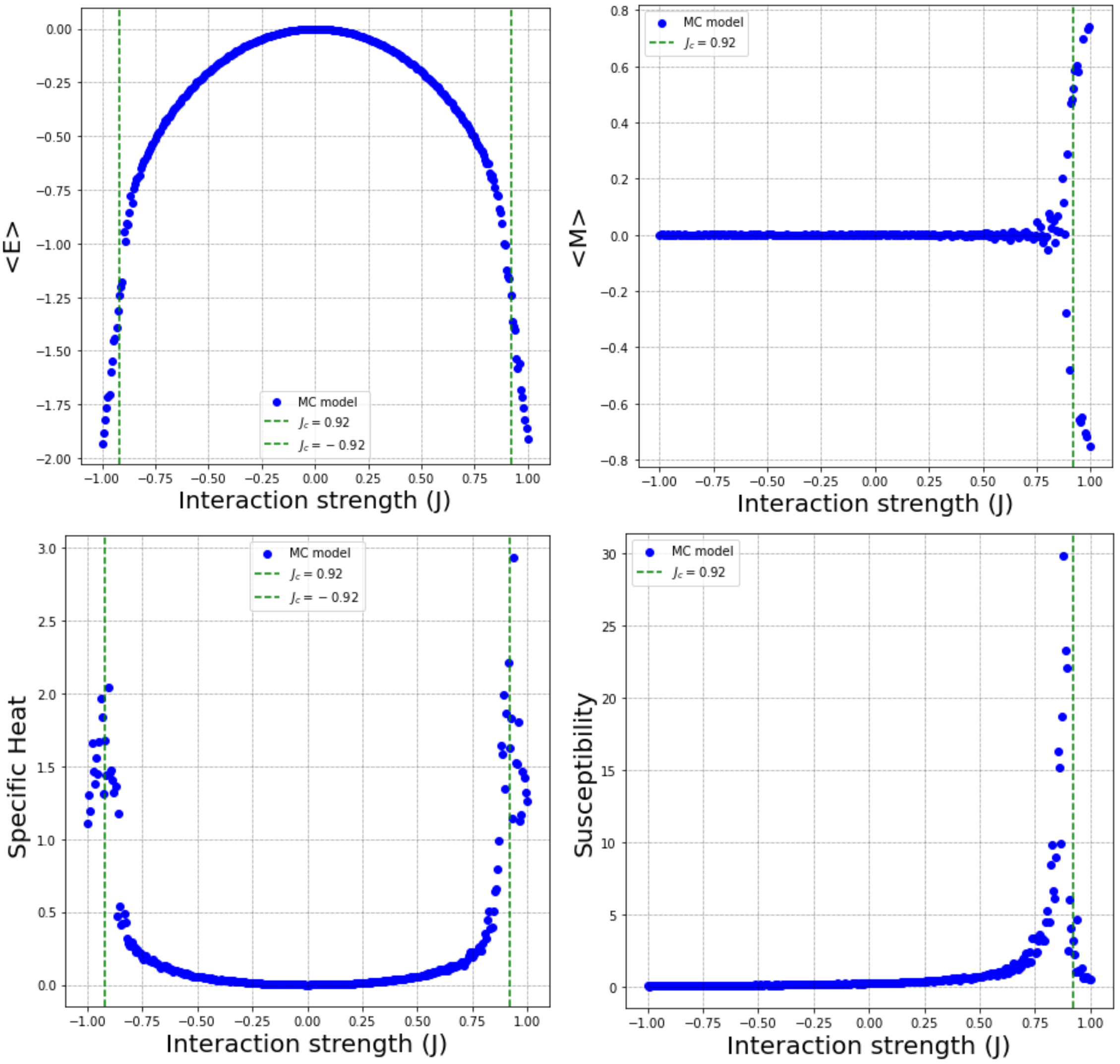

Figure 3.10: 3D NN Model - Interaction Strength Response - (T $= 4 \approx T_c$)

4 NEXT NEAREST NEIGHBOUR MODEL

These model also has no analytical solution. The lattice considered is same as before but now the next nearest interactions are also considered in the model. Again for simplification purposes, no external field is considered ($H = 0$), The interactions in the lattice are assumed to be isotropic and natural units is used ($k_B = 1$). Then from eqn.1.9,

