

Paralelização do modelo de Ising utilizando OpenMP



Aluno: Caetano Slaviero Pires
Orientador Heitor C. M. Fernandes

Modelo de Ising

O modelo de Ising é descrito através da interação de *spins* de Ising com seus vizinhos, onde os *spins* atingem estados binários 1(up) e -1(down). A energia do sistema é calculada a partir de

$$\mathcal{H} = -J \sum_{\langle ij \rangle} s_i s_j - \mu B \sum_i s_i \quad (1)$$

onde a primeira soma é sobre todos *spins* i e sobre os vizinhos j do *spin* i .

A evolução do sistema foi feita a partir do Algoritmo de Metropolis, onde calcula-se a variação de energia devido à mudança de estado de um spin. A mudança de estado é aceita através da regra

$$A(\mu \rightarrow \nu) = \begin{cases} \exp\left(-\frac{\Delta E}{k_B T}\right), & \text{se } \Delta E > 0 \\ 1, & \text{caso contrario.} \end{cases} \quad (2)$$

Se $\Delta E > 0$, gera-se um número aleatório que deve ser menor do que $\exp\left(-\frac{\Delta E}{k_B T}\right)$ para a transição ser aceita.

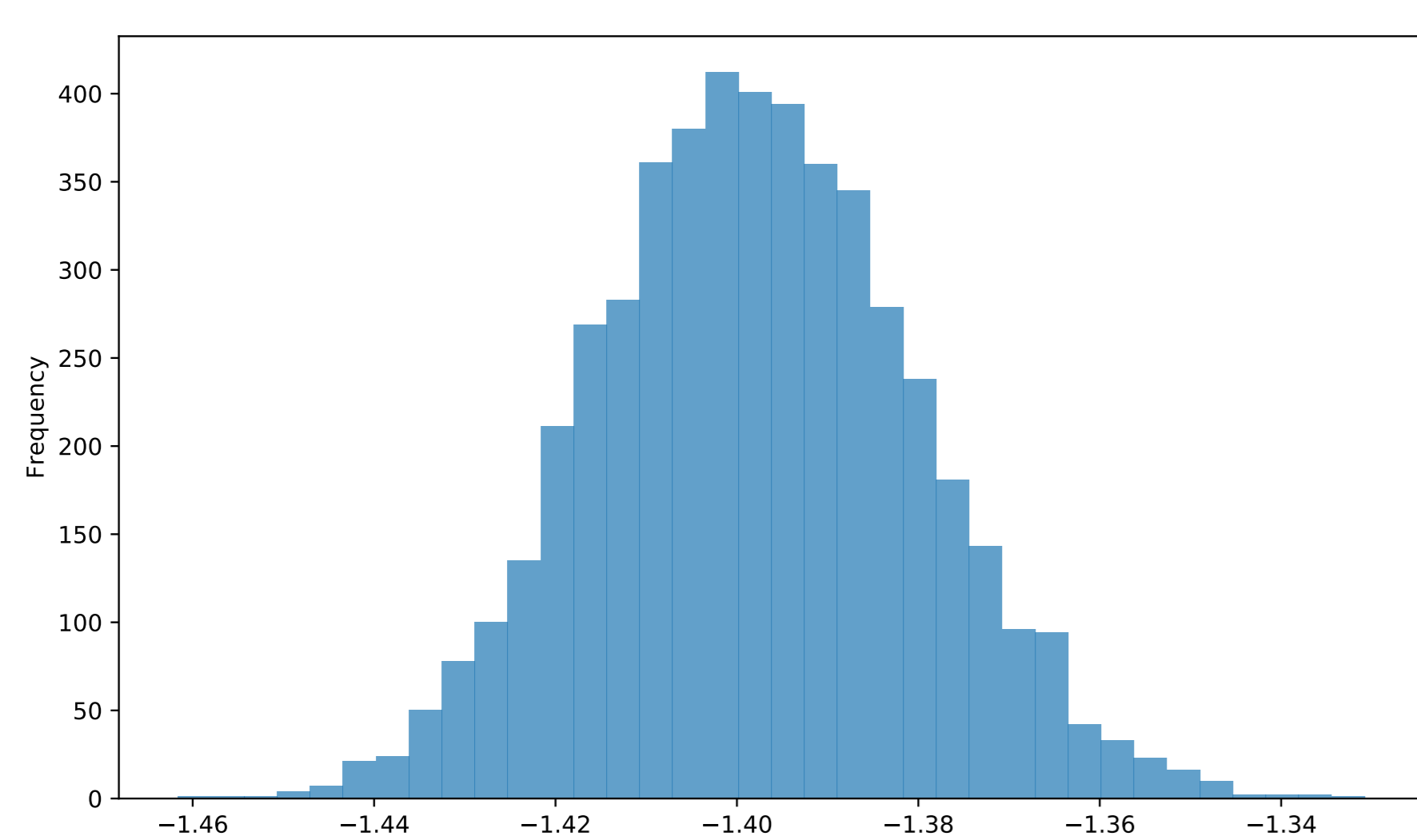


Figura 1: Densidade de estados de energia média para uma rede 100x100 na temperatura crítica $T=2.2691J$ gerada pelo algoritmo paralelo em 2×10^6 passos de Monte Carlo

