



DOCHMO ANALYTICS

RELATÓRIO TÉCNICO DE ESTATÍSTICA APLICADA

O MODELO ISING 2D: SIMULAÇÃO ESTOCÁSTICA E OTIMIZAÇÃO DE PERFORMANCE

Da Fundamentação Teórica à Aplicação Interativa

DOUGLAS CHAVES MOURA

FORTALEZA - CE

JULHO DE 2026

VERSÃO 1.0

DOCHMO REPORT SERIES

Nº 002/2026

Douglas Chaves Moura

O MODELO ISING 2D: SIMULAÇÃO ESTOCÁSTICA E OTIMIZAÇÃO DE PERFORMANCE

Da Fundamentação Teórica à Aplicação Interativa

Relatório técnico especializado em Física Estatística computacional, otimização de performance e comunicação científica aplicada. Desenvolvido como estudo aplicado de simulação estocástica do Modelo Ising bidimensional, abordando compiladores Just-in-Time (JIT), condições de contorno periódicas, vetorização gráfica, implantação de aplicação interativa e validação empírica qualitativa dos regimes térmicos.

Fortaleza - Ceará

2026

RESUMO

Este trabalho apresenta o estudo e a implementação computacional do modelo de Ising bidimensional (2D) em uma rede quadrada sob condições de contorno periódicas, constituindo um paradigma fundamental na mecânica estatística para a análise de transições de fase e fenômenos críticos. Discute-se a formulação matemática do modelo por meio de sua Hamiltoniana, as interações ferromagnéticas locais ($J > 0$) e a distribuição de probabilidade de Gibbs-Boltzmann, cuja dinâmica térmica exibe uma temperatura crítica exata T_c dada pela clássica solução analítica de Lars Onsager (1944). Do ponto de vista prático, simulações estocásticas de Monte Carlo são essenciais para estimar as propriedades macroscópicas devido ao custo exponencial do cálculo da função de partição. Os gargalos de desempenho em linguagens interpretadas são analisados, e propõe-se uma arquitetura computacional baseada em compilação *Just-In-Time* (JIT) com a biblioteca Numba no ecossistema Python, eliminando o overhead do interpretador CPython. A otimização aritmética das condições de contorno periódicas evita ramificações lógicas (*branching*) ineficientes na CPU. Para a renderização visual em tempo real, implementou-se uma técnica de vetorização de mapeamento de cores via Numpy acoplada a uma interface Streamlit. A aplicação correspondente foi disponibilizada na Streamlit Community Cloud, viabilizando a exploração pública e interativa do modelo. Com isso, os três regimes termodinâmicos — desordem, criticalidade e ordem — foram validados qualitativamente por meio de observação visual direta, permitindo a análise do fenômeno de desaceleração crítica (*Critical Slowing Down*) e da geometria fractal dos domínios magnéticos na vizinhança de T_c .

Palavras-chave: Modelo de Ising. Simulação Estocástica. Compilação JIT. Numba. Transição de Fase.

ABSTRACT

This work presents the study and computational implementation of the two-dimensional (2D) Ising model on a square lattice under periodic boundary conditions, representing a fundamental paradigm in statistical mechanics for the analysis of phase transitions and critical phenomena. The mathematical formulation of the model is discussed through its Hamiltonian, local ferromagnetic interactions ($J > 0$), and the Gibbs-Boltzmann probability distribution, whose thermal dynamics exhibit an exact critical temperature T_c derived from Lars Onsager's classic analytical solution (1944). From a practical standpoint, stochastic Monte Carlo simulations are essential for estimating macroscopic properties due to the exponential cost of calculating the partition function. The performance bottlenecks of interpreted languages are addressed, and a computational architecture is proposed based on *Just-in-Time* (JIT) compilation using Numba in the Python ecosystem, eliminating the CPython interpreter overhead. Arithmetic optimization of periodic boundary conditions avoids inefficient branching logic on the CPU. For real-time visual rendering, a vectorized color mapping technique via Numpy was coupled with a Streamlit interface. The corresponding application was deployed on the Streamlit Community Cloud, enabling public and interactive exploration of the model. Consequently, the three thermodynamic regimes — disorder, criticality, and order — were qualitatively validated through direct visual observation, allowing the analysis of the critical slowing down phenomenon and the fractal geometry of magnetic domains near T_c .

Keywords: Ising Model. Stochastic Simulation. JIT Compilation. Numba. Phase Transition.

LISTA DE FIGURAS

1.1	Modelo de Ising bidimensional - 2D.	4
3.1	Interface da aplicação interativa do Modelo de Ising 2D.	12
3.2	QR Code de acesso ao app - Streamlit Community Cloud.	12
4.1	Regime de alta temperatura (desordem) do Modelo de Ising 2D ($\beta = 0,10$). . .	15
4.2	Regime crítico (transição de fase) do Modelo de Ising 2D ($\beta = 0,44$).	15
4.3	Regime de baixa temperatura (ordem) do Modelo de Ising 2D ($\beta = 1,50$). . . .	16

LISTA DE TABELAS

3.1	Parâmetros de controle da simulação interativa.	11
4.1	Correspondência entre comportamento observado e previsão teórica.	17
A.1	Configurações sugeridas para exploração dos regimes térmicos.	22

SUMÁRIO

Resumo

Abstract

Lista de Figuras

Lista de Tabelas

1	Modelo de Ising	1
1.1	Origem Histórica	1
1.2	Contextualização	2
1.3	Regra de Aceitação de Metropolis-Hastings	4
1.4	Grandezas Termodinâmicas Observáveis	5
1.5	Convergência da Cadeia de Markov: Balanço Detalhado e Ergodicidade . . .	5
2	Implementação Computacional e Otimização de Performance	7
2.1	Gargalos em Linguagens Interpretadas	7
2.2	Arquitetura Proposta: Compilação <i>Just-In-Time</i> (JIT)	7
2.3	Localidade de Cache e Estratégia de Amostragem	8
2.4	Vetorização da Renderização Visual	9
3	Aplicação Interativa e Portfólio	10
3.1	Visão Geral da Plataforma	10
3.2	Interface e Controles da Aplicação	11
3.3	Acesso ao Aplicativo	12
3.4	Integração com o Portfólio	13
4	Resultados e Validação Empírica	14
4.1	Regimes de Temperatura e Comportamento Qualitativo	14
4.2	Temperatura Crítica e Parâmetro de Ordem	16
4.3	Desaceleração Crítico (<i>Critical Slowing Down</i>)	17
4.4	Domínios Fractais e Criticalidade	18
	Referências Bibliográficas	18

Sumário

A	Acesso ao Aplicativo Interativo	21
A.1	Endereço do Aplicativo	21
A.2	Como Interagir com a Simulação	21
A.3	Sugestões de Experimentos	22
A.4	Código-Fonte e Repositório	22

CAPÍTULO 1

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.

