

Método de Monte Carlo para o Modelo do Ferromagneto de Ising Computacional para Rede Bidimensional Quadrada

Benjamin Giriboni Monteiro
Daniel Carlos Souza Santos
Diego Rodrigues Cavalcante
Léo Cymrot Cymbalista

Universidade de São Paulo

Resumo

O método de Monte Carlo se baseia em cálculo de medias complicadas a partir de uma média muito mais simples. Calculando médias a partir de um número de configurações específicas do sistema em questão. O presente documento se baseia na ideia de Monte Carlo para uma simulação computacional do modelo de Ising de um ferromagneto simples com constituintes com spins.

1 Introdução

1.1 O Modelo de Ising

O problema consiste em analisar um modelo de constituintes (partículas) portadores de spin para um sistema de rede quadrada, onde cada sítio da rede (cada partícula do sistema) interage com cada vizinho, isso devido a interação de origem magnética gerada por seus spins $\sigma_i = \pm 1$.

O modelo de Ising se propõe a resolver essa questão introduzindo o seguinte hamiltoniano do sistema dado por:

$$\mathcal{H} = -J \sum_{i,j} \sigma_i \sigma_j, \quad (1)$$

onde a soma é indicada a ser realizada para os vizinhos mais próximo do sítio i , e J uma constante positiva.

Notemos que esse é o caso mais simples possível, onde o sistema em si não está em contato com um campo magnético H externo ao sistema. Ou seja, as interações no sistema são devidas apenas as configurações dos spins de cada sítio.

Apesar da simplicidade do sistema, essa soma sobre todos os sítios continua sendo uma tarefa complicada. Entretanto, podemos utilizar um método muito mais simples para realizar as medições e previsões termodinâmicas desse sistema, utilizando o Método de Monte Carlo nesse sistema.

1.2 O método de Monte Carlo e o Algoritmo de Metropolis

A ideia do Método de Monte Carlo consiste em obter uma estimativa de uma variável $\langle A \rangle$ por uma soma muito mais simples, uma média aritmética sobre um número relativamente pequeno, mas devidamente selecionado, de “configurações microscópicas representativas do equilíbrio”.

$$\langle A \rangle = \frac{1}{M} \sum_{i=1}^M A_i \quad (2)$$

A ideia desse método seria calcularmos o valor esperado da magnetização desse sólido através desse método, onde escolhemos um numero de preferência de passos de Monte Carlo M .

Juntamente com essa ideia, utilizaremos a ajuda de um outro algoritmo chamado de “Algoritmo de Metropolis”. A ideia é gerar uma cadeia de configurações que num numero grande de loops convergirá para as configurações de equilíbrio gibbsiano.

O método funciona da seguinte maneira: Escolhe-se uma configuração específica do nosso sistema (por exemplo todos os spins positivos $\sigma_i = 1, \forall i \in \mathbb{N}, 1 \leq i \leq N$) em seguida, selecionamos um sitio da rede e mudamos o spin desse sitio. Comparamos a energia da configuração anterior e da nova configuração com a mudança de spin dada pela equação 3:

$$\Delta E = -2\mathcal{H}_0 \quad (3)$$

Onde \mathcal{H}_0 é a energia do sitio antes da mudança no spin devido a interação com as 4 partículas vizinhas. E com esse valor de variação de energia, calculamos a seguinte relação:

$$r = e^{-\frac{\Delta E}{T}} \quad (4)$$

Sendo T a temperatura que a rede esta submetida. Com esse valor calculado, o algoritmo realiza os seguintes passos:

1. Se $\Delta E < 0$ aceite a nova configuração microscópica e continue o processo, selecionando um outro sitio sequencialmente da rede;
2. Se $\Delta E > 0$ compare o valor obtido de r com um número aleatório z entre 0 e 1. Aceite a mudança de sinal do spin escolhido se $r > z$; mantenha o mesmo sinal de antes se $r < z$.
3. Utilize a configuração ao final dessa sequência e siga para outro sitio para repetir todo o processo.