Hamiltonian of NNN Model

$$\boxed{\mathcal{H} = -J_1 \sum_{<ij>} S_i S_j - J_2 \sum_{<<ik>>} S_i S_k} \tag{4.1}$$

Here $< ij >$ means the nearest neighbours only, and $<< ik >>$ means the next nearest neighbours only. Again the minus sign in the hamiltonian is for conventional reason ($J_i > 0$, means the interaction is ferromagnetic). J_1 and J_2 are the interaction strengths for NN and NNN interactions respectively. J_1 and J_2 are assumed to be related by the following equation,

$$J_2 = rJ_1 \tag{4.2}$$

Where, r is just a dimensionless proportionality constant. The critical temperature is calculated using the Binder's cumulant method.

4.1 2D model

To study the effects of the NNN neighbours, a square lattice (10 x 10) is considered with periodic boundary conditions and the i^{th} spin in this model has 4 NN spins and 4 NNN spins. This model is simulated for various values of r $\in$ {0, 0.3, 0.6, 0.9}. The system is allowed to thermalise for 1000 MC steps and the quantities computed were averaged over 2000 MC steps.

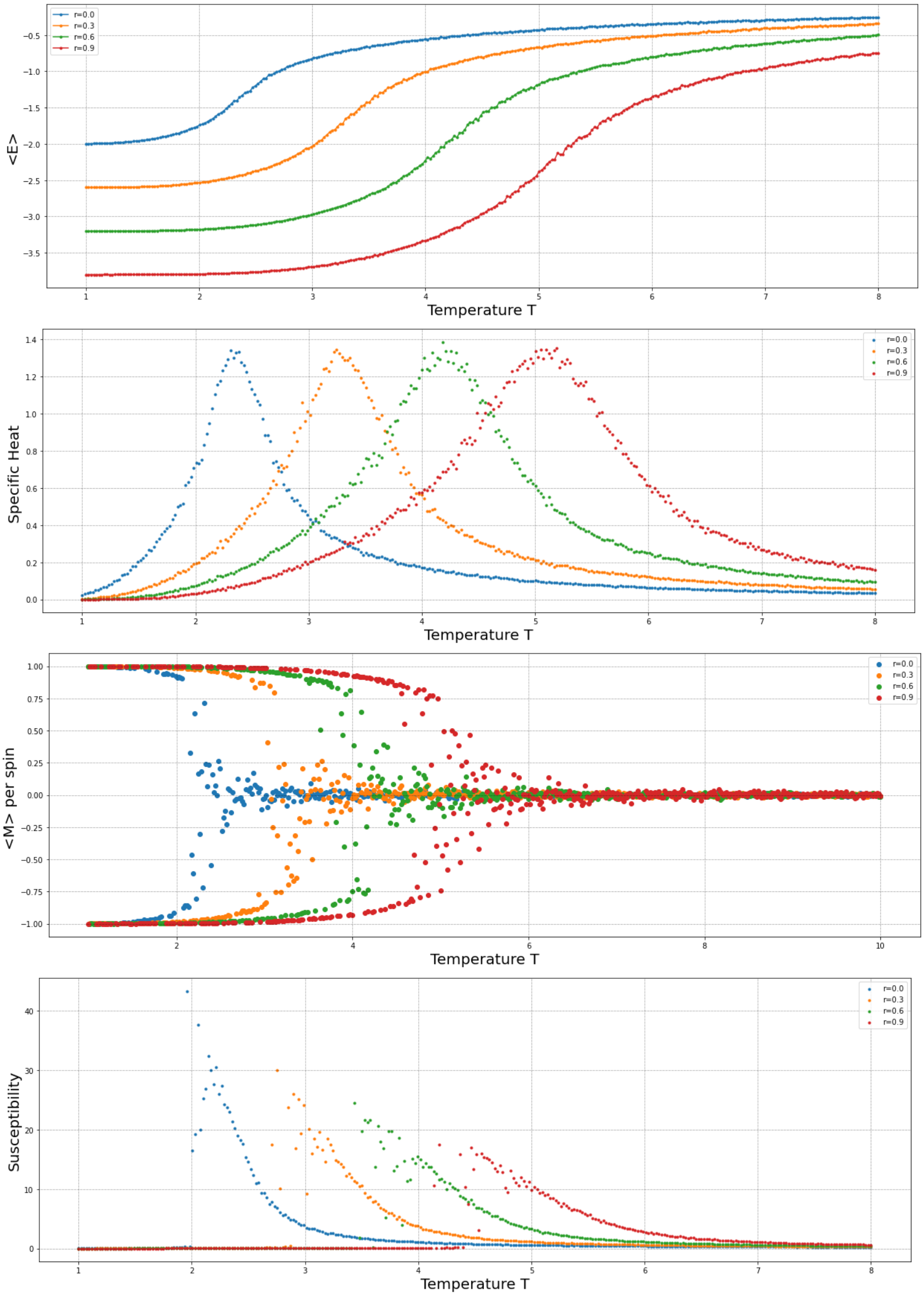

Figure 4.1: Plots of 2D NNN model

4.2 Binder's cumulant

In order to arrive at the phase transition temperature of the various NNN models, a quantity called Binder's cumulant is computed for all those systems. Binder's cumulant is a fourth order cumulant of the order parameter, magnetisation [7]. When Binder's cumulant as a function of temperature is plotted for different lattice sizes, it is observed that it crosses the same point at phase transition. Hence, the temperature at which the Binder cumulant curves intersect is considered as the **critical temperature** (T_c). It is defined as,

$$U = 1 - \frac{< M^4 >}{3 < M^2 >^2} = \begin{cases} \frac{2}{3} & T < T_c \\ U_B & T = T_c \\ 0 & T > T_c \end{cases} \tag{4.3}$$

In the process of computation, 5000 MC steps were ignored for the system to thermalise. The Binder's cumulant is averaged over 15000 MC steps.