1.1 Origem Histórica

O modelo de Ising possui uma história intelectual singular, marcada por uma série de contribuições que se estenderam por mais de duas décadas antes que seu comportamento em duas dimensões fosse plenamente compreendido.

Em 1920, o físico alemão Wilhelm Lenz (LENZ, 1920) propôs, como exercício para seu aluno de doutorado, um modelo simplificado de ferromagnetismo no qual variáveis binárias dispostas em uma rede interagem apenas com seus vizinhos imediatos. A tarefa de analisar esse modelo coube a Ernst Ising, que em 1925 apresentou sua tese de doutorado (ISING, 1925) resolvendo o caso unidimensional (cadeia linear de spins) de forma exata. A solução em 1D revelou ausência de transição de fase a qualquer temperatura finita, levando Ising a concluir, equivocadamente, que o mesmo se aplicaria a dimensões superiores. Essa conclusão freou o interesse no modelo por mais de uma década.

Foi Rudolf Peierls quem, em 1936, demonstrou rigorosamente que o modelo bidimensional de fato apresenta transição de fase a temperatura finita (PEIERLS, 1936), contrariando a previsão de Ising e reacendendo o interesse da comunidade científica. O passo definitivo veio em 1944, quando Lars Onsager obteve a solução analítica exata do modelo de Ising 2D sem campo magnético externo (ONSAGER, 1944) — um feito considerado um dos mais impressionantes da Física Matemática do século XX. A solução de Onsager forneceu o valor exato da temperatura crítica e de expoentes críticos (como a divergência logarítmica do calor específico)¹, tornando o modelo um caso de teste padrão para teorias de transição de fase.

Nas décadas seguintes, o modelo de Ising transcendeu o magnetismo e tornou-se um laboratório conceitual para fenômenos universais de criticalidade, com aplicações que abran-

¹ O artigo original de 1944 estabelece a temperatura crítica exata e o comportamento singular do calor específico. A fórmula fechada para a magnetização espontânea (com expoente crítico $\beta = 1/8$, apresentada na Equação 4.2) foi anunciada por Onsager (1944) sem demonstração publicada, sendo a prova rigorosa atribuída a Yang (1952).

gem desde polímeros e ligas binárias até redes neurais e teoria de campos (BAXTER, 1982; YEOMANS, 1992).

1.2 Contextualização

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 σ 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 (ONSAGER, 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 (METROPOLIS et al., 1953);
- dinâmica de Glauber (GLAUBER, 1963);
- algoritmos de Wolff (WOLFF, 1989) e Swendsen-Wang (SWENDSEN; WANG, 1987).

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:

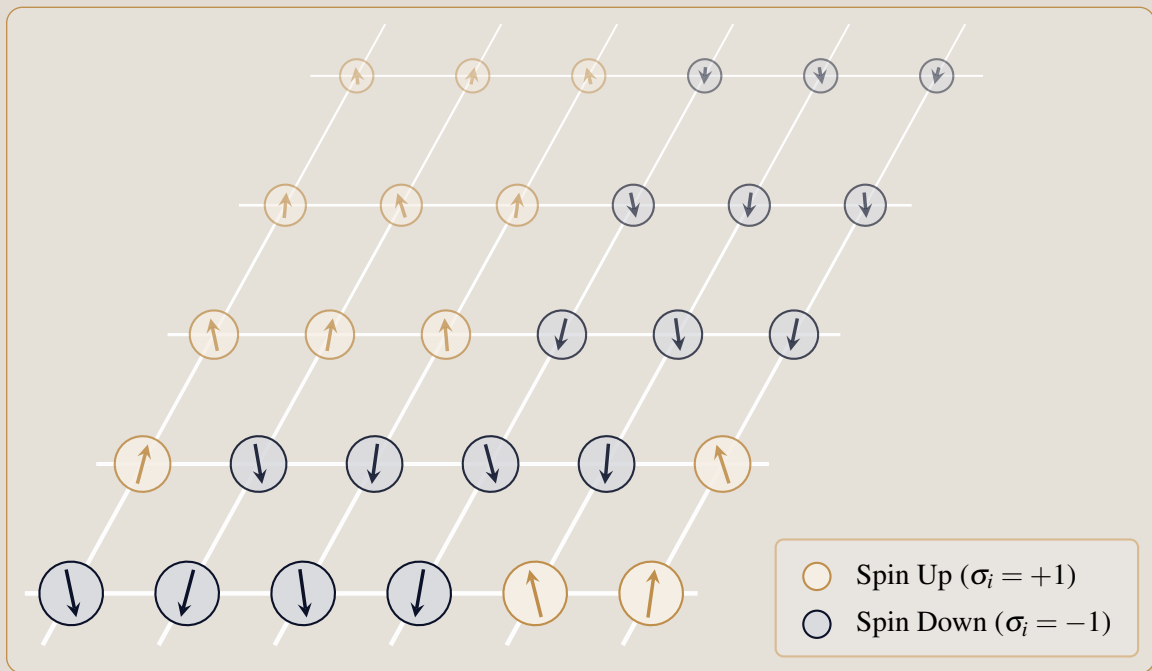
- redes neurais probabilísticas e aprendizado de máquina;
- otimização combinatória;
- biologia computacional;
- dinâmica social e formação de opinião;

- 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 1.1: Modelo de Ising bidimensional - 2D.

