Computational Simulation of Critical Phenomena in the 2D Ising Model: From Statistical Mechanics to Energy-Based AI Architectures

Yehia Saeed Gewily

Alexandria University, Dept. of Electronics and Communications Engineering

Abstract

This research details the design and implementation of a high-performance 2D Ising Model simulation, serving as Phase 1 of a broader study into Energy-Based Models (EBMs). We employ the Metropolis-Hastings Markov Chain Monte Carlo (MCMC) algorithm to investigate the thermodynamic evolution of a ferromagnetic lattice. To overcome the computational overhead of Pythonic loops, a vectorized checkerboard decomposition was implemented, enabling real-time simulation of 256×256 grids. We provide quantitative evidence of the second-order phase transition at $T_c \approx 2.269$ by measuring specific heat (C_v) and magnetic susceptibility (χ). The study concludes by establishing a formal isomorphism between the Ising Hamiltonian and the loss functions of associative memory networks, providing a bridge between 20th-century statistical physics and modern generative AI.

1 Introduction

The Ising model is a mathematical abstraction that describes the transition between ordered and disordered states in ferromagnetic materials. While the 1D model (Ising, 1925) failed to exhibit a phase transition, Lars Onsager's 1944 solution for the 2D lattice revealed a non-trivial critical temperature T_c, marking the onset of spontaneous magnetization.

In the context of modern Artificial Intelligence, the Ising model is more than a physics simulation; it is the foundational **Energy-Based Model (EBM)**. By interpreting spins as binary neurons and the exchange interaction as synaptic weights, we can map the physical process of energy minimization directly to the training and inference cycles of neural networks like Boltzmann Machines and Hopfield Networks.

2 Theoretical Framework

2.1 The Hamiltonian and Interaction

We define a 2D square lattice Λ of $N = L \times L$ sites. Each site hosts a discrete spin variable $\sigma_i \in \{+1, -1\}$. The configuration space Ω contains 2^N possible states. The energy of any configuration σ is defined by the Hamiltonian:

$$\mathcal{H}(\sigma) = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j - B \sum_i \sigma_i \tag{1}$$

where J is the exchange coupling constant. For $J > 0$, the system is *ferromagnetic*, favoring parallel alignment. B is the external magnetic field which breaks the $\mathbb{Z}_2$ symmetry of the system.

2.2 Thermodynamic Equilibrium

The probability of a state σ follows the Boltzmann distribution:

$$P(\sigma) = \frac{e^{-\beta \mathcal{H}(\sigma)}}{Z}, \quad Z = \sum_{\sigma \in \Omega} e^{-\beta \mathcal{H}(\sigma)} \tag{2}$$

where $\beta = (k_B T)^{-1}$ is the inverse temperature. As $T \to 0$, the system collapses into its ground state (minimum energy), whereas $T \to \infty$ leads to a maximum entropy state (random noise).

3 Computational Methodology

3.1 Metropolis-Hastings Sampling

Because the partition function Z involves 2^N terms, direct calculation is intractable for $L = 256$. We use the Metropolis algorithm to generate a Markov Chain that converges to the Boltzmann distribution. The acceptance ratio $A(\sigma \to \sigma')$ for a single spin-flip is:

$$A = \min(1, \exp(-\beta \Delta E)) \tag{3}$$

where $\Delta E = 2\sigma_i(J \sum \sigma_{neighbor} + B)$.

3.2 Vectorized Checkerboard Decomposition

To achieve high-performance updates, we exploit the **conditional independence** of the lattice. Since only nearest-neighbors interact, the lattice can be bipartitioned into "red" and "black" sub-lattices.

Given the state of all "black" sites, all "red" sites are mutually independent. This allows for **SIMD vectorization** using NumPy, updating $N/2$ spins in a single clock cycle.

```
def step(grid, T, J, B):
    # Parity masks for checkerboard
    for parity in [0, 1]:
        # Compute neighbor sum using periodic BCs
        neighbors = np.roll(grid, 1, 0) + np.roll
            (grid, -1, 0) + \
                    np.roll(grid, 1, 1) + np.roll
                        (grid, -1, 1)

        # Calculate Delta E for all sites in
            parity
        dE = 2 * grid * (J * neighbors + B)

        # Acceptance logic
        prob = np.exp(-dE / T)
        accept = (np.random.rand(*grid.shape) <
            prob) & mask[parity]
        grid[accept] *= -1
```

Listing 1: Vectorized Ising Core Implementation

4 Quantitative Analysis

We monitor two primary response functions derived from the fluctuation-dissipation theorem.

4.1 Magnetic Susceptibility (χ)

χ measures the system's sensitivity to an external field. It is proportional to the variance of magnetization:

$$\chi = \frac{\beta}{N} \left(\langle M^2 \rangle - \langle M \rangle^2 \right) \tag{4}$$

4.2 Specific Heat (C_v)

C_v represents the fluctuations in internal energy:

$$C_v = \frac{\beta^2}{N} \left(\langle E^2 \rangle - \langle E \rangle^2 \right) \tag{5}$$

At $T = T_c$, both χ and C_v exhibit a power-law divergence in the thermodynamic limit ($L \to \infty$), which we observe as sharp peaks in our finite-size simulation.

5 Simulation Results

- **Phase Transition:** At $T \approx 2.27$, we observe *Critical Opalescence.* The correlation length ξ grows, and clusters of all scales appear.
- **Hysteresis:** Below T_c, the system exhibits memory. Sweeping the field B from positive to negative reveals a lag in the magnetization response, a property we intend to exploit for associative memory in Phase 2.
- **Performance:** The vectorized approach maintained 60 FPS on a 256 × 256 grid, allowing for immediate visual feedback of thermodynamic annealing.

6 Conclusion: The AI Convergence

Phase 1 proves that a simple local update rule leads to complex global intelligence (symmetry breaking). In **Phase 2**, we will replace the constant J with a learnable weight matrix W_{ij} derived from the Hebbian rule:

$$W_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i^\mu \xi_j^\mu \tag{6}$$

where ξ^μ are target images. This transforms the Ising model into a **Hopfield Network**, where the "equilibrium states" of the physics simulation correspond to "recalled memories" in a software system.

References

[1] Onsager, L. (1944). Crystal Statistics. I. *Physical Review.*

[2] Metropolis, N. (1953). *Journal of Chemical Physics.*

[3] Ackley, D. H., Hinton, G. E. (1985). A Learning Algorithm for Boltzmann Machines. *Cognitive Science.*