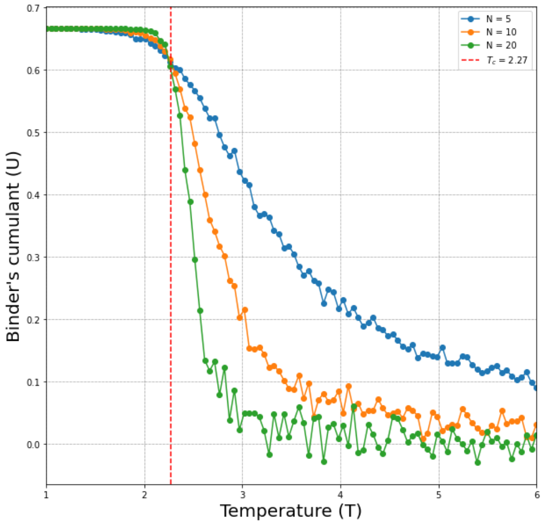

Figure 4.2: Binder's cumulant - 2D NN model: $T_c \approx 2.27$

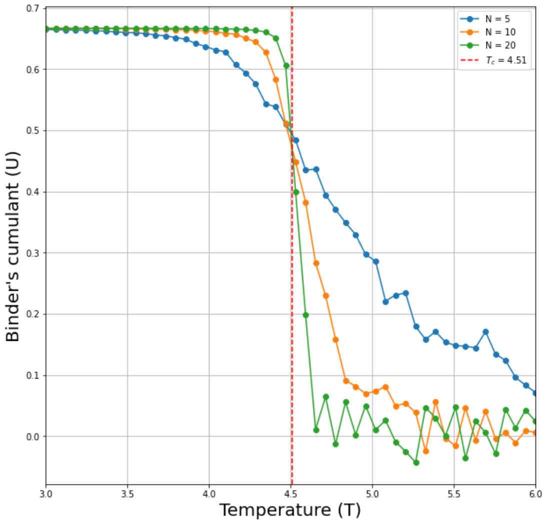

Figure 4.3: Binder's cumulant - 3D NN model: $T_c \approx 4.50$

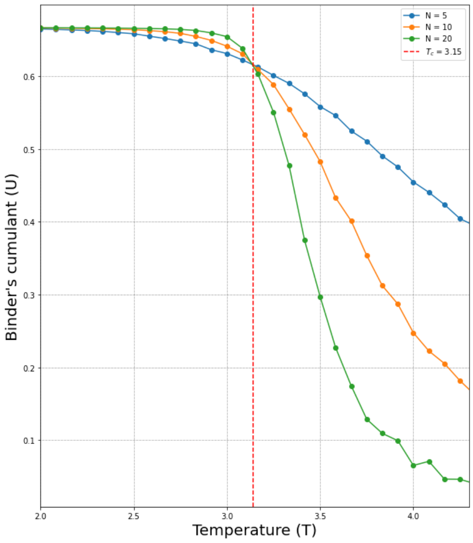

Figure 4.4: Binder's cumulant - 2D NNN model: $J_2 = 0.3$ - $T_c \approx 3.15$

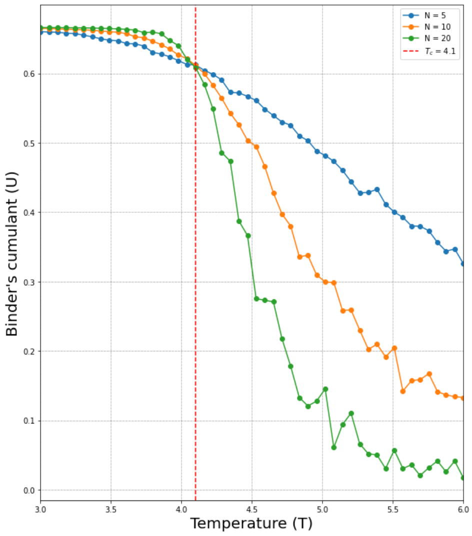

Figure 4.5: Binder's cumulant - 2D NNN model: $J_2 = 0.6$ - $T_c \approx 4.1$

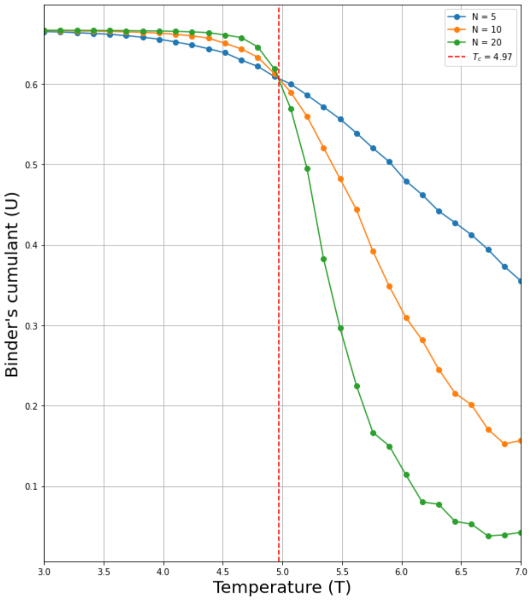

Figure 4.6: Binder's cumulant - 2D NNN model: $J_2 = 0.9$ - $T_c \approx 4.97$

5 SUMMARY AND FUTURE WORK

Summary

- The 1D NN Ising model simulation results was in agreement with the analytical results and no phase transition was observed in this system. The 2D NN Ising model was also in agreement with the analytical results but a second order phase transition was noticed.
- The 3D NN model simulations showed a second order phase transition and was also in agreement with the literature.
- While analysing the interaction strength response. 2D and 3D NN models showed similar kind of behaviour. When $T > T_c$, these responses matched with the 1D simulation results (no phase transition).
- The next nearest neighbour interaction was taken into consideration in the 2D model. These new interaction terms increased the critical temperature of the second order phase transition.
- Binder's cumulant results,

MODEL	$\approx k_B T_c / J$ (unitless)
1D NN	no phase change
2D NN	2.27
3D NN	4.50
2D NNN (r = 0.3)	3.15
2D NNN (r = 0.6)	4.10
2D NNN (r = 0.9)	4.97

the temperature (T) mentioned throughout this entire report was a dimensionless quantity equal to $k_B T/J$, since $k_B = 1$ assumption was made.

Future work

A real crystal spin systems with impurities or spinless sites could be simulated. This Monte Carlo simulation technique can also be used as an alternate method to DFT, when it comes to computing the exchange interaction (J_{ij}) values in the real spin systems with multiple exchange interactions. When the electronic structure involves multiple exchange couplings, DFT method becomes more difficult to compute them. In the context of this, O. El Rhazouani et al [1]. developed a method that combined the energy E, magnetization M and Monte Carlo simulation-Ising model. This method makes use of the experimental phase transition temperature to compute the exchange interactions.

