Project 3: Stochastic Dynamics and Phase Transitions: A Study of Random Walks and the Ising Model

Introduction

The current project consists of two parts: i) modeling diffusion with a random walk, and ii) investigating the thermodynamic properties of the Ising model via the Metropolis Algorithm.

A **random walk** is a mathematical model describing a path consisting of successive random steps. It is widely used in physics, biology, finance, and other fields to model phenomena such as Brownian motion, particle diffusion or stock market fluctuations.

The **2D Ising model** is an important model in statistical mechanics and has contributed to the development of our understanding of **phase transitions** and the behavior of complex systems. It is a mathematical model used to study the behavior of a collection of spins on a two-dimensional lattice. The Iaing model was introduced by Ernst Ising in 1925 as a **simplified model of ferromagnetism**, where each spin can either be in an "up" or "down" state. The model has found numerous applications in different fields of physics, including statistical mechanics, condensed matter physics, and even computer science. One of the most important applications of the 2D Ising model is its use as a **prototype for studying phase transitions**.
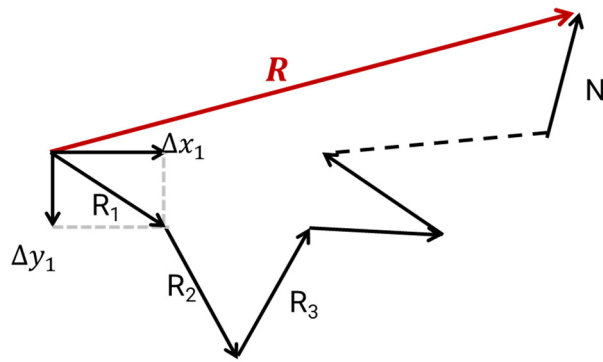
Random Walk

Fick's laws of diffusion describe the movement of particles from regions of high concentration to low concentration, driven by a concentration gradient. The first law states that the diffusive flux is proportional to the concentration gradient:

$$\boldsymbol{J} = -D\boldsymbol{\nabla}\rho \tag{1}$$

while the second law describes how concentration evolves over time due to diffusion:

$$\frac{\partial \rho}{\partial t} = D\boldsymbol{\nabla}^2 \rho \tag{2}$$

In the above Eqs $\boldsymbol{J}$ is the flux, ρ is the density or concentration of the substance, and D is the diffusion constant. Homogeneous medium is considered, i.e., D is the same everywhere in the system.

Figure 1 Schematic representation of *N* steps in a random walk simulation that end up a distance *R* from the origin.

The connection between Fick's laws and the random walk arises from the statistical behavior of particles undergoing Brownian motion. A random walk models diffusion at a microscopic level, where each particle moves in discrete steps in random directions. Over time, the probability distribution of particle positions follows the diffusion equation, which is derived from Fick's second law. Specifically, in a continuous limit, the mean squared displacement $\langle R^2 \rangle$ of a random walker grows linearly with time t, consistent with the diffusion coefficient in Fick's law:

$$\langle R^2 \rangle = 2Dt, \text{ (in 1D)}$$
$$\langle R^2 \rangle = 4Dt, \text{ (in 2D)} \tag{3}$$
$$\langle R^2 \rangle = 6Dt, \text{ (in 3D)}$$
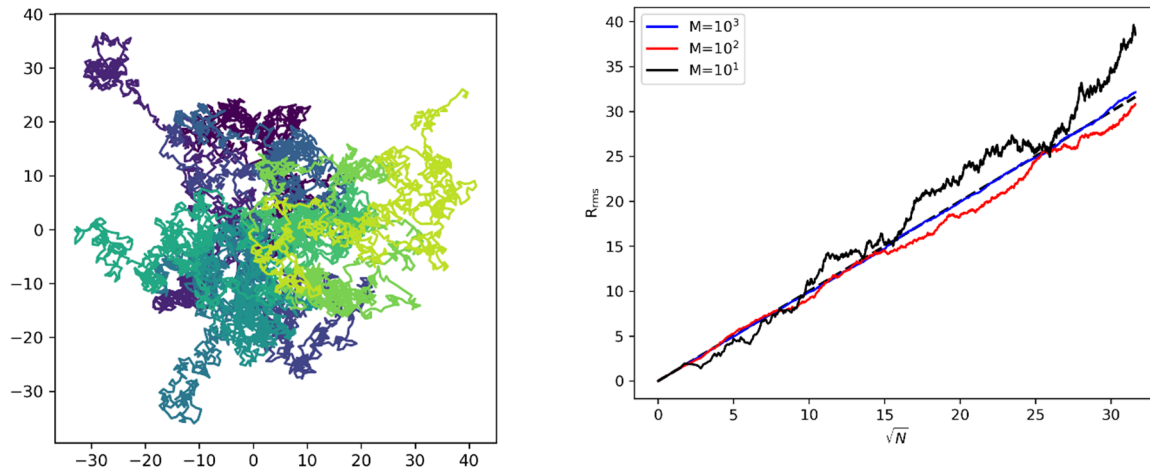
Figure 2: Left: Steps taken in ten 2D random walk simulations Right: RMS distance from origin vs the $\sqrt{N}$ calculated from 10^1, 10^2, and 10^3 random walks.

