

João Paulo Peçanha Navarro de Oliveira

**Simulação Computacional de
Nanoestruturas Magnéticas Usando
Particionamento Espacial**

Juiz de Fora

2009

João Paulo Peçanha Navarro de Oliveira

**Simulação Computacional de
Nanoestruturas Magnéticas Usando
Particionamento Espacial**

Orientador:

Marcelo Bernardes Vieira

Co-orientadores:

Marcelo Lobosco

Sócrates de Oliveira Dantas

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Juiz de Fora

2009

Projeto submetido à disciplina Projeto Final em Computação (DCC-109) do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como parte integrante dos requisitos necessários para obtenção do grau de bacharel em Ciência da Computação.

Prof. Marcelo Bernardes Vieira, D. Sc.
Orientador

Prof. Marcelo Lobosco, D. Sc.
Co-Orientador

Prof. Sócrates de Oliveira Dantas, D. Sc.
Co-Orientador

Sumário

Resumo

1	Introdução	p. 6
1.1	Definição do Problema	p. 6
1.2	Objetivos	p. 7
2	Propriedades Magnéticas da Matéria	p. 8
2.1	O <i>spin</i> do átomo	p. 8
2.1.1	<i>Spin</i> de Partículas Elementares	p. 9
2.1.2	Direção do <i>spin</i>	p. 9
2.2	Momento Magnético Atômico	p. 10
2.3	Magnetização e Suscetibilidade Magnética	p. 13
2.4	Paramagnetismo	p. 14
2.5	Ferromagnetismo	p. 15
2.6	Transição de Fase e Temperatura de Curie	p. 17
2.7	Energia do Sistema	p. 18
3	Modelo Computacional	p. 20
3.1	Simulação Computacional	p. 20
3.1.1	Método de Monte Carlo	p. 21
3.1.2	Implementação do algoritmo de Metropolis	p. 21
3.2	Fundamentos	p. 22
3.3	Particionamento Espacial	p. 24

3.3.1	Octree	p. 25
3.4	Divisão de Tarefas Entre <i>Threads</i>	p. 28
3.4.1	Divisão de Tarefas Usando Planos	p. 29
3.5	Divisão de Tarefas Usando <i>Octree</i>	p. 31
4	Experimentos	p. 35
4.1	Testes de Desempenho	p. 37
5	Conclusão	p. 41
	Referências	p. 42

Resumo

O estudo da interação de spins em compostos e elementos magnéticos vem se mostrando uma importante área de pesquisa por permitir uma melhor compreensão das propriedades magnéticas da matéria. Uma das abordagens utilizadas neste estudo é o emprego de modelos físicos, que então são simulados computacionalmente. Contudo, existe uma natural complexidade na resolução numérica dos modelos físicos, relacionada principalmente com o número de elementos presentes na estrutura simulada. Esta complexidade nos leva a buscar soluções computacionais que auxiliem na redução do tempo de processamento gasto na simulação. Assim sendo, este trabalho apresenta um modelo físico que descreve o comportamento de uma classe especial de materiais magnéticos, os ferromagnéticos, e dois modelos computacionais paralelos. Esses modelos são baseados em múltiplos fluxos de execução e em memória compartilhada, usando o conceito de particionamento espacial do volume de simulação. Esta paralelização se mostra capaz de reduzir os custos associados aos cálculos relacionados a resolução numérica do modelo apresentado. Resultados preliminares indicam uma redução de até 44 vezes no tempo de execução, quando comparamos a versão sequencial do simulador com a sua versão paralela.

Palavras-chave: Simulação Física, Compostos Ferromagnéticos, Particionamento Espacial, Paralelização, *Threads*.

1 *Introdução*

O estudo e compreensão das origens das propriedades magnéticas da matéria são de extrema importância para o desenvolvimento de novas tecnologias e materiais, tendo aplicações em distintas áreas como saúde, transportes, telecomunicações e armazenamento de dados.