A ideia é repetir todo esse processo em todas os sítios da rede muitas vezes seguidas. Nesse caso, o numero de vezes que esse algoritmo percorre toda a rede é identificado como um “passo de Monte Carlo”.

É esperado que o sistema chegue no equilíbrio, quando os constituintes do sistema não apresentam mais uma variação abrupta nos valores de spin.

Sabemos que nos limites termodinâmicos, os desvios estatísticos são muito desprezíveis se o numero N de constituintes da rede for suficiente grande. Portanto, os desvios em relação a média dos valores que obtivermos com as considerações estatísticas se aproximarão de forma mais satisfatória do valor “absoluto” dos resultados terão desvios muito pequenos.

1.3 Modelo comparativo

A fins de comparação com os resultados que obtivermos, podemos utilizar uma equação já estabelecida para a magnetização desse modelo, proposta por Onsager [L. Onsager, Nuovo Cimento, Suppl., 6, 621, 1949], e posteriormente confirmado por Yang [C. N. Yang, Phys. Rev.

85, 808, 1952], como a magnetização espontânea e adimensional m_0 que é dada por uma equação extremamente trivial e simples:

$$m_0 = \{1 - [\sinh(2\beta J)]^{-4}\}^{\frac{1}{8}} = \left\{1 - \left[\sinh\left(\frac{2}{T}\right)\right]^{-4}\right\}^{\frac{1}{8}} \quad (5)$$

Sendo que a temperatura crítica nesse modelo é dado quando $\sinh(2\beta_c J) = 1$, de forma a obter:

$$T_c = \frac{2}{\ln(1 + \sqrt{2})} = 2,269185 \dots \quad (6)$$

Cujo gráfico esquematizado da magnetização contra a temperatura, a partir desse modelo, é estabelecido na figura 1 abaixo:

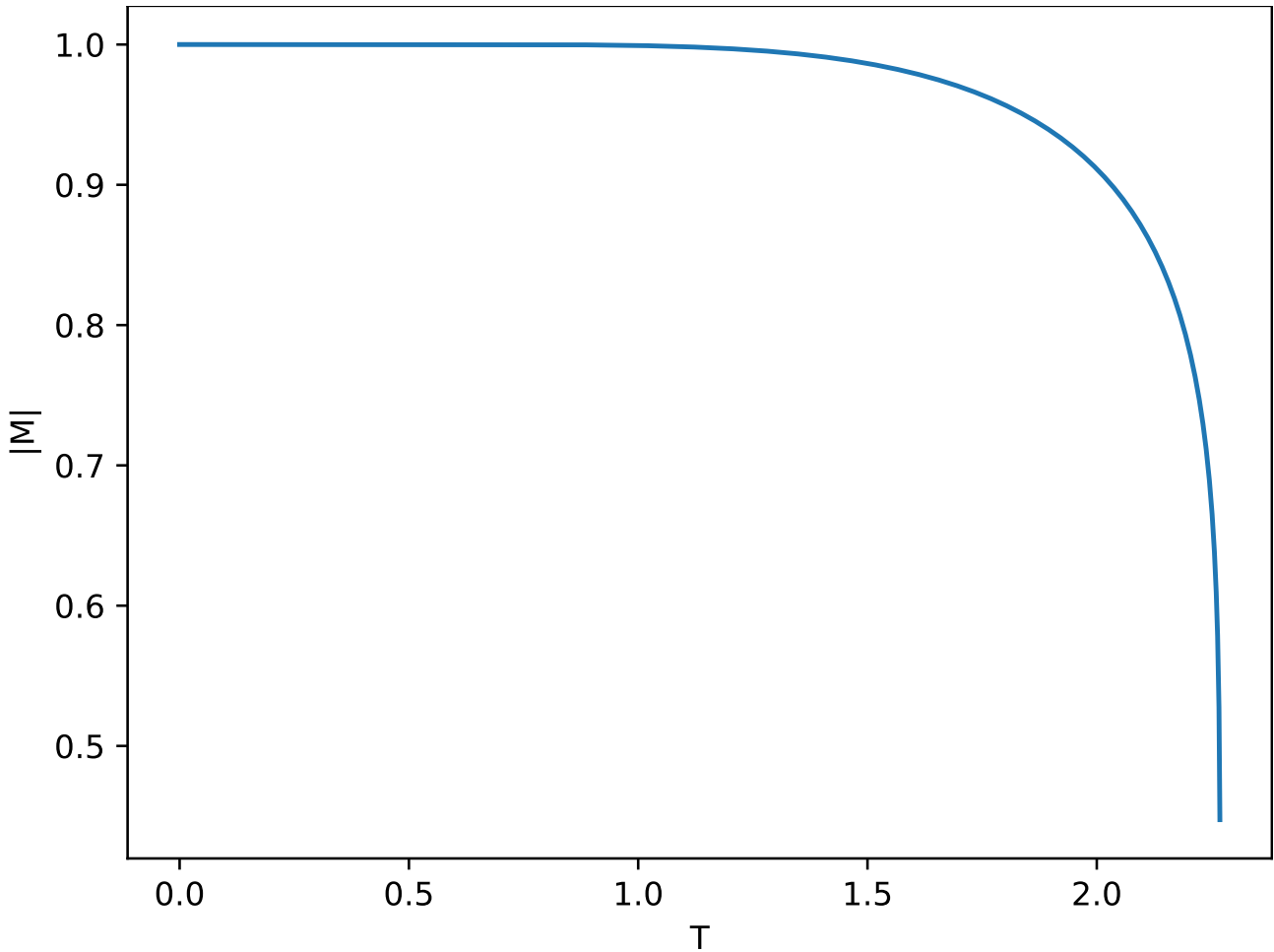


Figura 1: Gráfico da função teórica para a magnetização em função da temperatura, conforme a equação (5).

2 Implementação do Algoritmo em um Código Python

Com todas as considerações já feitas sobre o algoritmo e o método utilizado, escrevemos um programa para executar essas simulação do modelo de Ising:

```

1 kb = 1
2 e = 2.7182818284590452

```

```

3 import random as rnd
4 import copy
5 import matplotlib.pyplot as fig
6
7 def R(Range): #Número aleatório entre -Range e Range
8     return(2*Range*rnd.random() - Range)
9
10 def S():
11     if R(1) > 0:
12         return(1)
13     else:
14         return(-1)
15
16 def f(x):
17     return((x-1.32)**3 + 2.3)
18
19 graf, axs = fig.subplots(2)
20 N = int(input('Tamanho da rede: '))
21 P = 1000 #P é o numero de passos da simulação
22 fig.figure(figsize=(8, 14))
23 tam = 2
24 NTs = 40
25 TemperaturaGrafico = 0
26
27 L = []
28 M = []
29 for i1 in range(N):
30     L = []
31     for i2 in range(N):
32         L = L + [1]
33     M = M + [L]
34     M2 = copy.deepcopy(M)
35
36 for i6 in range(1, NTs + 1):
37     T = f(2.5*i6/NTs)
38     print(i6)
39     M = M2
40     MagTotal = 0
41     for i3 in range(P):
42         V = 0
43         for i4 in range(N):
44             for i5 in range(N):
45                 DeltaE = 2*kb*(M[i4][i5]*M[(i4+1)%N][i5] + M[i4][i5]*M[i4][(i5+1)%N] + M[i4][i5]*M[(i4-1)%N][i5] + M[i4][i5]*M[i4][(i5-1)%N])
46                 r = e**(-DeltaE/T)
47                 if DeltaE < 0:
48                     M[i4][i5] = -M[i4][i5]
49                 elif r > R(0.5)+0.5:
50                     M[i4][i5] = -M[i4][i5]
51                 else:
52                     M[i4][i5] = M[i4][i5]
53                 V = V + M[i4][i5]
54         if i3 > 499:
55             MagTotal = MagTotal + abs(V)
56         if i6 == NTs//3:
57             TemperaturaGrafico = T
58             x2 = i3
59             y2 = V
60             axs[0].scatter(x2, y2, s=tam, c="mediumblue")
61             axs[0].set(xlabel='Passos do Monte Carlo',ylabel='M')
62
63 MagMedia = MagTotal/(P-500)