In our random walk model we will consider a walker that takes sequential steps of unit length at random directions and with the direction of each step independent of the direction of the previous step. We start at the origin and take N steps in the XY plane of lengths:

$$(\Delta x_1, \Delta y_1), (\Delta x_2, \Delta y_2), (\Delta x_3, \Delta y_3), \dots, (\Delta x_N, \Delta y_N)$$

It can be shown that the root-mean-square (RMS) distance from the origin at step N is propotinal to the square root of N:

$$R_{rms} = \sqrt{\langle R^2 \rangle} \approx \sqrt{N} \tag{4}$$

Task 1: Implement a 2D Random Walk (5 pnts)

Steps:

- Initialize Parameters
  - Number of steps per walk, N
  - Number of trials, M
- Generate Random Steps
  - For each step, generate random $\Delta x, \Delta y \in [-1,1]$[1]
  - Normalize to ensure step length = 1:
  $$\Delta x = \frac{\Delta x}{\sqrt{\Delta x^2 + \Delta y^2}}, \Delta y = \frac{\Delta y}{\sqrt{\Delta x^2 + \Delta y^2}}$$
- Track Position
  - Update position: $x_{i+1} = x_i + \Delta x$, $y_{i+1} = y_i + \Delta y$
- Compute Squared Distance
  - For each step and each walk, calculate $R^2 = x^2 + y^2$

---

[1] Another equivalent way to compute the steps $\Delta x, \Delta y$ is to generate a random angle $\theta \in [0, 2\pi]$ at each step (`theta=2*np.pi*random.random()`). Then, the step components are given by $\Delta x = \cos\theta$ and $\Delta y = \sin\theta$.

- Average Over Trials
   - Compute $\langle R^2 \rangle$ across M trials

$$R_{\text{rms}}^2 = \langle R^2 \rangle = \frac{1}{M} \sum_{m=1}^{M} R_m^2$$

- Plot Results:
    - Plot steps taken in five 2D random walk simulations (see Fig. 2).
    - Plot $\sqrt{\langle R^2 \rangle}$ vs N (see Fig. 2).

The Ising Model

Ferromagnetic materials are materials that exhibit permanent magnetization in the absence of an external magnetic field. In a ferromagnetic material, the spins tend to align in the same direction, resulting in a net magnetic moment for the material. Ferromagnetic materials have a strong tendency to remain magnetized even after an external magnetic field is removed. However, the magnetization decreases with increasing temperature and vanishes above the critical temperature T$_C$, known as the **Curie temperature**, named after the French physicist Pierre Curie. As the temperature increases above the Curie temperature, entropic contributions to the free energy dominate, i.e., the thermal energy becomes strong enough to overcome the interactions between the spins, and the material loses its magnetization. At $T = T_C$ the system exhibits a second-order phase transition, i.e. the magnetization is not differentiable with respect to T, though it is continuous.