Representado como uma rede quadrada de spins interativos.



Fonte: Elaborado pelo Autor (2026).

1.3 Regra de Aceitação de Metropolis-Hastings

A amostragem estocástica da distribuição de Gibbs-Boltzmann é realizada por meio de uma cadeia de Markov cuja regra de transição obedece ao critério de Metropolis-Hastings. Dada uma configuração σ e uma configuração candidata σ' (obtida pela inversão de um único spin), a probabilidade de aceitação da transição é dada por

$$P_{\text{aceita}}(\sigma \rightarrow \sigma') = \min\left(1, e^{-\beta\Delta E}\right), \quad \Delta E = H(\sigma') - H(\sigma). \quad (1.1)$$

Para a inversão de um único spin no sítio i (isto é, $\sigma_i \rightarrow -\sigma_i$), com campo externo nulo ($h = 0$), a variação de energia associada possui a forma fechada

$$\Delta E_i = 2J \sigma_i \sum_{j \in \mathcal{N}(i)} \sigma_j, \quad (1.2)$$

em que $\mathcal{N}(i)$ denota o conjunto dos quatro vizinhos mais próximos do sítio i sob condições de contorno periódicas. Essa expressão fechada é a que se encontra implementada na função `metropolis_hastings_step` (variável de no código-fonte).

1.4 Grandezas Termodinâmicas Observáveis

Além da energia, duas grandezas macroscópicas são centrais para caracterizar a transição de fase do modelo. A magnetização por spin é definida como

$$m(\sigma) = \frac{1}{N} \sum_{i=1}^N \sigma_i, \quad M = \langle |m| \rangle, \quad (1.3)$$

atuando como parâmetro de ordem da transição ($M \rightarrow 0$ para $T > T_c$; $M > 0$ para $T < T_c$, no limite termodinâmico). A suscetibilidade magnética, obtida por relação de flutuação-dissipação, é dada por

$$\chi = \frac{\beta}{N} (\langle \mathcal{M}^2 \rangle - \langle \mathcal{M} \rangle^2), \quad \mathcal{M} = \sum_{i=1}^N \sigma_i, \quad (1.4)$$

e o calor específico por sítio, de forma análoga,

$$C = \frac{\beta^2}{N} (\langle H^2 \rangle - \langle H \rangle^2). \quad (1.5)$$

Ambas as grandezas exibem divergência (em sistemas de tamanho infinito) ou pico pronunciado (em sistemas finitos) na vizinhança de T_c , constituindo a assinatura quantitativa da transição de fase — em contraste com a observação puramente qualitativa e visual atualmente relatada. Para uma exposição unificada dessas relações de flutuação-dissipação, ver (KRAUTH, 2006).

1.5 Convergência da Cadeia de Markov: Balanço Detalhado e Ergodicidade

A convergência assintótica da cadeia de Markov gerada pelo algoritmo de Metropolis-Hastings para a distribuição estacionária de Gibbs-Boltzmann não é uma propriedade automática do procedimento, mas decorre de duas condições estruturais: o *balanço detalhado* (*detailed balance*) e a *ergodicidade* da cadeia.

Seja $T(\sigma \rightarrow \sigma')$ a probabilidade de transição entre duas configurações. Diz-se que a cadeia satisfaz o balanço detalhado em relação a uma distribuição π quando

$$\pi(\sigma) T(\sigma \rightarrow \sigma') = \pi(\sigma') T(\sigma' \rightarrow \sigma), \quad \forall \sigma, \sigma' \in \Sigma. \quad (1.6)$$

O balanço detalhado é condição suficiente para que π seja estacionária, pois sua soma sobre σ recupera a equação de balanço global $\sum_{\sigma} \pi(\sigma) T(\sigma \rightarrow \sigma') = \pi(\sigma')$. No algoritmo implementado, a transição decompõe-se em proposta e aceitação, $T(\sigma \rightarrow \sigma') = g(\sigma \rightarrow \sigma') A(\sigma \rightarrow \sigma')$, em que $g(\sigma \rightarrow \sigma') = 1/N$ é a probabilidade de proposta (seleção uniforme de sítio) e A é a probabilidade de aceitação de Metropolis. Como a proposta é simétrica ($g(\sigma \rightarrow \sigma') = g(\sigma' \rightarrow \sigma)$), já que inverter novamente o mesmo sítio retorna à configuração original com igual probabilidade), a razão de aceitações nos dois sentidos satisfaz exatamente $A(\sigma \rightarrow \sigma')/A(\sigma' \rightarrow \sigma) = \pi(\sigma')/\pi(\sigma) = e^{-\beta \Delta E}$, garantindo o balanço detalhado em relação a $\pi(\sigma) = e^{-\beta H(\sigma)}/Z$.

A ergodicidade exige irreducibilidade (toda configuração alcançável a partir de qualquer outra em número finito de passos) e aperiodicidade. A irreducibilidade decorre da própria estrutura de $\Sigma = \{-1, +1\}^N$: qualquer configuração é atingível por uma sequência finita de inversões individuais. A aperiodicidade decorre da probabilidade não nula de rejeição de propostas (quando $\Delta E > 0$), que introduz autoloops e elimina qualquer periodicidade estrutural. Satisfeitas essas condições, o Teorema Ergódico para cadeias de Markov garante que a distribuição empírica das amostras converge para π no limite de um número suficientemente grande de passos, independentemente da configuração inicial $\sigma^{(0)}$. Para uma exposição mais aprofundada do arcabouço teórico de balanço detalhado e ergodicidade em cadeias de Markov aplicadas à Mecânica Estatística, ver (NEWMAN; BARKEMA, 1999; LANDAU; BINDER, 2014).

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.

2.1 Gargalos em Linguagens Interpretadas

A amostragem de configurações via método de Metropolis-Hastings exige iterações na ordem de $\mathcal{O}(N \times S)$, 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 devido ao custo de interpretação e tipagem dinâmica a cada iteração.

Além disso, o cálculo da variação de energia (ΔE) 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 geral da execução.