```

```

62
63
64     x1 = T
65     y1 = MagMedia/N**2
66
67     axs[1].scatter(x1, y1, s=tam, c="mediumblue")
68     axs[1].set(xlabel='T',ylabel='|M|')
69
70 print(TemperaturaGrafico)
71 fig.show()

```

3 Análises dos Resultados

Com o código implementado com o algoritmo de metropolis, podemos obter as imagens da simulação:

Podemos perceber que, com base nos gráficos obtidos pela simulação, o gráfico da magnetização espontânea do ferromagneto pela temperatura segue de forma extremamente coerente com o modelo de Onsager visitado na Introdução. Portanto, o algoritmo de Metropolis combinado com o Método de Monte Carlo para o calculo de média, conseguiu convergir para o modelo termodinâmico do equilíbrio gibbsiano.

Contudo, com quantos mais passos de Monte Carlo estabelecidos, melhor a simulação representa o modelo da rede quadrada de Ising no limite termodinâmico.

Para uma rede quadrada 40×40 , com 1000 passos de Monte Carlo, obtivemos o seguinte resultado:

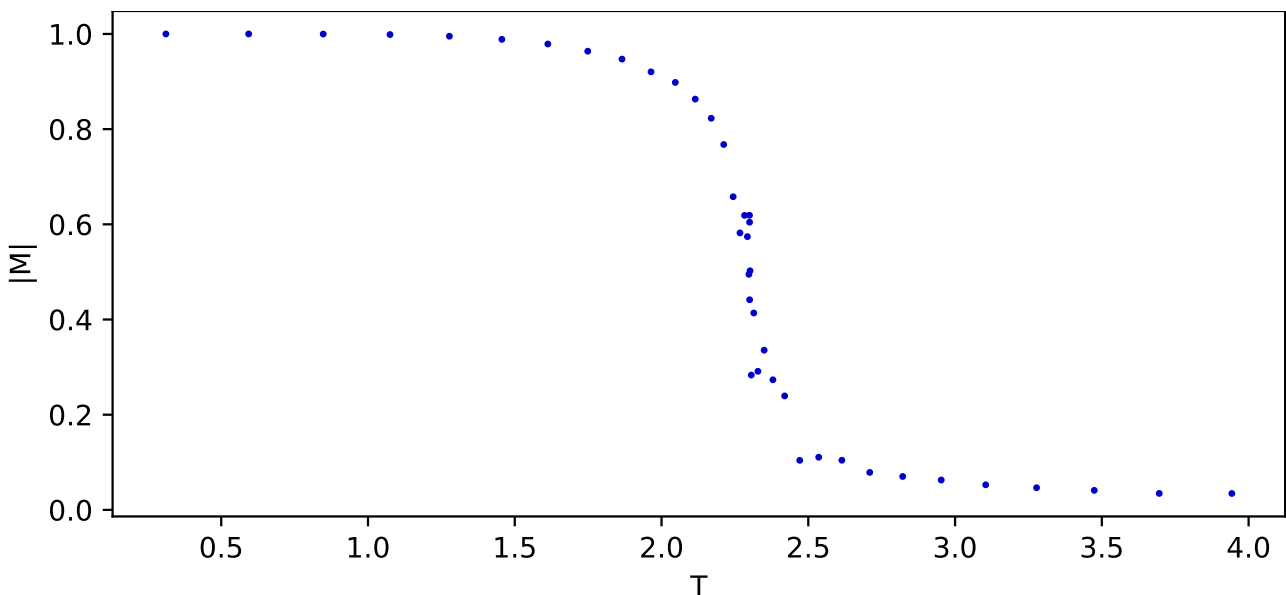


Figura 2: Gráfico da Magnetização em função da Temperatura para a rede de tamanho 40×40 .

Esse método poderoso pode ser explicado de uma maneira extremamente cuidadosa, fazendo uma análise estocástica do sistema. No presente artigo, não queremos nos aprofundar na ideia, mas podemos perceber que as previsões teóricas do método estocástico explicam bem os resultados obtidos.