Este estudo, bem como sua aplicação, vem se mostrando uma importante área de investigação sobretudo no último século. Como exemplo podemos citar a descoberta recente da Magnetoresistência Gigante: uma inovação que abriu caminho para uma das primeiras aplicações da nanotecnologia. A descoberta deste fenômeno proporcionou um aumento da ordem de 100 vezes na capacidade de armazenamento dos discos rígidos. Com isso, os físicos Albert Fert e Peter Grünberg foram contemplados com o prêmio Nobel de Física no ano de 2007, consolidando, portanto, a importância de estudos nessa área.

Contudo, é importante ressaltar que a análise física do comportamento macroscópica dessas substâncias está diretamente associada com a forma no qual as suas menores partes interagem entre si. Para tal, um melhor entendimento destes fenômenos está relacionado com seu estudo em escala nanométrica. O estudo prático de sistemas com propriedades magnéticas em nanoescala muitas vezes pode ser inviável devido aos mais diversos fatores. Sendo assim, a criação de laboratórios virtuais constitui uma alternativa interessante para a realização de experimentos. Esses laboratórios são fundamentalmente ferramentas computacionais de simulação capazes de modelar e simular o comportamento dos sistemas em questão de forma muito realista.

1.1 **Definição do Problema**

Este trabalho visa, sobretudo, apresentar uma ferramenta computacional que seja capaz de realizar simulações inerentes às interações atômicas em classes de materiais magnéticos específicos: paramagnéticos e ferromagnéticos. Com essa finalidade este trabalho

abordará um modelo físico consistente e um modelo computacional já consolidado para simulação de fenômenos físicos em computadores: Método de Monte Carlo. Nesse contexto serão analisadas algumas limitações existentes e propostas possíveis melhorias referentes ao tempo de resolução numérica do modelo discutido.

1.2 Objetivos

O objetivo primário desta monografia é a apresentação e estudo de modelos físicos e Computacionais que descrevam o comportamento atômico em substâncias magnéticas de forma realista. Como objetivos secundários, oriundos do objetivo primário temos:

- Introduzir a conceituação física necessária para simular o comportamento de classes de sistemas magnéticos específicos;
- Utilizar o conceito de particionamento espacial para paralelizar os cálculos inerentes às interações interatômicas nos sistemas em questão;
- Apresentar métodos e algoritmos que ofereçam suporte para a manipulação geométrica de espaços tridimensionais compostos por estruturas magnéticas genéricas.

2 *Propriedades Magnéticas da Matéria*

Em física o termo magnetismo é usado para descrever a forma na qual os materiais respondem, em nível microscópico, a um campo magnético externo ou que já possuem algum magnetismo intrínseco. Descrever esse comportamento é de fundamental importância para classificar as substâncias em fases. Neste capítulo serão analisadas algumas dessas fases de materiais magnéticos e proposto um estudo sobre a interação interatômica nos mesmo.

Para compreender o comportamento dos materiais magnéticos, é necessário analisar a interação de seus componentes em escala microscópica para que generalizações macroscópicas possam ser tomadas. Sabe-se que toda matéria se compõe, fundamentalmente, de átomos e cada átomo se constitui de elétrons em movimento. É importante dizer que o magnetismo está intimamente ligado com esse movimento, pois, toda carga em movimento gera um campo magnético. Com isso podemos dizer que a mais importante fonte da magnetização são, especificamente, o movimento orbital angular do elétron ao redor do núcleo atômico e seu momento magnético intrínseco ($\vec{\mu}$). Neste capítulo também serão apresentados conceitos básicos sobre esses dois itens.

2.1 O *spin* do átomo

Em mecânica Quântica o *spin* (giro) é uma característica fundamental das partículas subatômicas (prótons, elétrons, nêutrons, etc.). Como o próprio nome sugere, o *spin* foi inicialmente concebido como o movimento de rotação das partículas elementares em torno de um determinado eixo. Como exemplo considere o elétron do átomo: todo elétron apresenta a propriedade intrínseca *spin*. Sabe-se que o *spin* do elétron está diretamente relacionada com o seu momento angular e admite-se que o *spin* pode ser interpretado como a quantização desse momento. Este momento é decorrente do movimento de rotação que

o elétron realiza em torno de si.