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

Para viabilizar simulações de milhões de iterações por segundo, foi adotada uma arquitetura baseada na compilação JIT, utilizando a biblioteca Numba (LAM; PITROU; SEIBERT, 2015) no ecossistema Python. A principal função de atualização do estado da malha foi decorada com a diretiva `@njit`, o que permite:

1. **Eliminação do Overhead do Interpretador:** O código compilado via LLVM executa em nível de máquina, eliminando o custo de despacho de bytecode, a verificação dinâmica de tipos e o *boxing/unboxing* de objetos característicos do interpretador CPython (BEAZLEY, 2010). Observa-se que a compilação `@njit`, por padrão, não libera o *Global Interpreter Lock* (GIL); esse comportamento exigiria a declaração explícita do parâmetro

`nogil=True`, relevante apenas em cenários de paralelismo multi-thread — o que não se aplica à presente implementação, de natureza sequencial e single-threaded.

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 substituição elimina uma instrução de desvio condicional do corpo do laço, simplificando o fluxo de controle e favorecendo a geração de código vetorizável pelo compilador. Ressalta-se que a operação de módulo não é executada em um único ciclo de *clock*; em arquiteturas x86-64, divisão e módulo inteiros apresentam latências tipicamente entre 20 e 40 ciclos. O ganho de desempenho decorre, portanto, da eliminação da dependência de controle imposta pelo *branch*, e não do custo da operação aritmética. Quando n é potência de dois, o módulo pode ser substituído por uma operação *bitwise AND*:

$$(j - 1) \bmod n \equiv (j - 1) \& (n - 1), \quad \text{válido apenas se } n = 2^k. \quad (2.1)$$

Essa substituição não foi adotada, pois a implementação permite tamanhos de grade arbitrários.

2.3 Localidade de Cache e Estratégia de Amostragem

Um aspecto de desempenho adicional, não explorado na arquitetura descrita acima, diz respeito ao padrão de acesso à memória imposto pela estratégia de amostragem adotada. A implementação seleciona o sítio (i, j) a ser atualizado em cada iteração por meio de `np.random.randint(0, n)`, de modo que o acesso à matriz `grid` é essencialmente aleatório a cada passo.

Essa escolha tem duas implicações relevantes. Primeiro, para uma matriz armazenada em ordem *row-major* (padrão do Numpy), o acesso aos vizinhos verticais `grid[i_top, j]` e `grid[i_bottom, j]` exige saltos de memória de tamanho n elementos — para grades grandes (por exemplo, $n = 500$), esse salto frequentemente ultrapassa o tamanho de uma linha de cache (*cache line*, tipicamente 64 bytes) e pode exceder também a capacidade das caches L1/L2. Combinada à seleção aleatória do sítio a cada iteração, que impede qualquer aproveitamento de *prefetching* sequencial pelo hardware, a simulação torna-se, na prática, limitada pela latência de memória (*memory-latency-bound*) para grades de maior dimensão, mesmo após a eliminação do overhead do interpretador pela compilação JIT.

Essa característica contrasta com a abordagem clássica de varredura sequencial (*Monte Carlo sweep*), na qual cada sítio da rede é visitado uma vez por ciclo completo, tipicamente em ordem *checkerboard* (ou *red-black*): os sítios são particionados em duas sub-redes, de modo que todos os vizinhos de um sítio de uma sub-rede pertencem exclusivamente à outra. Além de

preservar a localidade sequencial de acesso à memória, essa decomposição permite a atualização **paralela** de todos os sítios de uma mesma cor em uma única passada, já que não compartilham vizinhos entre si.

A escolha pela amostragem aleatória de sítio, adotada no presente trabalho, alinha-se à formulação teórica clássica do algoritmo de Metropolis por sua simplicidade estatística. A varredura *checkerboard* constitui, portanto, uma direção natural para trabalhos futuros voltados à otimização de performance em grades de grande escala.

2.4 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` em Python ou `ggplot2` em R) constroem objetos gráficos complexos a cada quadro, o que se torna o gargalo limitante da simulação iterativa.

A solução implementada utiliza **Manipulação Matricial Direta** via Numpy (HARRIS et al., 2020). 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*).

>_ Console // 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(N), sem overhead de laço
    # interpretado
    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 desaceleração crítica (*Critical Slowing Down*)¹ sem comprometer a latência da interface.

¹ O termo refere-se ao aumento do tempo de autocorrelação da cadeia de Markov nas vizinhanças da temperatura crítica T_c , fenômeno intrínseco a algoritmos de atualização local como o Metropolis-Hastings (SOKAL, 1997; BINDER; HEERMANN, 2010).

CAPÍTULO 3

APLICAÇÃO INTERATIVA E PORTFÓLIO

A etapa final deste projeto consiste na apresentação pública dos resultados por meio de uma aplicação interativa acessível a qualquer pessoa com conexão à internet, sem necessidade de instalação de software. Este capítulo descreve a plataforma escolhida, os controles disponíveis na interface e a integração com o portfólio pessoal do autor.

3.1 Visão Geral da Plataforma

O *Streamlit* é um *framework* de código aberto para a construção de aplicações de dados em Python (Streamlit Inc., 2019). Sua proposta central é permitir que cientistas e engenheiros transformem scripts analíticos em interfaces web interativas com mínimo esforço de engenharia de *front-end*: widgets de controle (controles deslizantes, botões, seletores) são definidos no próprio script Python e o *framework* gerencia automaticamente o ciclo de atualização da interface em resposta a eventos do usuário.

O *Streamlit Community Cloud* é a plataforma de implantação (*deploy*) gratuita associada, que permite publicar aplicações Streamlit diretamente a partir de um repositório GitHub público. A cada atualização do repositório, a plataforma recompila e republica a aplicação automaticamente — um fluxo de integração contínua adequado ao ciclo de desenvolvimento científico incremental.