6 APPENDIX

6.1 Python code - 3D NN Model - Interaction strength response

```
import numpy as np
from numpy.random import rand
import matplotlib.pyplot as plt
##### FUNCTIONS USED IN THE MAIN CODE #####
def initialstate(N):
    # creates a 3D array of numbers
    state = 2*np.random.randint(2, size=(N,N,N))-1
    return state

def mc_loop(config, beta):
    # Monte carlo loop for updating configuration
    for i in range(N):
        for j in range(N):
            for k in range(N):
                # select a  spin
                spin =  config[i, j, k]
                # calculate delta_E due to spin flip
                nn = (config[(i+1)%N, j, k]\
                    + config[(i-1)%N, j, k]\
                    + config[i, (j+1)%N, k]\
                    + config[i, (j-1)%N, k]\
                    + config[i, j, (k+1)%N]\
                    + config[i, j, (k-1)%N])   # nn = nearest neighbours
                delta_E = 2 * spin * nn
                r = rand()
                # condition for acceptance of spin flip
                if (delta_E <= 0):
                    spin *= -1
                elif (delta_E > 0 and r <= np.exp(-delta_E * beta)):
                    spin *= -1
                elif (delta_E >0 and r > np.exp(-delta_E * beta)):
                    spin = spin
                # updating configuration
                config[i, j, k] = spin
    return config

def Energy(config, J):
    # Calculating the energy of a configuration
    energy = 0
    for i in range(len(config)):
        for j in range(len(config)):
            for k in range(len(config)):
                spin = config[i,j,k]
                nn = (config[(i+1)%N, j, k]\
                   + config[(i-1)%N, j, k]\
                   + config[i, (j+1)%N, k]\
                   + config[i, (j-1)%N, k]\
                   + config[i, j, (k+1)%N]\
                   + config[i, j, (k-1)%N])   # nn = nearest neighbours
                energy += -J * nn * spin
    return energy/2.    # to compensate for over-counting

def Mag(config):
    # calculates the magnetization of the configuration
    mag = np.sum(config)
    return mag

##### MAIN CODE #####
# Parameters of MC loop
```

```python
jp = 300             #   number of J points
N = 10               #   lattice size
eqSteps = 2000       #   number of MC steps to thermalise
mcSteps = 2000       #   number of MC sweeps for averaging quantities

# Parameters of the simulation
T = 3                # Temperature value
J = np.linspace(-1,1, jp);
Jc = 0.23
E = np.zeros(jp)
M = np.zeros(jp)
C = np.zeros(jp)
X = np.zeros(jp)
n1 = 1.0/(mcSteps*N*N*N)
n2 = 1.0/(mcSteps*mcSteps*N*N*N)

##### MAIN LOOP #####
for jj in range(jp):
    configuration = initialstate(N)             # initialise
    E1 = M1 = E2 = M2 = 0
    iT=1.0/T; iT2=iT*iT;                        # defining beta
    for i in range(eqSteps):                    # Loop to thermalising
        mc_loop(configuration ,J[jj], iT)
    for i in range(mcSteps):                    # Loop to compute quantities
        mc_loop(configuration ,J[jj], iT)
        Ene = Energy(configuration , J[jj])
        # Calculating quantities
        E1 = E1 + Ene
        M1 = M1 + Mag
        M2 = M2 + Mag*Mag
        E2 = E2 + Ene*Ene
    # Storing the computed values in an array
    E[jj] = n1*E1
    M[jj] = n1*M1
    C[jj] = (n1*E2 - n2*E1*E1)*iT2
    X[jj] = (n1*M2 - n2*M1*M1)*iT

## Plot the graphs appropriately ##
```