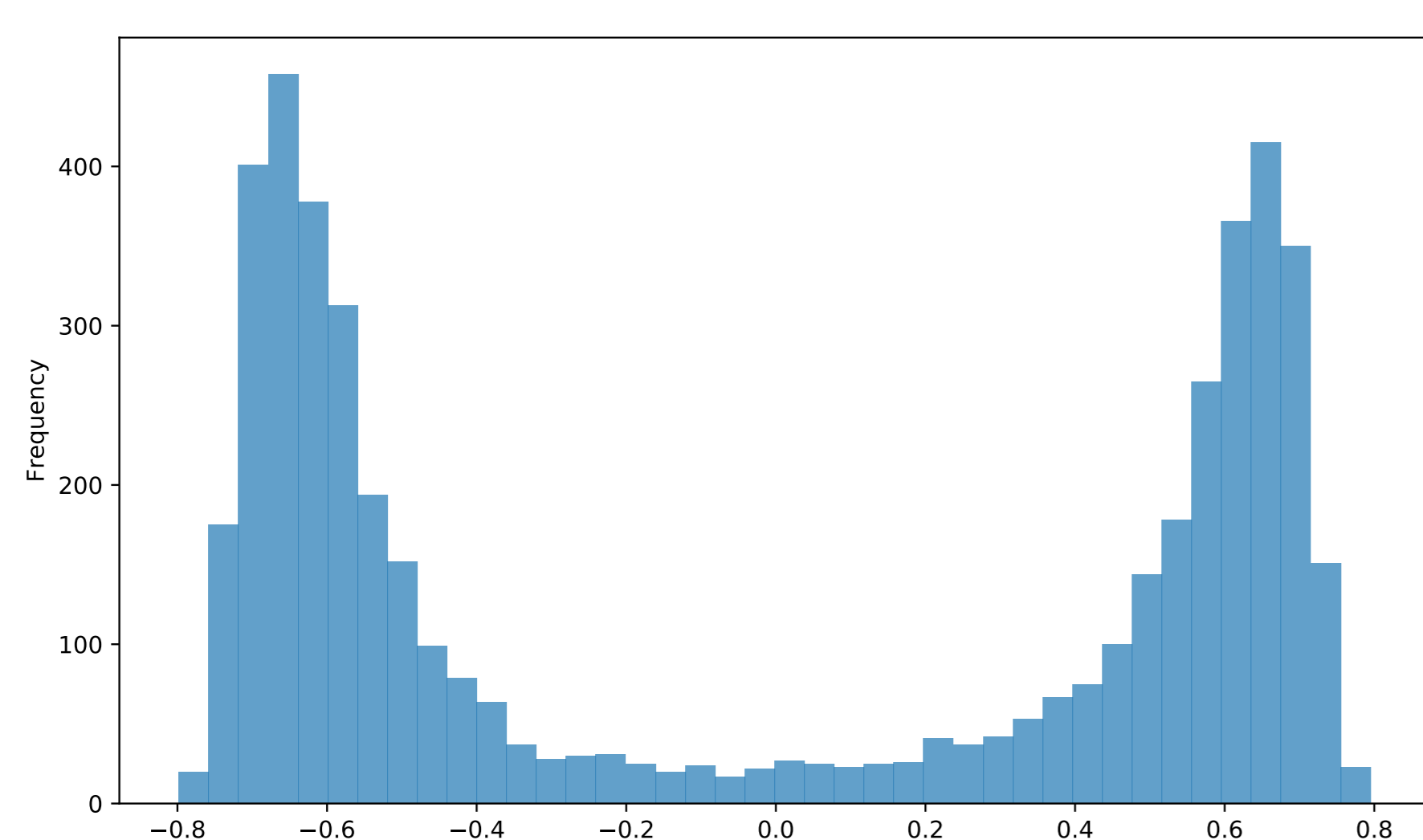


Figura 2: Densidade de estados de magnetização total para uma rede 100x100 na temperatura crítica $T=2.2691J$ gerada pelo algoritmo paralelo em 2×10^6 passos de Monte Carlo

Resultados da paralelização

As simulações em paralelo foram feitas em um notebook com CPU Intel Core i5-7200U 2.50GHz e GPU NVIDIA GeForce 940MX. Foram simuladas redes de tamanho 100×100 , 200×200 , 300×300 , 400×400 e 500×500 para *threads* 1,2,3 e 4 e medidos os tempos de simulação para cada passo de Monte Carlo. Estão tabelados abaixo tempos médios de simulação para 1 passo. As otimizações se mostram muito eficazes para redes grandes. 1 passo de Monte Carlo simulado com 4 *threads* em uma rede 500×500 leva cerca da metade do tempo de 1 passo utilizando 1 *thread* para a rede 300×300 .

Lado	Tempo para 1 passo ($10^{-2}s$)				
	Nº Threads	1	2	3	4
100		0.054	0.029	0.026	0.021
200		0.208	0.115	0.104	0.087
300		1.021	0.547	0.458	0.204
400					0.359
500					0.585

Tabela 1: Tempo médio de simulação para cada passo de Monte Carlo

As simulações em paralelo geram estados conforme uma distribuição Gaussiana (figura 1), possuindo transição de fase na temperatura crítica $T_c = 2.2691$ (figura 2). A transição de fase do modelo de Ising não-paralelizado[1] na figura 3 foi simulada em uma rede 100×100 com 10^5 passos e 1 *thread*. A simulação foi refeita para a mesma rede com 10 vezes mais passos e 4 *threads*, obtendo-se resultados mais precisos, conforme a figura 4.