A escolha do Streamlit para este projeto atende a três critérios complementares:

1. **Compatibilidade nativa:** o motor computacional da simulação (Numpy, Numba) roda integralmente em Python, sem necessidade de serialização ou tradução para outra linguagem;
2. **Renderização em tempo real:** o mecanismo de atualização de estado (`st.empty().image()`) permite substituir a imagem da malha de spins a cada *frame* do laço de animação, viabilizando a visualização em tempo real;
3. **Acessibilidade:** qualquer leitor deste relatório pode acessar a simulação ao vivo diretamente pelo navegador, sem instalar Python, Numba ou qualquer dependência.

3.2 Interface e Controles da Aplicação

A interface é organizada em dois painéis: uma barra lateral (*sidebar*) com os controles do sistema e uma área principal com a visualização. O painel de visualização exibe dois estados lado a lado — a configuração inicial aleatória e a evolução em tempo real — permitindo comparação direta entre o estado desordenado e o estado emergente após a termalização.

A Tabela 3.1 descreve os cinco parâmetros de controle expostos ao usuário, seus intervalos de variação e o efeito físico associado.

Tabela 3.1: Parâmetros de controle da simulação interativa.

Parâmetro	Intervalo	Padrão	Efeito Físico
$\beta = 1/k_B T$ (inverso da temperatura)	[0; 3,0]	0,44	Controla a temperatura. $\beta < \beta_c \approx 0,441$ induz desordem (paramagnetismo); $\beta > \beta_c$ induz ordem (ferromagnetismo).
Tamanho da grade $N \times N$	[50; 500]	150	Define a resolução espacial. Grades maiores revelam domínios detalhados, mas elevam o custo computacional.
Total de iterações	[100.000; 5.000.000]	1.000.000	Total de tentativas de inversão de spin. Valores maiores garantem melhor equilíbrio térmico (termalização).
Quadros de atualização (<i>frames</i>)	[10; 200]	50	Frequência de atualização da malha na tela. Controla a fluidez da animação visual.
Atraso por quadro (s)	[0; 1,0]	0,1	Pausa entre atualizações de quadros, permitindo observar a evolução dos domínios em câmera lenta.

Fonte: Elaborado pelo Autor (2026).

A Figura 3.1 apresenta uma captura de tela da aplicação em funcionamento, obtida com os parâmetros padrão ($\beta = 0,44$, grade 150×150 , 10^6 iterações).

Figura 3.1: Interface da aplicação interativa do Modelo de Ising 2D.

Painel esquerdo: estado inicial aleatório. Painel direito: estado evoluído após 10^6 iterações com $\beta \approx \beta_c$.



Fonte: Elaborado pelo Autor (2026).

3.3 Acesso ao Aplicativo

A simulação está disponível publicamente no endereço:

 <https://douglasmoura-ising-2d.streamlit.app>

O código-fonte completo, incluindo o script da aplicação (`app_ising.py`), o arquivo de dependências (`requirements.txt`) e o script R original da atividade acadêmica (`modelo_ising.R`), está disponível no meu repositório público no GitHub:

 github.com/douglasmoura/modelo_ising_2d_metropolis

Figura 3.2: QR Code de acesso ao app - Streamlit Community Cloud.

Aponte a câmera do celular para acessar a simulação.



Fonte: Elaborado pelo Autor (2026).

3.4 Integração com o Portfólio

O aplicativo interativo e o presente relatório estão integrados ao meu portfólio pessoal, disponível em:



`douglas-moura-portfolio.pages.dev`

Na página do *case* dedicado a este projeto (`/case_ising_2d`), o visitante encontra dois botões de acesso:

- **Relatório Técnico** — redireciona para o presente documento em formato PDF, permitindo a leitura completa da fundamentação teórica, das decisões de implementação e dos resultados;
- **Live App** — redireciona diretamente para `douglasmoura-ising-2d.streamlit.app`, onde é possível interagir com a simulação em tempo real sem qualquer instalação.

Essa arquitetura de apresentação — relatório técnico complementar ao aplicativo interativo — reflete uma abordagem de comunicação científica em dois níveis: o rigor formal para o leitor especializado e a acessibilidade imediata para o público geral.

RESULTADOS E VALIDAÇÃO EMPÍRICA

A observação dos fenômenos críticos do modelo de Ising na aplicação interativa permite uma validação qualitativa do comportamento teórico previsto. Este capítulo descreve os três regimes termodinâmicos identificados, a correspondência com a temperatura crítica de Onsager, o fenômeno de desaceleração crítica (*Critical Slowing Down*) e o surgimento de domínios magnéticos em torno de T_c .

4.1 Regimes de Temperatura e Comportamento Qualitativo

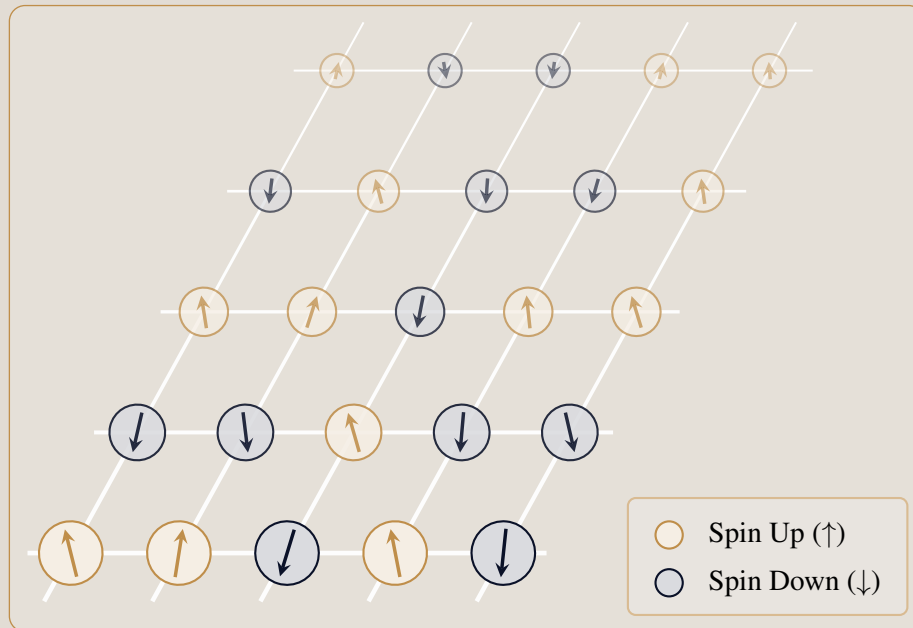
O comportamento macroscópico do modelo depende fundamentalmente da relação entre o parâmetro $\beta = 1/k_B T$ e o valor crítico $\beta_c \approx 0,4407$. Três regimes distintos são observados:

- **Regime de alta temperatura** ($\beta \ll \beta_c$, ex.: $\beta = 0,1$): a agitação térmica domina as interações ferromagnéticas. Os spins se invertem quasi-aleatoriamente, sem formação de domínios coerentes. A magnetização média $\langle |m| \rangle$ converge para zero.
- **Regime crítico** ($\beta \approx \beta_c \approx 0,44$): o sistema se encontra exatamente na fronteira entre ordem e desordem. Surgem domínios magnéticos de todas as escalas de tamanho, com estrutura fractal característica. É nesse regime que as flutuações são máximas e a suscetibilidade χ e o calor específico C atingem seus valores de pico.
- **Regime de baixa temperatura** ($\beta \gg \beta_c$, ex.: $\beta = 1,5$): as interações ferromagnéticas dominam. O sistema evolui rapidamente para um estado de alta ordem, com quase todos os spins alinhados em uma única direção. A magnetização $\langle |m| \rangle \rightarrow 1$.

As Figuras 4.1, 4.2 e 4.3 ilustram esquematicamente os três regimes por meio de redes 5×5 representativas em perspectiva 3D.

Figura 4.1: Regime de alta temperatura (desordem) do Modelo de Ising 2D ($\beta = 0,10$).

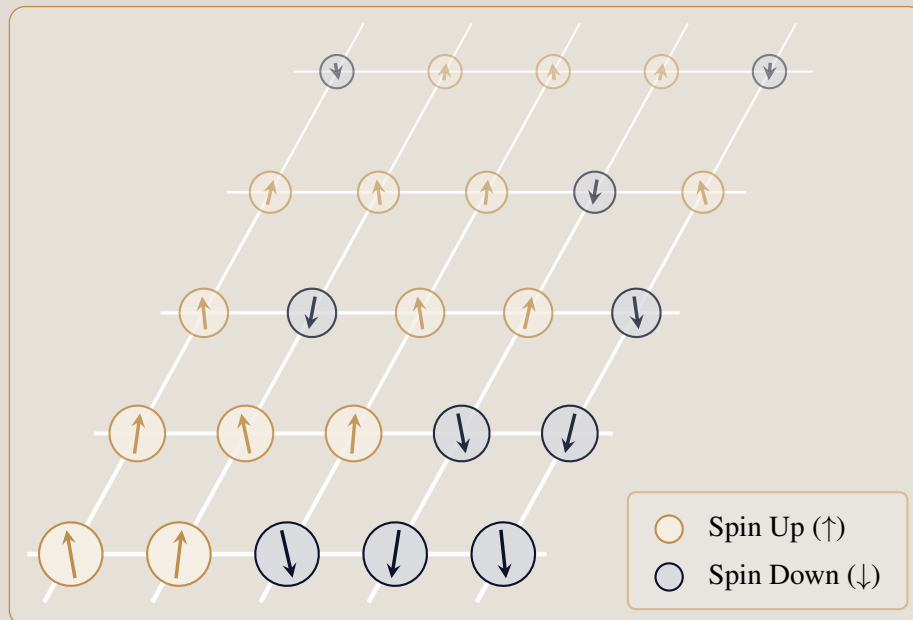
Agitação térmica elevada superando o acoplamento ferromagnético; spins altamente desordenados.



Fonte: Elaborado pelo Autor (2026).

Figura 4.2: Regime crítico (transição de fase) do Modelo de Ising 2D ($\beta = 0,44$).

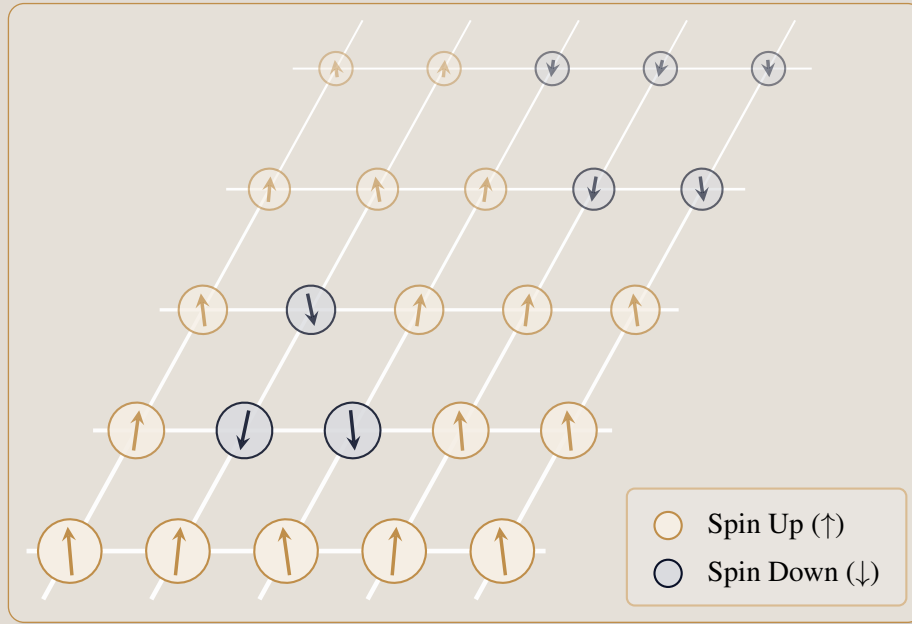
Surgimento de domínios fractais de todas as escalas com flutuações magnéticas máximas.



Fonte: Elaborado pelo Autor (2026).

Figura 4.3: Regime de baixa temperatura (ordem) do Modelo de Ising 2D ($\beta = 1,50$).

Domínio ferromagnético ordenado e unificado, com defeitos térmicos isolados.



