

# Modelo de Ising

O modelo de Ising constitui um dos modelos mais importantes da Mecânica Estatística, sendo amplamente utilizado no estudo de fenômenos coletivos e transições de fase. Originalmente proposto para descrever propriedades ferromagnéticas de materiais, o modelo tornou-se um paradigma fundamental para a compreensão de sistemas complexos compostos por muitas partículas interagentes.

Considere uma rede bidimensional quadrada de dimensão  $n \times n$ , composta por  $N = n^2$  sítios. Em cada sítio  $i$  da rede associa-se uma variável aleatória denominada *spin*, representada por

$$\sigma_i \in \{-1, +1\},$$

em que:

- $\sigma_i = +1$  representa um spin orientado para cima (*up spin*);
- $\sigma_i = -1$  representa um spin orientado para baixo (*down spin*).

Uma configuração do sistema é dada pelo vetor

$$\sigma = (\sigma_1, \sigma_2, \dots, \sigma_N),$$

o qual descreve simultaneamente a orientação de todos os spins da rede.

Fisicamente, cada spin interage apenas com seus vizinhos mais próximos. A energia total associada a uma configuração  $\sigma$  é descrita pela Hamiltoniana do modelo de Ising:

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

em que:

- $J$  representa a constante de interação entre spins vizinhos;
- $\langle i, j \rangle$  indica que a soma é realizada sobre pares de vizinhos adjacentes;
- $h$  representa um campo magnético externo aplicado ao sistema.

Quando  $J > 0$ , o sistema favorece configurações em que spins vizinhos apresentam a mesma orientação, caracterizando o comportamento ferromagnético. Nesse caso, estados alinhados minimizam a energia do sistema.

A distribuição de probabilidade associada às configurações do sistema é dada pela distribuição de Gibbs-Boltzmann:

$$P_{\beta}(\sigma) = \frac{e^{-\beta H(\sigma)}}{Z},$$

em que:

$$\beta = \frac{1}{k_B T},$$

sendo  $k_B$  a constante de Boltzmann e  $T$  a temperatura absoluta do sistema. O termo

$$Z = \sum_{\sigma \in \Sigma} e^{-\beta H(\sigma)}$$

é denominado função de partição e atua como constante normalizadora da distribuição.

A temperatura exerce papel fundamental na dinâmica do modelo. Para temperaturas elevadas, a agitação térmica domina as interações locais, produzindo configurações desordenadas. Em contrapartida, para temperaturas baixas, o sistema tende espontaneamente a estados ordenados, nos quais há predominância de spins alinhados.

Um dos aspectos mais importantes do modelo de Ising é a existência de uma transição de fase. No caso bidimensional sem campo magnético externo ( $h = 0$ ), existe uma temperatura crítica  $T_c$  dada por

$$T_c = \frac{2J}{k_B \ln(1 + \sqrt{2})}.$$

Abaixo dessa temperatura, o sistema apresenta magnetização espontânea; acima dela, o sistema perde ordenamento magnético devido ao aumento da desordem térmica.

A solução analítica exata do modelo de Ising bidimensional sem campo externo foi obtida por Lars Onsager em 1944, constituindo um marco histórico da Física Estatística moderna. Entretanto, em dimensões superiores, soluções fechadas gerais ainda permanecem desconhecidas.

Do ponto de vista computacional, a principal dificuldade do modelo reside no cálculo da função de partição  $Z$ , cujo custo computacional cresce exponencialmente com o tamanho da rede. Dessa forma, técnicas de simulação estocástica tornam-se essenciais. Entre os métodos mais utilizados destacam-se:

- algoritmo de Metropolis-Hastings;
- dinâmica de Glauber;
- algoritmos de Wolff e Swendsen-Wang.

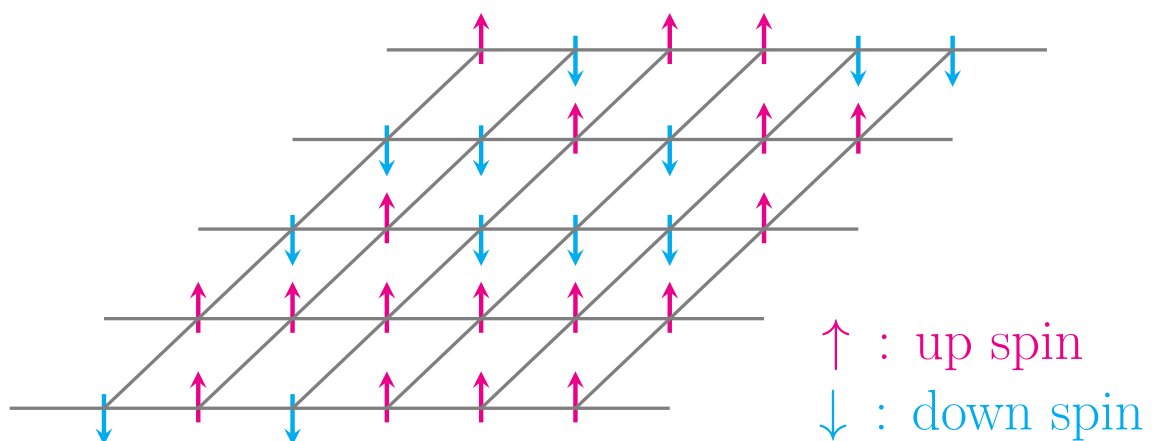
Esses métodos permitem gerar amostras de configurações cuja distribuição estacionária coincide com a distribuição de Gibbs do modelo.