The microscopic origin of the aforementioned macroscopic phenomenon is based on the **exchange interaction** between identical particles, i.e., between the atoms or molecules forming the material. The exchange interaction is a purely quantum mechanical effect which is a consequence of the Coulomb
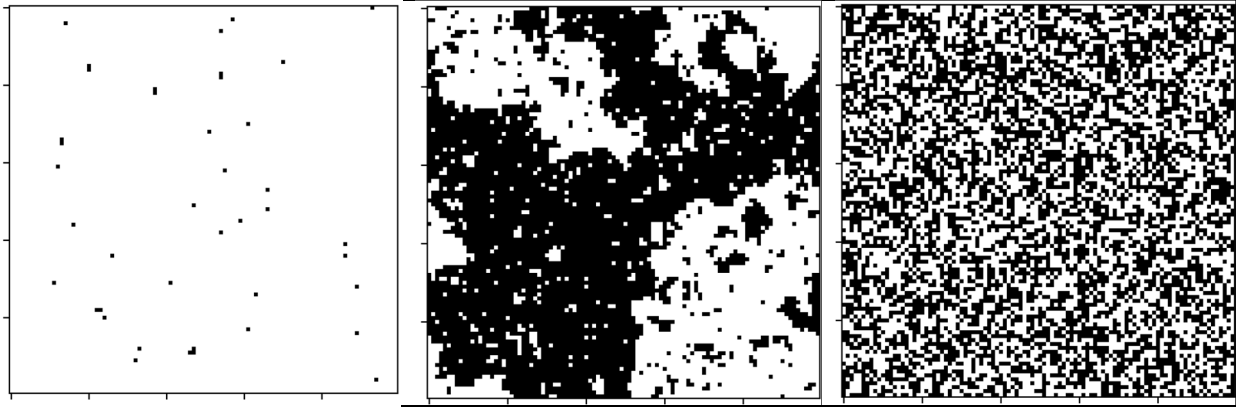
Figure 3: Characteristic spin configuration maps for $T < T_C$ (left), $T \approx T_C$ (middle) and $T > T_C$ (right). Black and white shaded areas correspond to spin up and spin down sites, respectively.

interaction in combination with the Pauli exclusion principle.

Let us consider a model in 2D, where the spin variables are located on the sites of an $N \times N$ square lattice. The spins can be labeled with a 2-index notation as S_{ij}, where i and j denote the indices of the two spatial directions. In our code, it will be considerably more convenient to use a single index notation for the spins, i.e., S_α.

Each of these spins can be either "up" ($S_\alpha = +1$) or "down" ($S_\alpha = -1$). In our model, we will take the spins to be classical degrees of freedom and the Hamiltonian of the system can be written as:

$$H = -\frac{1}{2}J \sum_\alpha S_\alpha \sum_\beta S_\beta - B \sum_\alpha S_\alpha \tag{5}$$

Where J is the exchange constant and B is the external magnetic field. The sums Σ_α are over all lattice sites while the sum Σ_β is over the neighbors of atom α. The prefactor $1/2$ prevents double counting.
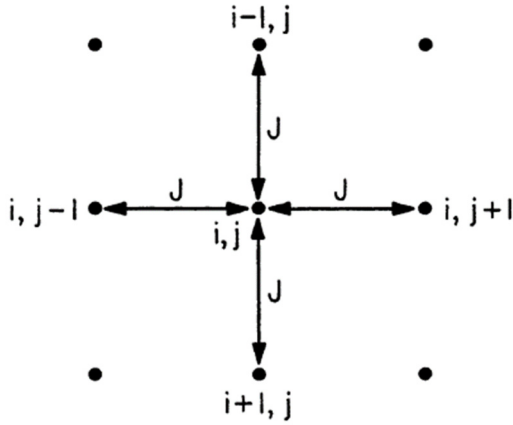
Figure 4: Schematic representation of the interactions in the 2D Ising model. An atom at site $\{i,j\}$ interacts only with the four nearest neighbors at sites $\{i-1,j\}$, $\{i+1,j\}$, $\{i,j-1\}$, and $\{i,j+1\}$. The strength of these interactions is given by the exchange parameter J.

If $J > 0$ then it is favorable for the spins to align in a parallel orientation, and the ferromagnetic behavior arises for $T < T_C$. If, on the other hand $J < 0$ then the antiparallel orientation is favorable and if the temperature is below a critical temperature (now called Néel temperature T_N) the antiferromagnetic behavior arises.
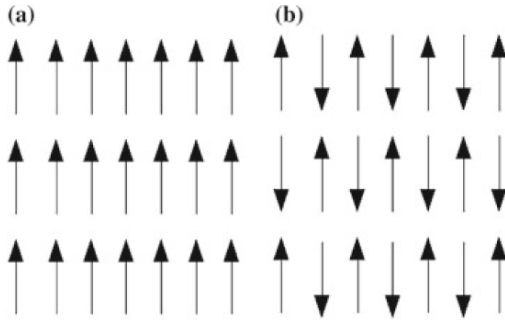
Figure 5: Schematic representation of the spin-confuguration in (a) ferromagnetic ($J > 0$) and (b) antiferromagnetic ($J < 0$) two-dimensional crystal.

The exact solution to the 2D Ising model was developed by Onsager in 1944. In the limit of infinite lattice size ($N \to \infty$) the Curie Temperature can be calculated from the following equation:

$$\frac{J}{k_B T_C} = \frac{1}{2}\ln\left(1 + \sqrt{2}\right) \approx 0.4406868 \tag{6}$$

Where k_B is the Boltzmann constant.