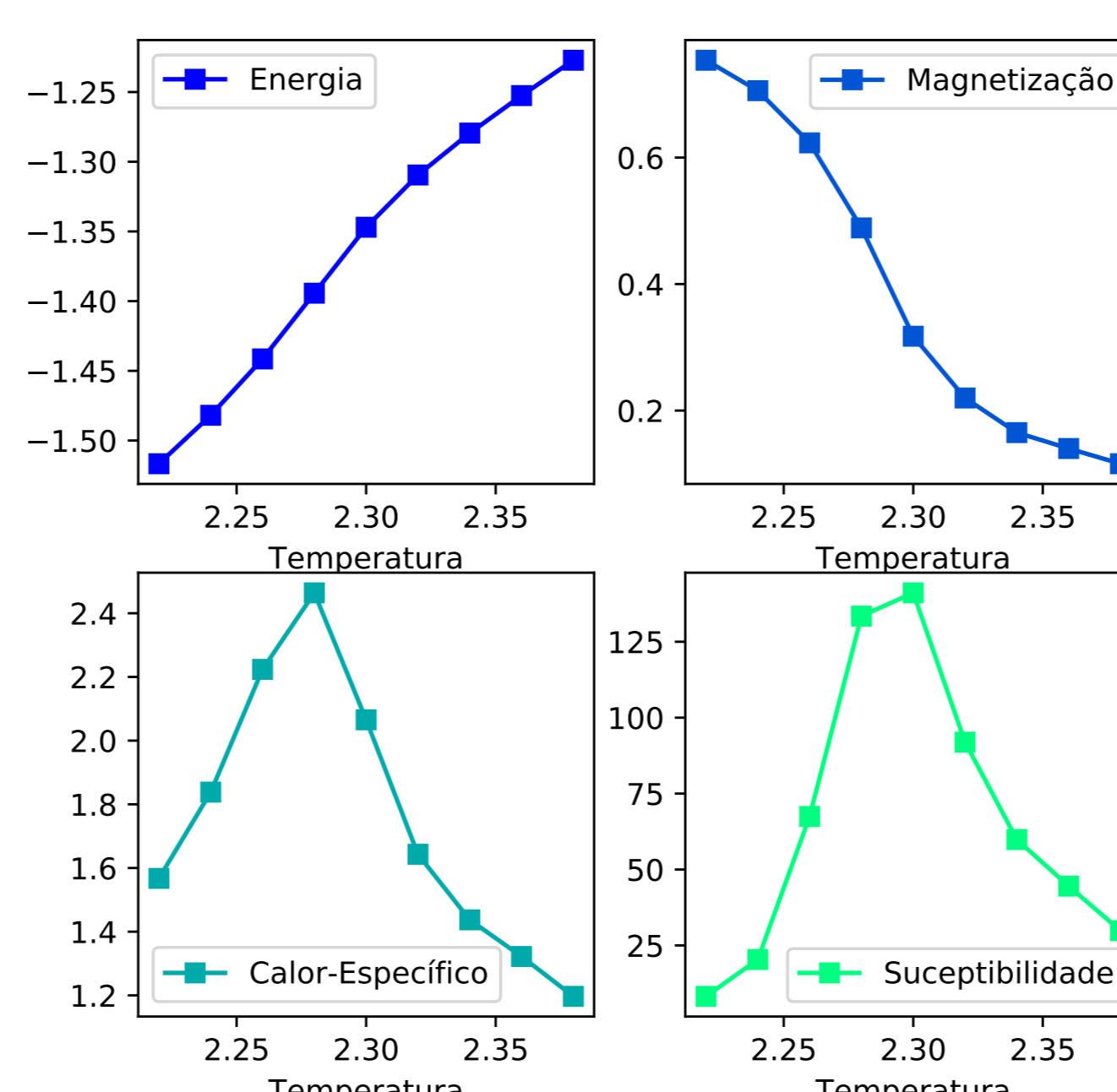


Figura 3: Transição de fase para uma rede de tamanho 100×100 em 10^6 passos de Monte Carlo com 4 *threads*

Detalhes da paralelização

O sistema de rede quadrada com vizinhança de *von Neumann* foi paralelizado de forma *multi-thread* utilizando o *OpenMP*[2] através da decomposição por *checkerboard*[3], onde são atualizados em paralelo apenas spins não vizinhos. Foi utilizado o gerador *gm61* para gerar os números pseudo aleatórios e aplicada a biblioteca *RNGAVXLIB*[4] para que os números fossem gerados de forma segura para *multi-thread* através das instruções *AVX*.

Conclusões

As otimizações e as paralelizações foram feitas apenas utilizando ferramentas abertas e livres a partir da linguagem C, mostrando a possibilidade do uso de ferramentas desse tipo para o meio científico. Os conhecimentos estudados para a paralelização deste modelo computacional serão aplicadas em modelos menos triviais, como a difusão no modelo de gás de rede[5].

