

2D Ising Model Simulation in R

RJ Cass

BYU

5 Dec. 2025

Outline

1 Context

2 Mathematics

3 Process

4 Results

5 Conclusion

6 Appendix

What is an Ising model?

- Array of nodes where each node can be in one of two states (usually represented as +1 and -1)
- Core concept is that each node is influenced by its neighbors, and larger system-wide factors
- Used for modeling in a variety of fields, such as material sciences, epidemiology, even sociology
- As N grows, quickly expands past the bounds of analytical solutions

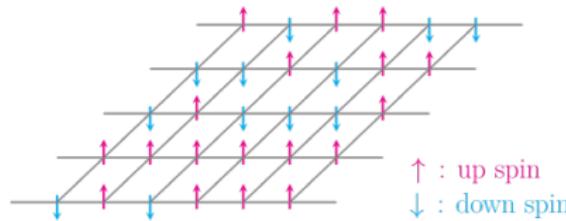


Figure: Representation of a 2D Ising Model [1].

Ferromagnetism

In ferromagnets, each dipole (node) has a spin (magnetic moment: positive/negative). This alignment is affected by two factors:

- Orientation of Neighbors: dipoles tend to align with their neighbors to reduce entropy of system
- Temperature: the external energy from temperature can excite the dipoles and cause their spin to flip spontaneously
 - There is a Curie Temperature (T_C) above which the energy of the system overwhelms the 'neighborly influence', resulting in random alignment

The behavior of a ferromagnetic material can be solved analytically in relatively simple scenarios, but becomes virtually unsolvable for more complex scenarios.

Goal/Approach

Goal: We want to better understand the properties of ferromagnetic materials near the Curie Temperature.

Approach: We will use Markov chain Monte Carlo to imitate a ferromagnetic material's behavior over a range of temperatures and investigate its properties.

Note: For this project I am not using actual scientific values (ie. scaling things differently than real-world scenarios). This is not a rigorous study of a real scenario: it's more like a proof of concept of the process.

Node Equations

Hamiltonian [2, Eq. 2] - influence of the neighbors on a node

$$H_i = -J \sum_{j=1}^4 s_i s_j \quad (1)$$

- J : coupling constant (+1: ferromagnetic, -1: antiferromagnetic)
- s_i : spin at the node of interest
- s_j : spin of the nearest neighbors (in 2D there are 4: above, below, left, right)

Metropolis Ratio [2, Eq. 12] - influence of temperature and neighbors

$$\frac{W(x_i \rightarrow x_{i'})}{W(x_{i'} \rightarrow x_i)} = \begin{cases} e^{-\delta H_i \beta}, & \delta H_i < 0 \\ 1, & \text{otherwise} \end{cases} \quad (2)$$

- $\delta H_i \geq 0 \implies e^{-\delta H_i \beta} \leq 1$, hence the acceptance ratio is just 1
- $\beta = \frac{1}{k_b T}$, k_b is Boltzman's constant, T is Temperature. For this project I am using $k_b = 1$ and $1 \leq T \leq 5$ (unitless)

Lattice Equations

Energy [2, Eq. 18]

$$\langle E \rangle = \frac{1}{2} \sum_{i=1}^N H_i \quad (3)$$

Magnetization [2, Eq. 17]

$$\langle M \rangle = \frac{1}{N} \sum_{i=1}^N s_i \quad (4)$$

Specific Heat [2, Eq. 19] - How much energy is required to make the elements more 'active'

$$C = \frac{\beta}{T} (\langle E^2 \rangle - \langle E \rangle^2) \quad (5)$$

Susceptibility [2, Eq. 20] - How responsive the material is to an applied magnetic field

$$\chi = \beta (\langle M^2 \rangle - \langle M \rangle^2) \quad (6)$$

Process Outline

One MCMC step requires, for each element in a given lattice:

- ① Determine the Hamiltonian for that element as shown in Equation (1)
- ② Calculate the acceptance ratio shown in Equation (2)
- ③ Flip the spin ($s_i = -s_i$) if $\text{runif}(1)$ is less than the acceptance ratio

The overall process is:

- ① Get initial temperature, initialize lattice (random node alignments)
- ② Perform a number of MCMC steps until the lattice reaches equilibrium (effectively the burn-in)
- ③ Perform a number of MCMC steps, recording the Energy and Magnetization values at each step
- ④ Calculate $\langle E \rangle$, $\langle M \rangle$, C , χ , MCE for that temperature value
- ⑤ Repeat steps 1-4 for each temperature value

Note: One MCMC 'unit' is the full lattice state, which encapsulates the arrangement of all the nodes

Brief Coding Overview

I attempted to make my code reusable, defining functions with sensible parameters that could easily be adapted. The Python code provided by Singh [3] offered a nice framework as a launching point for setup.