Atualmente, o modelo de Ising transcende aplicações em magnetismo, sendo empregado em diversas áreas do conhecimento, tais como:

- redes neurais probabilísticas;
- aprendizado de máquina;
- otimização combinatória;
- biologia computacional;
- dinâmica social e formação de opinião;
- teoria da informação;
- sistemas complexos e criticalidade.

Sua importância decorre do fato de que regras locais extremamente simples são capazes de produzir comportamentos macroscópicos complexos, tornando o modelo de Ising um dos exemplos mais clássicos de emergência em sistemas físicos e estatísticos.

Figura 1: Modelo de Ising bidimensional mostrado como uma rede de spins interativos.



**Fonte:** Por Ta2o - Obra do próprio, CC BY 4.0, <<https://commons.wikimedia.org/w/index.php?curid=161501738>>.

# Implementação Computacional e Otimização de Performance

Apesar da simplicidade das regras locais descritas pela Hamiltoniana do sistema, a simulação estocástica do modelo de Ising apresenta desafios computacionais significativos, especialmente quando se deseja observar a termodinâmica do sistema em tempo real e em grades de alta resolução.

## Gargalos em Linguagens Interpretadas

A amostragem de configurações via método de Metropolis-Hastings exige iterações na ordem de  $O(N \times S)$ , onde  $N$  é o número de sítios da malha e  $S$  é a quantidade de passos de tempo (*sweeps*). Em linguagens interpretadas como Python ou R, o uso de laços de repetição (*loops for*) para iterar sobre cada spin individualmente é intrinsecamente ineficiente.

Além disso, o cálculo da variação de energia ( $\Delta E$ ) requer a verificação de condições de contorno (bordas da malha). Abordagens ingênuas utilizando declarações condicionais (*if-else*) dentro do laço principal forçam ramificações de código (*branching*) que interrompem a previsão de instruções da CPU (*pipeline stall*), degradando substancialmente a performance.

## Arquitetura Proposta: Compilação *Just-In-Time* (JIT)

Para viabilizar simulações de milhões de iterações por segundo, adotamos uma arquitetura baseada na compilação JIT, utilizando a biblioteca Numba no ecossistema Python. A principal função de atualização do estado da malha foi decorada com a diretiva `@njit`, o que permite:

1. **Desvio do Global Interpreter Lock (GIL):** O código compilado executa em nível de máquina (C/C++), isolado do *overhead* do interpretador Python.
2. **Condições de Contorno Aritméticas:** A validação das bordas, necessária para simular um toro bidimensional (condições de contorno periódicas), foi substituída pelo operador de resto de divisão (*módulo aritmético*). Por exemplo, o vizinho à esquerda de um sítio na coluna  $j$  de uma grade de tamanho  $n$  é computado como  $(j - 1) \% n$ . Essa operação escalar ocorre em um único ciclo de *clock*, evitando ramificações lógicas.

## Vetorização da Renderização Visual

A observação empírica da transição de fase exige uma taxa de quadros (*frames per second* - FPS) fluida. Bibliotecas gráficas tradicionais (como `matplotlib` ou `ggplot2`) constroem objetos gráficos complexos a cada quadro, o que se torna o gargalo limitante da simulação iterativa.

A solução implementada utiliza **Manipulação Matricial Direta**. O estado da malha, representado por uma matriz de inteiros  $\{-1, 1\}$ , é convertido diretamente para um tensor numérico de três canais (*RGB arrays* de 8-bits) via indexação Booleana (*masking*) com a biblioteca `numpy`.

### Exemplo de vetorização visual sem overhead gráfico.

```
def render_grid_to_rgb(grid):
    # Inicializa tensor RGB
    rgb_img = np.zeros((grid.shape[0], grid.shape[1], 3),
                      dtype=np.uint8)

    # Mapeamento de cores vetorizado O(1)
    rgb_img[grid == 1] = [220, 50, 47] # Spin Up (
    Vermelho)
    rgb_img[grid == -1] = [38, 139, 210] # Spin Down (Azul)

    return rgb_img
```

Esse tensor numérico bruto é então ejetado diretamente para o *frontend* (`Streamlit`), permitindo visualizar a termalização do sistema, o surgimento de domínios magnéticos fractais próximos a  $T_c$ , e o efeito de *Critical Slowing Down* sem comprometer a latência da interface.