Podemos perceber que para valores de temperatura abaixo da temperatura crítica, vemos que a magnetização converge bem e rapidamente ao valor esperado, conforme a figura 3.

Em contrapartida, próximo a temperatura critica, os valores oscilam de forma abrupta, resultando na necessidade de um grande numero de passos de Monte Carlo para a convergência

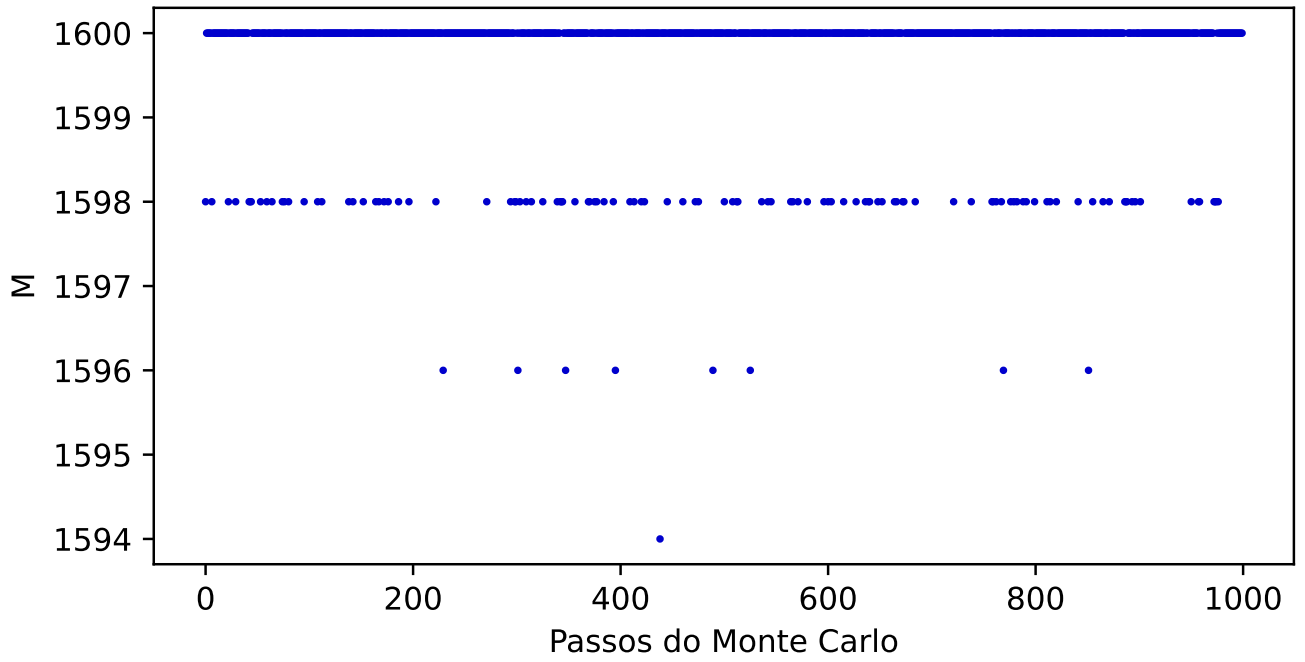


Figura 3: Gráfico da Magnetização em função dos Passos de Monte Carlo para a temperatura $T = 0,84750 \dots$

