2D Ising Models Draft 1

Salar Ghaderi, Xinyu Liu, and Tongzhou Wang

October 22, 2024

Abstract

In this first draft, we complete the tasks in section **1. Prep work**, explain what periodic boundary conditions mean in section **2. Writing the code**, and interpret the physical meaning of the slope of the energy versus temperature plot, as required in section **3. Testing and running the code**.

1 Prep work

- Explain in words why you should find $M \sim 0$ for sufficiently high T, and why you should find an extremal M (all spins aligned) for sufficiently low T.

From a physics perspective, during any isothermal and isobaric processes, a system's change in its Gibbs free energy satisfies $dG \leq 0$, where the equality only holds for reversible processes, which are unrealistic idealizations. For isothermal processes we have $dG = dH - TdS$.

At high T, dG is dominated by $-TdS$, so the system must change in a way such that $dS > 0$. For our system, this means that there will be a mixture of spin up and spin down, so $M = \sum_i s_i \sim 0$.

At low T, dG is dominated by dH, so the system must change in a way such that $dH < 0$. For our system, the energy is lower when more spins are aligned because $E = \sum_{\langle ij \rangle} E_{\langle ij \rangle} = -J \sum_{\langle ij \rangle} s_i s_j$. Thus, the spins have a tendency to become all aligned, resulting in an extremal M.

From an algorithmic perspective, in the metropolis algorithm, we accept the move with acceptance probability ((10.60) in Newman [1])

$$P_a = \begin{cases} 1, & \Delta E \leq 0, \\ e^{-\beta \Delta E}, & \Delta E > 0. \end{cases} \tag{1}$$

At high T, $\beta \to 0$, (1) becomes

$$P_a \approx \begin{cases} 1, & \Delta E \leq 0, \\ 1, & \Delta E > 0. \end{cases} \tag{2}$$

This means that the move will be accepted no matter it decreases the energy (aligns more spins) or increases the energy (reverses more spins). The upshot will be an equal amount of spins in both directions, so $M = \sum_i s_i \sim 0$.

At low T, $\beta \to \infty$, (1) becomes

$$P_a \approx \begin{cases} 1, & \Delta E \leq 0, \\ 0, & \Delta E > 0. \end{cases} \tag{3}$$

This means that the move will be accepted only if it decreases the energy (aligns more spins) or doesn't change the energy. The upshot will be that all spins are aligned, resulting in an extremal M.

■ Read and briefly explain the Metropolis algorithm described in Newman section 10.3.2.

The Metropolis Algorithm

The Metropolis algorithm is a sampling technique that belongs to the benefits Markov Chain Monte Carlo . It is generaly used when direct sampling from a desired probability distribution is not feasible due to the complexity of the system being modeled. The algorithm is particularly valuable in simulations of physical systems in thermal equilibrium, where system configurations follow the Boltzmann distribution.

In his book, Newman introduces the idea of Markov processes, stating:

"In the vast sweep of nature's processes, there are many in which the future is predictable not from the distant past, but only from the present state of affairs. This kind of memoryless system is known as a Markov process." [2]

This memoryless property is fundamental to the Metropolis algorithm, which generates a Markov chain of successive states. By occasionally accepting transitions to higher-energy states, the algorithm avoids local minima and samples the correct thermodynamic properties of the system.

Simulation for Equilibrium

The objective of the Metropolis algorithm is to simulate a system at thermal equilibrium, where the states are distributed according to the Boltzmann distribution:

$$P(\mathbf{x}) \propto \exp\left(-\frac{E(\mathbf{x})}{k_B T}\right)$$

Where:

- $E(\mathbf{x})$ represents the energy of the system in state $\mathbf{x}$,
- k_B is the Boltzmann constant,
- T is the system's temperature.

The algorithm aims to generate a series of configurations that follow this distribution, with lower-energy states being more likely but allowing occasional exploration of higher-energy states to ensure the full state space is sampled.

Markov Chains and Importance Sampling

The Metropolis algorithm constructs a Markov chain where each state depends only on the previous one. As the algorithm iterates, the chain converges to the Boltzmann distribution. Instead of uniformly sampling configurations, the algorithm uses importance sampling, where more probable (lower-energy) states are favored, improving the overall efficiency of the simulation.

Steps of the Metropolis Algorithm

1. Initial Configuration: Begin with a random initial configuration of the system, $\mathbf{x}_0$.

2. Initiating a Change: Propose a small random modification to the current configuration, generating a new configuration, $\mathbf{x}_1$. For example, in a system of particles, this could involve changing the position of one particle.