Fonte: Elaborado pelo Autor (2026).

4.2 Temperatura Crítica e Parâmetro de Ordem

A temperatura crítica do modelo de Ising 2D sem campo externo foi determinada analiticamente por Onsager (ONSAGER, 1944):

$$T_c = \frac{2J}{k_B \ln(1 + \sqrt{2})} \approx 2,269 \frac{J}{k_B}, \quad \Rightarrow \quad \beta_c = \frac{\ln(1 + \sqrt{2})}{2J} \approx 0,4407. \quad (4.1)$$

A aplicação interativa utiliza $\beta_c \approx 0,44$ como valor padrão do controle deslizante, alinhado ao valor analítico. A Tabela 4.1 sintetiza a correspondência entre o comportamento observado na simulação e as previsões teóricas para cada regime.

Tabela 4.1: Correspondência entre comportamento observado e previsão teórica.

Regime	β	Observação na Simulação	Previsão Teórica
Alta Temperatura	0,1	Distribuição aleatória de spins; ausência de domínios; magnetização nula	$\langle m \rangle \rightarrow 0$; fase paramagnética
Crítico	0,44	Domínios de todas as escalas; estrutura fractal; flutuações máximas	$\langle m \rangle \rightarrow 0^+$; pico em χ e C
Baixa Temperatura	1,5	Quasi-totalidade dos spins alinhados; um único domínio dominante	$\langle m \rangle \rightarrow 1$; fase ferromagnética

Fonte: Elaborado pelo Autor (2026).

A magnetização espontânea abaixo de T_c é dada pela célebre expressão analítica anunciada por Onsager em 1949 e demonstrada rigorosamente por Yang (YANG, 1952) como

$$M(T) = \left[1 - \sinh^{-4} \left(\frac{2J}{k_B T} \right) \right]^{1/8}, \quad T < T_c, \quad (4.2)$$

exibindo o expoente crítico $\beta_{\text{exp}} = 1/8$, característico da classe de universalidade do modelo de Ising 2D (YANG, 1952; BAXTER, 1982; YEOMANS, 1992).

4.3 Desaceleramento Crítico (*Critical Slowing Down*)

Um fenômeno notável observável na simulação próximo a β_c é o desaceleramento crítico (*Critical Slowing Down*): o número de iterações necessário para a cadeia de Markov convergir para o equilíbrio — medido pelo tempo de autocorrelação integrado τ_{int} (SOKAL, 1997) — diverge ao se aproximar de T_c segundo a lei de potência

$$\tau_{\text{int}} \sim \xi^z \sim |T - T_c|^{-z\nu}, \quad (4.3)$$

em que ξ é o comprimento de correlação, ν é o expoente crítico do comprimento de correlação ($\nu = 1$ para o modelo de Ising 2D) e $z \approx 2$ é o expoente dinâmico do algoritmo de Metropolis-Hastings com atualização local (BINDER; HEERMANN, 2010).

Na prática, isso significa que, para $\beta \approx \beta_c$, a simulação demora significativamente mais iterações para alcançar o equilíbrio do que em regimes de temperatura distante de T_c . Esse efeito é observável diretamente na interface: para $\beta = 0,44$, a estrutura de domínios continua evoluindo visivelmente mesmo após 10^6 iterações — ao passo que para $\beta = 1,5$ a convergência para o estado ordenado ocorre em poucas centenas de iterações por sítio.

Algoritmos de agrupamento (*cluster algorithms*), como os de Wolff (WOLFF, 1989) e Swendsen-Wang (SWENDSEN; WANG, 1987), foram desenvolvidos especificamente para mitigar o desaceleramento crítico, pois operam sobre *clusters* de spins em vez de sítios individuais, reduzindo drasticamente o expoente dinâmico para valores próximos de zero ($z \approx 0,25$ para o algoritmo de Wolff no modelo de Ising 2D). A adoção desses algoritmos representa uma extensão natural do presente trabalho.

4.4 Domínios Fractais e Criticalidade

Em torno de T_c , o sistema exhibe domínios magnéticos — regiões contíguas de spins alinhados — com uma distribuição de tamanhos que segue uma lei de potência, característica de sistemas invariantes de escala. Essa ausência de escala preferencial manifesta-se visualmente como uma estrutura fractal: independentemente do nível de zoom aplicado à malha, a textura dos domínios se repete com padrões estatisticamente similares.

Essa invariância de escala é a assinatura visual da criticalidade: no ponto crítico, o comprimento de correlação $\xi \rightarrow \infty$ (no limite termodinâmico), e domínios de qualquer tamanho coexistem com probabilidade não negligenciável. A dimensão fractal dos contornos de domínio no ponto crítico do modelo de Ising 2D é conhecida analiticamente como $D_f = 11/8 = 1,375$, sendo originalmente prevista via invariância conforme e gás de Coulomb por Duplantier (DUPLANTIER, 1989) e demonstrada de maneira rigorosa no âmbito da Evolução de Schramm-Loewner (SLE) por Smirnov (SMIRNOV, 2010).

Nota sobre Versões Futuras

Os resultados apresentados neste capítulo são de natureza qualitativa e baseados na observação visual da simulação. Uma versão futura deste relatório incorporará resultados quantitativos: séries temporais de $\langle |m| \rangle$ e $\langle H \rangle$ em função de β , estimativas de χ e C via relações de flutuação-dissipação (Seção 1.4), e ajuste da curva $M(T)$ à solução exata de Onsager. Esses experimentos computacionais requererão a implementação de um modo de *batch* (sem interface gráfica) para coleção sistemática de médias de equilíbrio.