- ① `create_array()`: Generates the initial lattice
- ② `calc_ham()`: Calculates the Hamiltonian for a given node
- ③ `calc_energy()`: Calculates the energy of the lattice
- ④ `calc_mag()`: Calculates the magnetization of the lattice
- ⑤ `mc_step()`: Performs one MCMC step (updates lattice to next state)
- ⑥ `ising_sim()`: For a given temperature sequence, performs MCMC simulation at each temperature and records the metrics at that temperature
- ⑦ `ising_sim_plots()`: For a given temperature, perform MCMC simulation and plot the lattice at certain points (for display purposes)
- ⑧ `ising_sim_mag()`: For a given temperature, perform MCMC simulation and plot the magnetization per step

Temperature Distributions

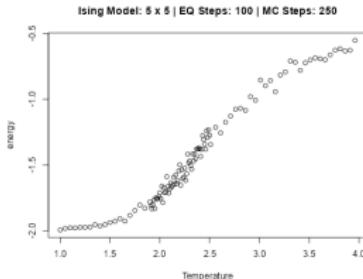


Figure: Energy

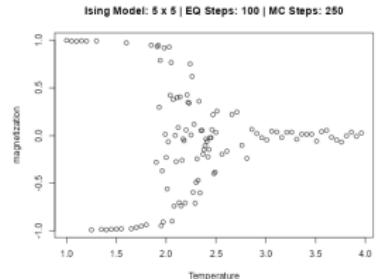


Figure: Magnetization

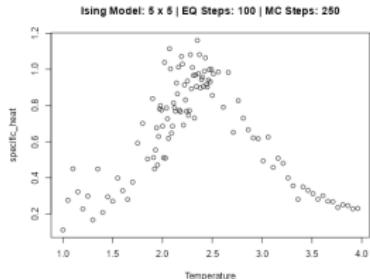


Figure: Specific Heat

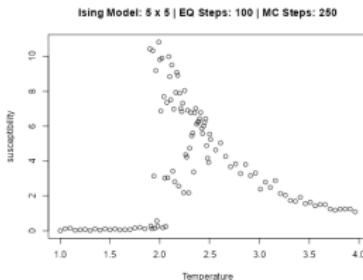


Figure: Susceptibility

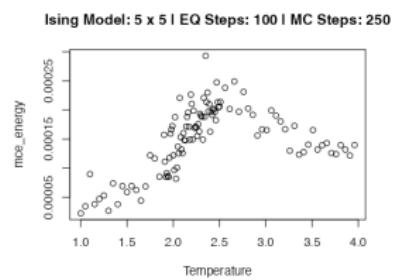


Figure: MCE Energy

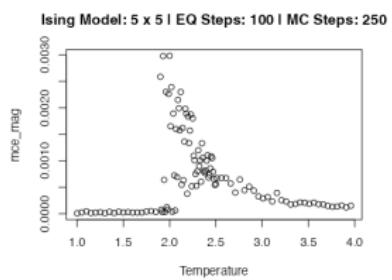


Figure: MCE Mag.

Behavior at different temperatures

Figure: Temperature =
1.5

Figure: Temperature =
2

Figure: Temperature =
4

Magnetization Phase Change

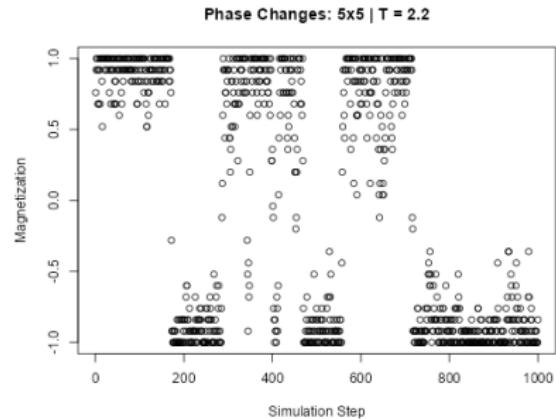


Figure: Magnetization at each simulation step showing spontaneous phase change

Figure: Animation showing spontaneous phase change

Next Steps - Code

There are a few adjustments that would be good for making my code more re-usable

- Make the function used for the Metropolis Ratio dynamic, allowing a user to define their own and pass it in
- Make the code for plot generation more robust (for both the final MCMC plots, and the lattice plots)
- Optimize, find a way to make it run fast enough to operate on grids that are magnitudes larger than what I tested

There are a few things that, using what I currently have, would be interesting next steps

- Adapt the code to use actual material properties (ie. use real value of k_b , T , etc.)
- This can be expanded into 3D applications generally (the Metropolis ratio is much more complex in my scenario, but possible)
- An actual field of study: how to make this work not just looking at nearest neighbor, but have the Hamiltonian depend on full material (Dr. Page)

Bibliography

- [1] Ta2o, *Ising model*, Sep. 2025. [Online]. Available:
https://en.wikipedia.org/wiki/Ising_model#/media/File:2D_ising_model_on_lattice.svg
- [2] J. Kotze, “Introduction to monte carlo methods for an ising model of a ferromagnet,” *arXiv:0803.0217v1 [cond-mat.stat-mech]*, 2008.
- [3] R. Singh, *Ising model*, [Online]. Available:
<https://rajeshrinet.github.io/blog/2014/ising-model/>