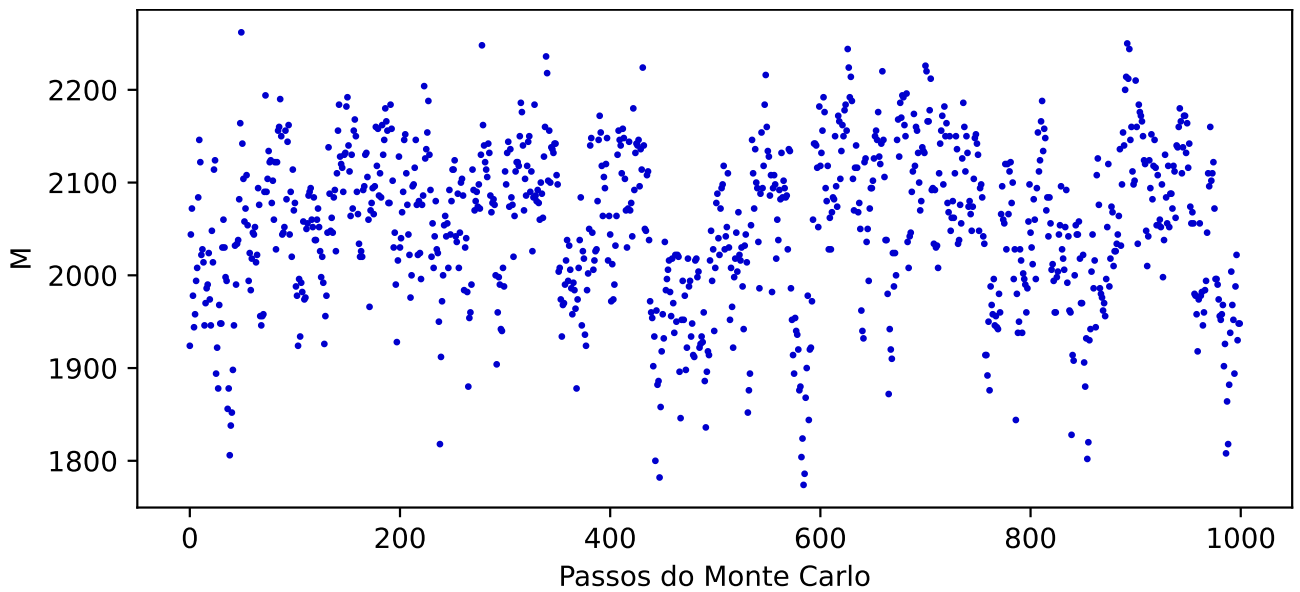


Figura 4: Gráfico da Magnetização em função dos Passos de Monte Carlo para a temperatura $T = 2,16929 \dots$

da magnetização.

Ou seja, percebe-se que há uma instabilidade maior na região da temperatura crítica, sendo necessário um número maior de passos de Monte Carlo para que o valor convirja, quando comparado com temperaturas menores e maiores que a temperatura crítica.