Calculating Observables

We would like to calculate numerically observables like the expectation value of the energy or of the magnetization. The expectation value of an observable O can be calculated as:

$$\langle O \rangle = \frac{1}{Z(T)}\sum_r O(r)\exp\left(-\frac{H_r}{k_B T}\right) \tag{7}$$

Where the sum is over all possible states r and H_r is the energy of the state r. $Z(T)$ is the canonical partition function:

$$Z(T) = \sum_r \exp\left(-\frac{H_r}{k_B T}\right) \tag{8}$$

To calculate the observables, we need numerical methods: Summing over all possible configurations is not feasible even for lattice sizes as small as 100×100. In this case, we have N^2 lattice sites and 2^{N^2} spin configurations. Therefore, the sums in Eqs (7) and (8) will be over $2^{10000} \approx 10^{3000}$ configurations!

However, the Boltzmann weight $\exp\left(-\frac{H_r}{k_B T}\right)$ of the majority of the spin configurations is very small (high energy configurations) and therefore their contribution to the partition function is negligible. Hence, it is a good approximation to use mainly those spin configurations with a significant contribution for the calculation of the partition function. The technique to generate these important configurations is called **importance sampling**. To achieve this, we can generate spin configurations r with a probability $\exp\left(-\frac{H_r}{k_B T}\right)$ using the **Metropolis algorithm** and then average the required observables over these configurations.

Metropolis Algorithm

Given a start spin configuration $S = \{S_0, S_1, \ldots, S_{N^2-1}\}$ we create a series of configurations by doing the following steps:

1. Choose one lattice site r (spin S_r) and calculate the energy difference between the trial[2] and the current configuration. In principle, we have to calculate the energy of the trial configuration using Eq. (5). However, our interactions are restricted to the first nearest neighbors. This allow to calculate the change in the total energy using the following equation:

$$\Delta H = 2JS_r\left(\sum_\beta S_\beta + B\right) \tag{9}$$

   Where the sum is over the four neighbors of atom r.
2. Accept the trial configuration as the new configuration with the probability:

$$P = \begin{cases} 1, & \text{if } \Delta H < 0 \\ \exp\left(-\dfrac{\Delta H}{k_B T}\right) & \text{otherwise} \end{cases} \tag{10}$$

Applying the Metropolis algorithm, we calculate the expectation values of an observable O by:

$$\langle O \rangle = \frac{1}{n_{\text{tot}}} \sum_{i=1}^{n_{\text{tot}}} O_i \tag{11}$$

Where O_i is the the observable measured on the i-th generated configuration and n_{tot} is the number of configurations.

The thermodynamic quantities that we will consider are the following:

---

[2] Trial configuration is the configuration that will emerge if we apply the flip change.

- **Energy:**

$$\langle E \rangle = \frac{1}{n_{\text{tot}}} \sum_{i=1,n_{\text{tot}}} H_i \tag{12}$$

- **Heat Capacity:**

$$c = \frac{\partial \langle E \rangle}{\partial T} = \frac{1}{k_B T^2}(\langle E^2 \rangle - \langle E \rangle^2) = \frac{1}{k_B T^2} \text{var}(E) \tag{13}$$
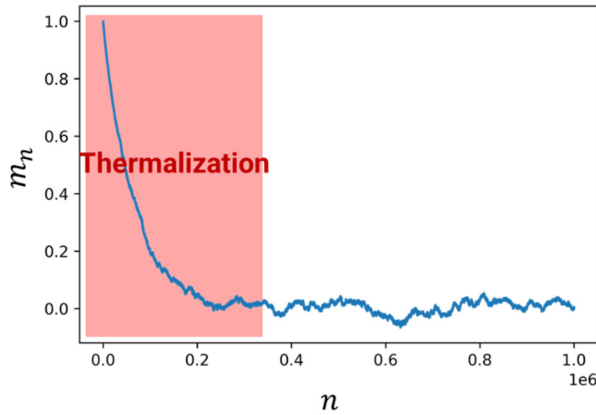
- **Magnetization:**

$$\langle M \rangle = \frac{1}{n_{\text{tot}}} \sum_{i=1,n_{\text{tot}}} M_i \, , M_i = \sum_k S_k \tag{14}$$

- **Susceptibility:**

$$\chi = \frac{\partial \langle M \rangle}{\partial B} = \frac{1}{k_B T}(\langle M^2 \rangle - \langle M \rangle^2) = \frac{1}{k_B T} \text{var}(M) \tag{15}$$