2.1.1 *Spin* de Partículas Elementares

Alguns estudos vêm mostrando que o *spin* presente nas partículas elementares não podem ser interpretados utilizando os conceitos da mecânica clássica. O momento de *spin* apresentado por estas partículas pode ser considerado uma propriedade física intrínseca bem como massa e carga elétrica. Em mecânica quântica, o momento angular de *spin* S de um sistema é dado por:

$$S = \hbar\sqrt{s(s+1)} \quad (2.1)$$

onde \hbar a constante de Plank e s um número inteiro ou fracionário positivo do tipo $\frac{n}{2}(0, \frac{1}{2}, 1, \frac{3}{2}, 2, \dots)$. O valor s dependerá exclusivamente do tipo de partícula em questão e é invariável ao longo do tempo. Veremos a seguir que a **direção do *spin*** pode adotar diferentes orientações (devido aos mais diversos fatores).

2.1.2 Direção do *spin*

Em mecânica clássica, uma determinada partícula com momento angular detém não somente magnitude (o quão rápido o corpo está em rotação), mas também direção. O componente de momento angular de *spin* em mecânica quântica pode tomar as seguintes direções (ao longo do eixo- z):

$$\hbar s_z, s_z = -s, -s+1, \dots, s-1, s, \quad (2.2)$$

sendo s o número quântico do *spin* discutido previamente. Note que existem $2s+1$ possibilidades para valores de s_z . O número $2s+1$ é chamado de multiplicidade do *spin* e indica quantas orientações o *spin* em questão poderá adotar. Por exemplo: por convenção os elétrons tem valor $s = \frac{1}{2}$, logo os possíveis valores para seu *spin* são: $s_z = +\frac{1}{2}$ ou $s_z = -\frac{1}{2}$. Neste caso específico o *spin* poderá se alinhar apenas com $+z$ e $-z$, que são freqüentemente referenciados como "*spin* para cima" e "*spin* para baixo".

Em muitas situações, as possíveis orientações que os *spins* podem adotar (apresentadas acima) são plausíveis para descrever, por exemplo, curvas de criticalidade em sistemas magnéticos, apresentando alta precisão. Contudo, existem ocasiões onde o grau de liberdade oferecido não é suficiente. Para tal, foi proposto um modelo onde é possível descrever o

spin como sendo um componente vetorial. Esse modelo de *spin* é conhecido como modelo de Heisenberg e foi apresentado no ano de 1928.

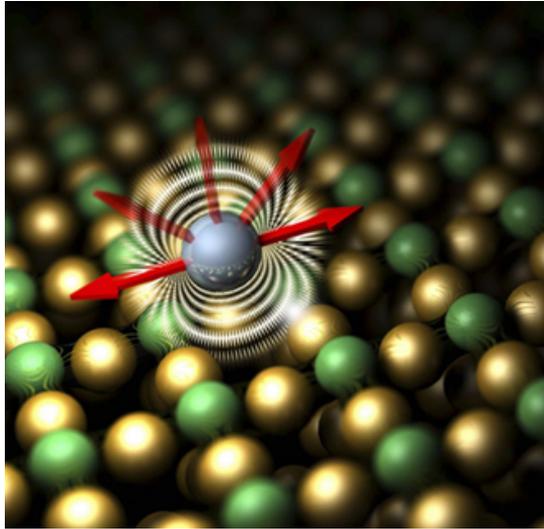


Figura 1: Ilustração da orientação do *spin* de um átomo (exibindo apenas algumas orientações, por simplicidade).

Portanto o *spin* do i -ésimo elemento localizado no sítio i pode ser expresso como:

$$\vec{S}_i = (S_i^x, S_i^y, S_i^z), \quad (2.3)$$

com $\vec{S}_i \in \mathbb{R}^3$ e, por conveniência, assumimos $|\vec{S}_i| = 1$.