REFERÊNCIAS BIBLIOGRÁFICAS

- BAXTER, R. J. *Exactly Solved Models in Statistical Mechanics*. London: Academic Press, 1982.
- BINDER, K.; HEERMANN, D. W. *Monte Carlo Simulation in Statistical Physics: An Introduction*. 5th. ed. Berlin: Springer, 2010.
- DUPLANTIER, B. Fractal dimension of critical loops in two dimensions. *Physical Review Letters*, APS, v. 62, n. 9, p. 1041–1044, 1989.
- GLAUBER, R. J. Time-dependent statistics of the ising model. *Journal of Mathematical Physics*, AIP Publishing, v. 4, n. 2, p. 294–307, 1963.
- HARRIS, C. R. et al. Array programming with NumPy. *Nature*, Nature Publishing Group, v. 585, n. 7825, p. 357–362, 2020.
- ISING, E. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik*, Springer, v. 31, n. 1, p. 253–258, 1925.
- KRAUTH, W. *Statistical Mechanics: Algorithms and Computations*. Oxford: Oxford University Press, 2006.
- LAM, S. K.; PITROU, A.; SEIBERT, S. Numba: A llvm-based python jit compiler. In: *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. [S.l.: s.n.], 2015. p. 1–7.
- LANDAU, D. P.; BINDER, K. *A Guide to Monte Carlo Simulations in Statistical Physics*. 4th. ed. Cambridge: Cambridge University Press, 2014.
- LENZ, W. Beitrag zum verständnis der magnetischen erscheinungen in festen körpern. *Physikalische Zeitschrift*, v. 21, p. 613–615, 1920.
- METROPOLIS, N. et al. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, AIP Publishing, v. 21, n. 6, p. 1087–1092, 1953.
- NEWMAN, M. E. J.; BARKEMA, G. T. *Monte Carlo Methods in Statistical Physics*. Oxford: Clarendon Press, 1999.
- ONSAGER, L. Crystal statistics. i. a two-dimensional model with an order-disorder transition. *Physical Review*, APS, v. 65, n. 3-4, p. 117–149, 1944.
- PEIERLS, R. On Ising’s model of ferromagnetism. *Mathematical Proceedings of the Cambridge Philosophical Society*, Cambridge University Press, v. 32, n. 3, p. 477–481, 1936.
- SMIRNOV, S. Conformal invariance in random cluster models. i. holomorphic fermions in the ising model. *Annals of Mathematics*, JSTOR, v. 172, n. 2, p. 1435–1467, 2010.
- SOKAL, A. D. Monte carlo methods in statistical mechanics: Foundations and new algorithms. In: *Functional Integration: Basics and Applications*. Boston, MA: Springer, 1997. p. 131–192.

Streamlit Inc. *Streamlit — The fastest way to build and share data apps*. 2019. <<https://streamlit.io>>. Acessado em: julho de 2026.

SWENDSEN, R. H.; WANG, J.-S. Nonuniversal critical dynamics in monte carlo simulations. *Physical Review Letters*, APS, v. 58, n. 2, p. 86–88, 1987.

WOLFF, U. Collective monte carlo updating for spin systems. *Physical Review Letters*, APS, v. 62, n. 4, p. 361–364, 1989.

YANG, C.-N. The spontaneous magnetization of a two-dimensional ising model. *Physical Review*, APS, v. 85, n. 5, p. 808–816, 1952.

YEOMANS, J. M. *Statistical Mechanics of Phase Transitions*. Oxford: Oxford University Press, 1992.

APÊNDICE A

ACESSO AO APLICATIVO INTERATIVO

Este apêndice reúne as informações necessárias para que o leitor acesse e explore a simulação interativa do Modelo de Ising 2D, disponível publicamente na plataforma Streamlit Community Cloud. Nenhuma instalação de software é necessária: basta um navegador web moderno.

A.1 Endereço do Aplicativo

A simulação ao vivo está disponível no seguinte endereço:

 <https://douglasmoura-ising-2d.streamlit.app>

Alternativamente, o leitor pode acessar o aplicativo diretamente pela página do *case* no portfólio do autor:

 douglas-moura-portfolio.pages.dev/case_ising_2d

A.2 Como Interagir com a Simulação

Ao acessar a aplicação, o leitor encontrará a seguinte sequência de passos:

1. **Ajustar os parâmetros** na barra lateral esquerda. Os controles disponíveis são:
 - **Beta (β)**: controla o regime térmico. Valores próximos de 0,44 correspondem ao ponto crítico. Reduza β para observar desordem; aumente para observar ordem ferromagnética.
 - **Tamanho da Grade ($N \times N$)**: define a resolução espacial. Grades menores (ex.: 50×50) rodam mais rápido; grades maiores (ex.: 300×300) revelam mais detalhes estruturais.
 - **Total de Iterações**: número de tentativas de inversão de spin. Recomenda-se ao menos 10^6 para observar a termalização.
 - **Quadros de Atualização**: número de vezes que a imagem é redesenhada durante a simulação.

- **Atraso por Quadro:** insira um atraso (em segundos) para desacelerar a animação e observar a evolução com mais calma.
2. **Clicar em “▶ Iniciar Simulação”** para iniciar o processo.
 3. **Observar** o painel esquerdo (estado inicial aleatório) e o painel direito (estado evoluído em tempo real).
 4. Após a conclusão, o aplicativo exibe o tempo de cálculo puro (excluídas as pausas visuais).

A.3 Sugestões de Experimentos

Para explorar os três regimes descritos no Capítulo 4, recomenda-se:

Tabela A.1: Configurações sugeridas para exploração dos regimes térmicos.

Regime	β	Grade	O que observar
Desordem	0,10	150×150	Distribuição aleatória sem formação de domínios
Crítico	0,44	200×200	Domínios fractais de múltiplas escalas; evolução lenta
Ordem	1,50	150×150	Convergência rápida para um único domínio dominante

Fonte: Elaborado pelo Autor (2026).

A.4 Código-Fonte e Repositório

O código-fonte completo da aplicação está disponível publicamente no GitHub:

 github.com/douglascmoura/modelo_ising_2d_metropolis

O repositório contém:

- `app_ising.py` — script principal da aplicação Streamlit;
- `modelo_ising.R` — implementação original em R da atividade acadêmica;
- `requirements.txt` — lista de dependências Python para execução local ou deploy.

**EMAIL**

douglasmoura21@gmail.com

**LINKEDIN**

linkedin.com/in/douglas-chaves-moura

**GITHUB**

github.com/douglasmoura

**WEBSITE**

douglas-moura-portfolio.pages.dev