3. Compute the Energy Difference: Calculate the energy difference between the new and current configurations:

$$\Delta E = E(\mathbf{x}_1) - E(\mathbf{x}_0)$$

4. Acceptance guide: Determine whether to accept the new configuration:

- If $\Delta E \leq 0$, accept the new state.
- If $\Delta E > 0$, accept the new state with probability:

$$P(\text{accept}) = \exp\left(-\frac{\Delta E}{k_B T}\right)$$

  A random number r between 0 and 1 is drawn, and the new configuration is accepted if $r < \exp\left(-\frac{\Delta E}{k_B T}\right)$.

5. Update the configuration: If the new configuration is accepted, set $\mathbf{x}_0 = \mathbf{x}_1$; otherwise, retain the current configuration.

6. Repeat the Process: Repeat steps 2–5 for many iterations. Over time, the sequence of configurations will approximate the Boltzmann distribution.

Key aspect of exploration in the algorithm

The Metropolis algorithm efficiently balances exploration and exploitation. By occasionally accepting higher-energy configurations, it prevents the system from getting stuck in local energy minima. This allows the system to reach thermal equilibrium, where the states it samples reflect the correct thermodynamic properties.

Convergence and Thermal Equilibrium

- After a sufficient number of iterations, the system reaches equilibrium, and the generated states follow the Boltzmann distribution.
- Once equilibrium is achieved, the algorithm continues to sample configurations, which can be used to compute physical observables such as average energy or magnetization (in models like the Ising model).

Dimensionless physical constant

■ Rescale the problem so that you only need one physical constant (which will be a combination of J, k, and T).

In the metropolis algorithm, in the Markov chain, when the initial energy is E and the final energy is E', the acceptance function is:

$$f(E, E') = \begin{cases} 1, & \text{if } E \leq E' \\ e^{-(E'-E)/k_B T}, & \text{if } E' > E \end{cases}$$

We know that the only dimensionless factor relevant to our proposal is:

$$\lambda = \frac{\Delta E}{k_B T} \tag{4}$$

We also know that, for our system which is a 2D Ising model, with the Hamiltonian:

$$H = \sum_{i,j} J s_i s_j \tag{5}$$

Where the i and j is summed when they are neighbours. We know that in metropolis algorithm, each step only one spin is flipped(or not flipped). There's a total number of 4 neighbours for each spin and we assume that there are $n_\uparrow$ number of neighbours pointing up and $n_\downarrow$ number of neighbours pointing down. Then the energy change with a spin flipped will be:

Table 1: Energy changing under flipping a spin

$n_\uparrow$ and $n_\downarrow$	40	31	22	13	04
ΔE	$\pm 8J$	$\pm 4J$	0	$\pm 4J$	$\pm 8J$

We can see that in any case the shift of energy will only be a integer number of 4J. So that the only important physical constant will be:

$$\lambda = \frac{4J}{k_B T} \tag{6}$$

2 Writing the code

■ Write into code the Metropolis algorithm found in section 10.3.2 of Newman. Note that you do not need to recalculate the full energy of the system at each trial, since most of it is unchanged (i.e. you only need the terms involving the lattice atom you picked). Use **periodic boundary conditions**.

Before starting working on the simulation code, we'd like to make some comment on better understanding the periodic boundary condition. This boundary condition might seems non-physical from the beginning, since there's no reason to assume that the infinitely large system is periodical.

However, in a limited amount of time, we cannot do the measurement on an infinitely large system, so what we simulate is always a small subsystem. In order to minimize the finiteness of

our system, the best way is to assume the periodic boundary condition, so that we're effectively simulating an infinitely large system with periodicity. In another word, we're measuring a 2D Ising model in a 2-torus. As long as the relevant physics is local and there's no topological effect, we can believe that the physics is similar to a 2D flat space.

3 Testing and running the code

- What is the physical meaning of the slope of the energy versus temperature? Can you describe in words what this slope means in terms of how much work it is to raise the temperature?

The slope of the energy versus temperature plot is the heat capacity at constant volume C_V, because $C_V \equiv (\frac{đQ}{dT})_V = (\frac{\partial U}{\partial T})_V$, where U is the internal energy. The slope represents the energy required for raising the temperature by an infinitesimal amount.

References

[1] M. Newman. *Computational Physics*, chapter 10.3. CreateSpace Independent Publishing Platform, 2013.

[2] M. E. J. Newman and G. T. Barkema. *Monte Carlo Methods in Statistical Physics*. Oxford University Press, 1999.