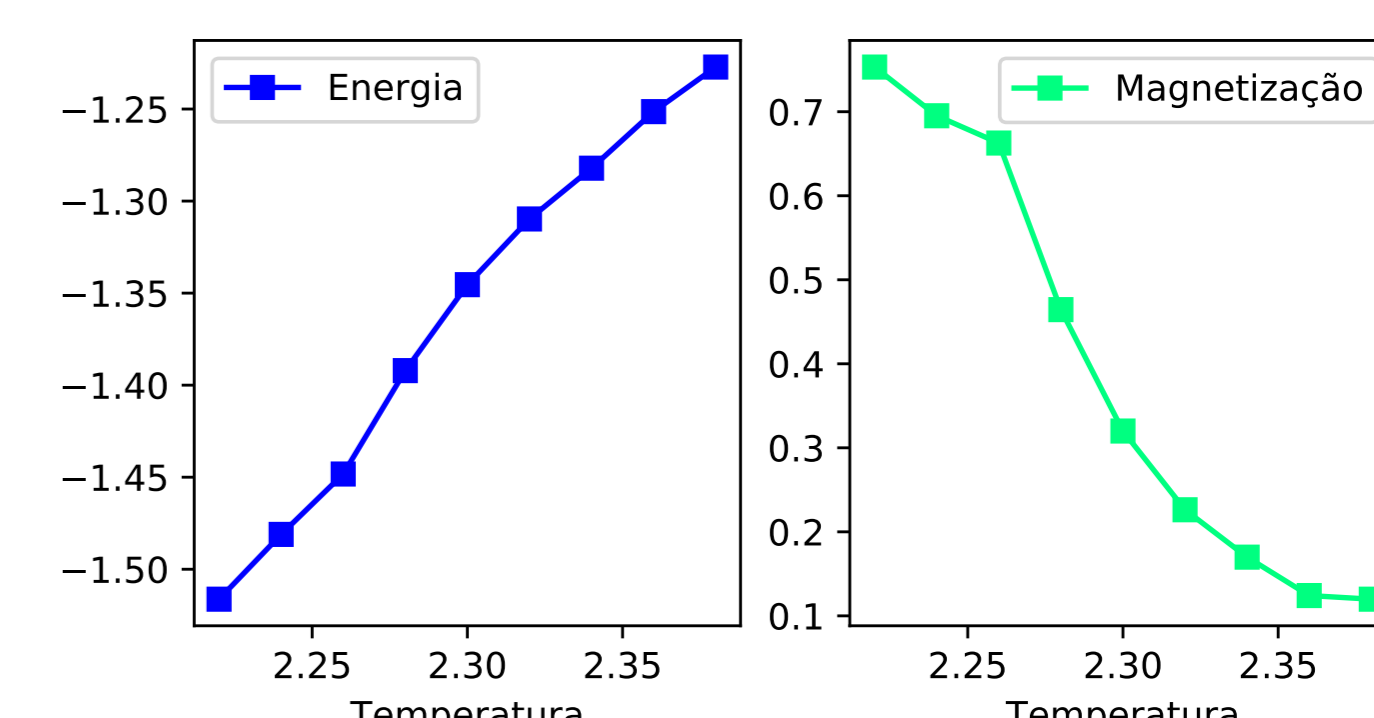


Figura 4: Transição de fase para uma rede de tamanho 100×100 em 10^5 passos de Monte Carlo com 1 *threads*

Referências

- [1] Anderson Rosa, Caetano Pires, and Lucas Doria. Modelo de ising 2d, 2017. [Online: https://fiscomp.if.ufrgs.br/index.php/Grupo_-_Ising_2D; acessado 05-Maio-2017].
- [2] Robit Chandra, Leonardo Dagum, Dave Kohr, Dror Maydan, Jeff McDonald, and Ramesh Menon. *Parallel Programming in OpenMP*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001.
- [3] Martin Weigel. Simulating spin models on gpu. *Computer Physics Communications*, 182(9):1833 – 1836, 2011. Computer Physics Communications Special Edition for Conference on Computational Physics Trondheim, Norway, June 23-26, 2010.
- [4] M.S. Guskova, L.Yu. Barash, and L.N. Shchur. *Rngavxlib: Program library for random number generation, avx realization*. *Computer Physics Communications*, 200:402 – 405, 2016.
- [5] Heitor C. M. Fernandes. *Gas de rede com exclusão de vizinhos*. PhD thesis, IF-UFRGS, 2007.