Graph Plotting section of the code is not shown

6.2 Python code - 2D NNN model Binder's Cumulant

```
import numpy as np
import matplotlib.pyplot as plt
from numpy.random import rand
##### Functions Used in Main Code #####
def initialstate(N):
    # creates a 2D array of numbers
    state = 2*np.random.randint(2, size=(N,N))-1
    return state

def mc_loop(config, beta):
    # Monte carlo loop for updating configuration
    for i in range(N):
        for j in range(N):
            # select a  spin
            spin =  config[i, j]
            # calculate delta_E due to spin flip
            nn = (config[(i+1)%N, j]\
                + config[(i-1)%N, j]\
                + config[i, (j+1)%N]\
                + config[i, (j-1)%N])       # nn = nearest neighbours
            E_nn = J1 * spin * nn
            nnn = (config[(i+1)%N,(j-1)%N]\
                + config[(i+1)%N,(j+1)%N]\
                + config[(i-1)%N,(j+1)%N]\
                + config[(i-1)%N,(j-1)%N])  # nnn = next nearest neighbours
            E_nnn = J2 * spin * nnn
            delta_E = 2*(E_nn + E_nnn)
            r = rand()
            # condition for acceptance of spin flip
            if (delta_E <= 0):
                spin *= -1
            elif (delta_E > 0 and r <= np.exp(-delta_E * beta)):
                spin *= -1
            elif (delta_E >0 and r > np.exp(-delta_E * beta)):
                spin = spin
            # updating configuration
            config[i, j] = spin
    return config

def Mag(config):
    # calculates the magnetization of the configuration
    mag = np.sum(config)
    return mag

##### Main code #####
# Parameters

nt      = 50            # number of temperature points
N       = [5,10,20]     # lattice sizes
eqSteps = 1000          #  number of MC steps to thermalise
mcSteps = 3000          #  number of MC sweeps for averaging quantities
J1 = 1
```

```
J2 = 0.3 * J1           # 0.3 = r value
T=np.linspace(3,6,nt)
binder = np.zeros((len(N),nt))

# Main loop
for n in range(len(N)): # repeats the loop for different lattice size
    for tt in range(nt):
        configuration = initialstate(N[n])   # initialise
        M2 = M4 = 0
        beta=1/T[tt]                         # defining beta
        for i in range(eqSteps):             # Loop to thermalising
            mc_loop(configuration, beta)
        for j in range(mcSteps):             # Loop to compute binder cumulant
            mc_loop(configuration, beta)
            mag = Mag(configuration)
            M2=M2+mag**2
            M4=M4+mag**4
            m2=M2/(mcSteps)
            m4=M4/(mcSteps)
        binder[n,tt]=(1-m4/(3*m2**2))

# Plotting the graph
a= plt.figure(figsize=(10,10))
plt.xlim(3,6)
plt.rc(’grid’, linestyle=’:’, color=’black’, linewidth=0.5)
plt.grid(True)
plt.xlabel("Temperature (T)", fontsize=20);
plt.ylabel("Binder’s cumulant (U)", fontsize=20);
for n in range(len(N)):
    plt.plot(T,binder[n],marker=’o’,label =’N = %d’%N[n])
plt.legend()
```

Bibliography

[1] O El Rhazouani **andothers**. “Exchange couplings calculation in the Double Perovskite Sr_2CrWO_6”. **in***Physica A: Statistical Mechanics and its Applications*: 476 (2017), **pages** 38–42.

[2] Dr.Abhijit Kar Gupta. *Scientific Computing in Python*. Techno World, 2021.

[3] “Computational analysis of 3D Ising model using metropolis algorithms”. **in***Journal of Physics: Conference Series*: **volume** 630. 1. IOP Publishing. 2015, **page** 012057.

[4] Kerson Huang. *Statistical mechanics*. John Wiley & Sons, 2008.

[5] Anosh Joseph. “Markov Chain Monte Carlo Methods in Quantum Field Theories: A Modern Primer”. **in**(2019): DOI: `10.1007/978-3-030-46044-0`. eprint: `arXiv:1912.10997`.

[6] Frederick Reif. *Fundamentals of statistical and thermal physics*. Waveland Press, 2009.

[7] W. Selke. “Critical Binder cumulant for isotropic Ising models on square and triangular lattices”. **in**(2007): eprint: `arXiv:cond-mat/0701515`.

[8] Badis Ydri. *Computational Physics: An Introduction to Monte Carlo Simulations of Matrix Field Theory*. 2015. eprint: `arXiv:1506.02567`.