Thermalization

In principle the initial configuration in our calculations is far from thermodynamics equilibrium and the generated configurations are not representative. Therefore, the system needs time to thermalize before we calculate the observables. This is demonstrated in Fig. (6). The simulation began with a "cold start", where all spins were initially parallel. As a result, the value began at 1 and gradually decreased towards zero until it stabilized and fluctuated around this value, indicating that the system had reached equilibrium. The duration of this stabilization period is known as the thermalization time, and it represents the number of configurations required for the simulation to reach a state of equilibrium. It is recommended to only compute the expectation values of observables using configurations generated after the thermalization time.

Figure 6: Total magnetization per atom of the system as a function of time.

Tasks (20 pnts)

- Perform a Monte Carlo simulation of the 2D Ising model. Consider $J = 1$, and $B = 0$ and $B = 1$.
- Plot the expectation values of the energy and total magnetization per atom as well as of the heat capacity and susceptibility as a function of $k_B T$.
- Try to estimate $k_B T_C$. The value of the Curie Temperature increases or decreases when you switch on the external magnetic field? Can you explain your finding? Can you predict $k_B T_C$ when $B = -1$?
- Plot and explain the spin configuration maps for $T < T_C$, $T \approx T_C$, and $T > T_C$.

Optional Task

- Instead of $J = 1$ (ferromagnetic) set $J = -1$ (antiferromagnatic).

Arrays

Although the 2-index notation of spins, i.e., S_{ij}, is more intuitive, it is computationally more convenient to use a single index ("*super-index*") notation, i.e., S_α. This can be done by using two auxiliary matrices that perform the $\alpha \rightarrow \{i, j\}$ and $\{i, j\} \rightarrow \alpha$ mappings:

```
aux1: N²×2
aux1[α,0]=i
aux1[α,1]=j
α=0,…,N²-1

aux2: N×N
aux2[i,j]=α
i,j=0,…,N-1
α=0,…,N²-1
```

It will also be computationally efficient if we set an array that stores the indices of the four neighbors of each atom α:

```
nn: N²×4
If α is the atom at {i,j} then:
nn[α,0]=index of atom at {i-1,j}
nn[α,1]=index of atom at {i+1,j}
nn[α,2]=index of atom at {i,j-1}
nn[α,3]=index of atom at {i,j+1}
α=0,…,N²-1
```

Functions

We will need three functions:

Function for periodic boundary conditions:

```
per(i,N)=If(i>N-1) then per =0
        else if (i<0) then per =N-1
        else per=i
```

Function to calculate the total energy:

Let's call it `Hamiltonian`. It will take as input the exchange interaction `J`, the magnetic field `B`, the array of neighbors `nn`, and the array of spins `S`. It will provide as output the total energy, using Eq. (5). This function will be used only once to calculate the total energy of the initial spin configuration. The total energy for the other configurations will be updated by using the following function.

Function to calculate the total energy difference when a spin flips:

Let's call it `DeltaH`. It will take as input the exchange interaction `J`, the magnetic field `B`, the array of neighbors `nn`, and the array of spins `S` and the spin to be flipped `r` [see Eg. (9)]

Initial Spin Configuration

There are two possible initial spin configurations. The one is called as "*cold-start*" and the other as "*hot-start*".

In the cold-start all spins are parallel:
$S_\alpha=+1$ or $S_\alpha=-1$ for each $\alpha=0,\dots,N^2-1$.

In the hot-start the direction of the spins is set randomly:
`Sₐ=+1 if r<=0.5, otherwise Sₐ=-1, r random number between 0.0 and 1.0`

Observables

In order to calculate the expectation values of the observables [Eqs (8)-(11)] we need to calculate mean values. However, the new configurations are created from previous ones. Therefore, correlations between the values may be present. To avoid these, we do not include all configurations when we calculate the observables, but we skip `Nskip` configurations.

How to choose Spins to flip

There are three ways to choose the spins to flip:

1. Sequantial: We go through all lattice sites in order. Not recommended as the generated configurations will be highly correlated.
2. Random update: Choose a spin randomly. E.g., `r=random number bewteen 0 and N`^{2}`-1`. **This is recommended**.
3. Random update without doubling: We choose the spins randomly but ensure that a site is chosen only once within a lattice update (i.e., within N^2 updates). Not recommended, as it is computationally expensive.

For more information, it is strongly recommended to refer to the slides uploaded on the e-class platform