

# Final Project for Computational Physics: This is only for my personal use, not final report

Xinyu Liu

December 12, 2024

## Abstract

This is a code to simulate the 2D ising model under certain temperature. The basic idea for studying a thermal system is to investigate the partition function by computational method. In a word, we can choose a ensemble of states by computers, attributing each state with a boltzman probability and sum over all the state to calculate the expectation energy for the system under a certain temperature. However, a random sampling is highly ineffective for low temperature physics, since most possible states are irrelevant in low temperature. The way to carry out an more effective sampling method is by using Metropolis Algorithm, which is introduced in detail in Newman Chapter 10.3. In this work, we utilize Metropolis Algorithm to study a 2D spin system with 2D exchanging interaction and external field.

## Contents

<b>1</b>	<b>My Github Page URL</b>	<b>2</b>
<b>2</b>	<b>Physics Model</b>	<b>2</b>
<b>3</b>	<b>Structure of the Code</b>	<b>2</b>

# 1 My Github Page URL

<https://github.com/rising1227/phys-ga2000>

## 2 Physics Model

We investigate the physics with simplest exchange Ising interaction with an external field:

$$H = \sum_{\langle i,j \rangle} -J(S_i S_j + 1) + B_k S_k \quad (1)$$

Here the index  $i,j,k$  denote the label of particular spin in the spin lattice. We modified it a bit such that the ground state of the system has zero energy (without magnetic field).  $B_k$  denotes an external field and  $\langle i,j \rangle$  denotes neighbouring pairs of spins.  $B_k$  denotes an external field. First of all, we carry out the simulation with no external field, i.e.  $B = 0$ , then we study this model with constant external field  $B_k = B_0$ . Finally, we investigate the system with spatially varying field  $B_k$ .

## 3 Structure of the Code

In order to make the code clear and make it easier for collaboration, we divide our code into different blocks. Each account for a specific purpose.

- The physical system is represented by a  $M \times M$  numpy array. Each spin is represented by a bool type variable. This is the most compact variable to represent a spin. We use True to represent spin up  $\uparrow$  and False to represent spin down  $\downarrow$ . We denote our matrix as  $S$ . We also use a function to generate a initial spin array  $S_0$
- A external magnetic field is denoted by a  $M \times M$  numpy array  $B$ .  $B[i][j]$  represent the external magnetic field at each point. The total energy of initial system can be calculated.
- A energy calculation function take  $S$  and  $B$  as input and calculate the total energy.

- A transfer function "Trans" is one step in the markov chain in the Metropolis method. It takes S, B, Total magnetization M, total energy E and temperature T as the input. It changes the S internally, while outputting a renewed M and E.
- Then we define a Metropolis function "Metro" which takes "Trans" function as an input. Also inputing the temperature T, S, B, M, E. This metropolis function reiterates the transfer function for many times and detect whether our system goes to thermal equilibrium. After thermal equilibrium, it outputs a averaged total energy and magnetization at a certain temperature.
- At main function, we utilize this metropolis function many times for different temperature. Finally, we can have a full curve of E and M as a function of temperature.