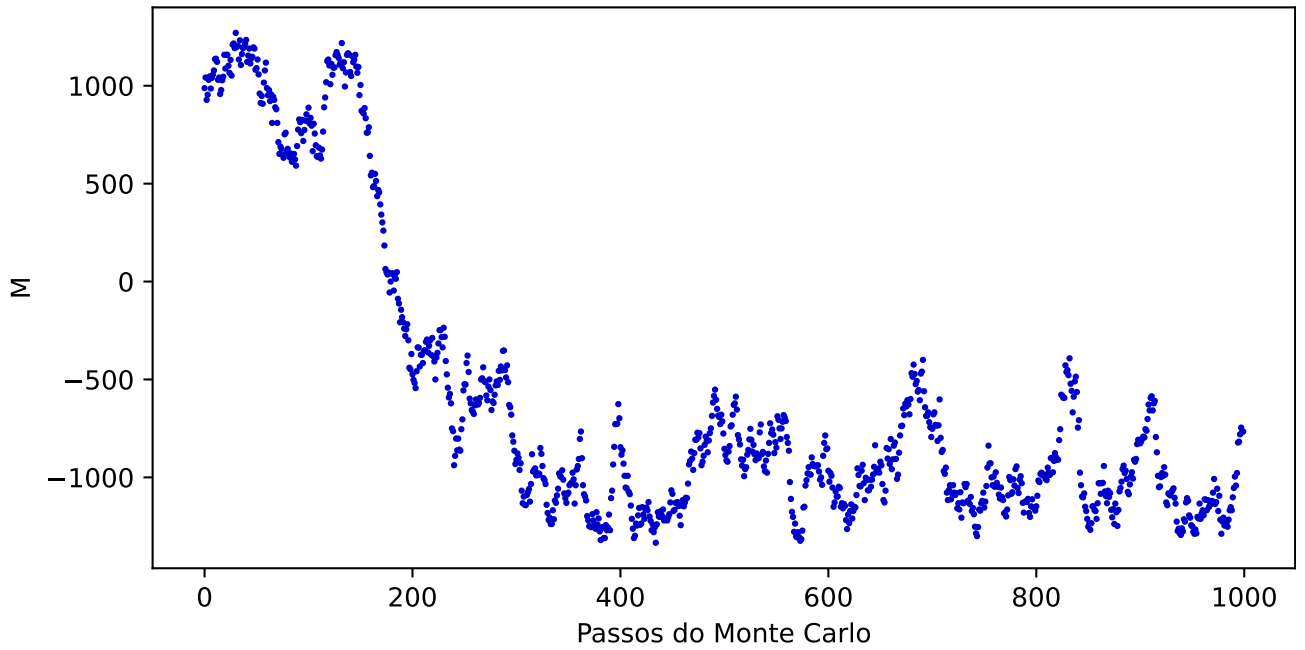


Figura 5: Gráfico da Magnetização em função dos Passos de Monte Carlo para a temperatura $T = 2,29965\dots$

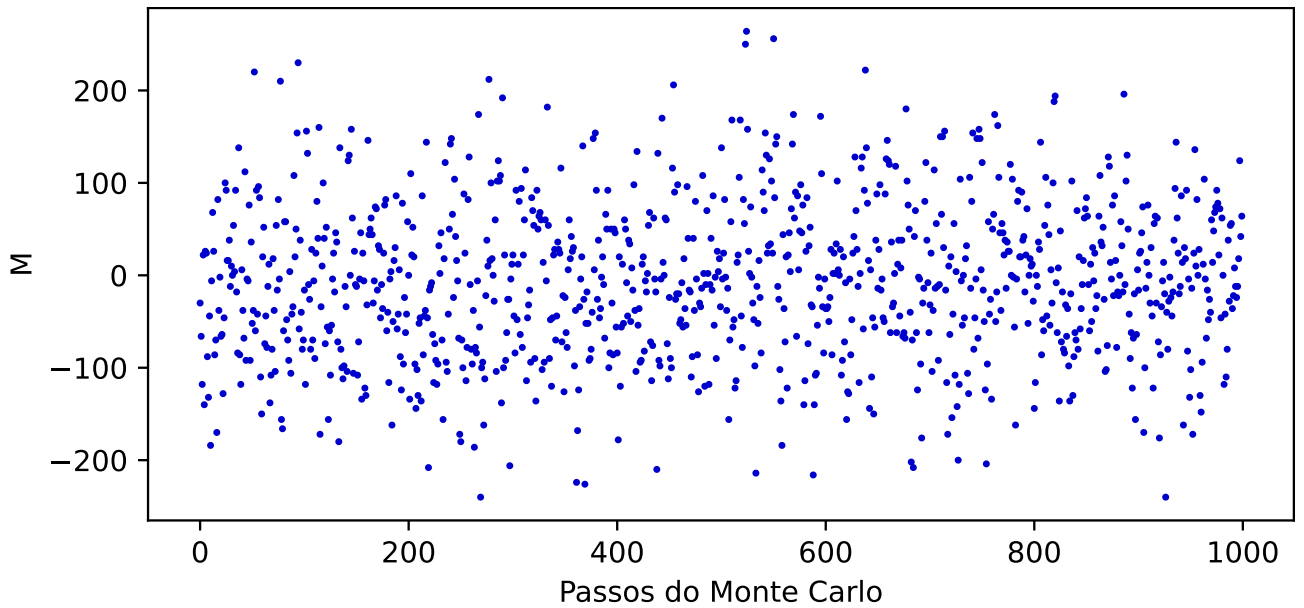


Figura 6: Gráfico da Magnetização em função dos Passos de Monte Carlo para a temperatura $T = 3,94303\dots$

4 Considerações Finais

Tendo em vista a forma como o gráficos do modelo descrito pela equação (5) se sobrepõem para magnetizações positivas, podemos concluir que o modelo feito pelo algoritmo presente na seção 2 representa bem o modelo de Ising, porém o valor da magnetização não vai a zero de fato quando a temperatura crítica é atingida, conforme mostra a figura 2.