Foi descrito até agora o momento de *spin* das partículas subatômicas. Entretanto, é de nosso interesse, principalmente, analisar o comportamento atômico (ou molecular) do material magnético em questão. Para isso, define-se o ***spin do átomo*** como sendo a soma vetorial dos *spins* de elétrons desemparelhados da substância (KITTEL, 1982). Desse ponto em diante, denotaremos \vec{S}_i como o *spin* resultante (ou apenas *spin*), do i -ésimo átomo pertencente a estrutura magnética analisada.

2.2 Momento Magnético Atômico

A magnetização de um material pode ser relacionada com os momentos magnéticos permanentes de cada átomo presente em sua estrutura. Esse momento magnético individual está diretamente associado com o movimento angular atômico. Considere uma partícula de massa m e carga q em movimento com velocidade v sobre um círculo com raio r como ilustrado na Figura 2. A magnitude do momento angular da partícula é dada

por:

$$L = mvr \quad (2.4)$$

A magnitude do momento magnético é o produto da corrente pela área do círculo (MARTINS, 1975):

$$\mu = IA = I\pi r^2 \quad (2.5)$$

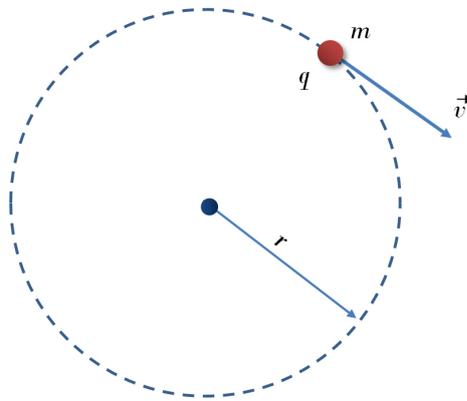


Figura 2: Partícula com carga q e massa m em movimento sobre uma circunferência de raio r .

Se T é o tempo necessário para a carga completar uma volta, a corrente será q/T . Uma vez que o período T é a distância $2\pi r$ dividido pela velocidade v , a corrente será:

$$I = \frac{q}{T} = \frac{qv}{2\pi r} \quad (2.6)$$

O momento magnético será, portanto:

$$\mu = IA = \frac{qv}{2\pi r} \pi r^2 = \frac{1}{2} qvr \quad (2.7)$$

Usando $vr = L/m$ da 2.4, teremos o momento magnético sendo:

$$\mu = \frac{q}{2m} L \quad (2.8)$$

Ou, para grandezas vetoriais:

$$\vec{\mu} = \frac{q}{2m} \vec{L} \quad (2.9)$$

A Equação 2.9 reporta a relação clássica entre os momentos angular e magnético do átomo (MARTINS, 1975; TIPLER; MOSCA, 2007). Como mencionado, toda carga em movimento gera um campo magnético. Se os elétrons dos átomos estão em constante movimento de rotação então a estes também podemos associar um momento magnético.

Uma vez que o momento angular é quantizado, o momento magnético de um átomo também é quantizado (TIPLER; MOSCA, 2007). Temos que $\hbar = h/2\pi$ representa a quantização do momento angular onde h é a constante de Plank. Podemos escrever o momento magnético em termos de \vec{L}/\hbar como:

$$\vec{\mu} = \frac{q\hbar}{2m} \frac{\vec{L}}{\hbar} \quad (2.10)$$

Para um elétron com massa $m = m_e$ e carga $q = -e$, seu momento magnético devido ao movimento orbital será:

$$\vec{\mu}_l = \frac{e\hbar}{2m_e} \frac{\vec{L}}{\hbar} = -\mu_B \frac{\vec{L}}{\hbar} \quad (2.11)$$

onde

$$\mu_B = \frac{e\hbar}{2m_e} \quad (2.12)$$

é chamado de **Magneton de Bohr** (TIPLER; MOSCA, 2007). O momento magnético do i -ésimo elétron devido ao seu momento angular de *spin* \vec{S}_i é:

$$\vec{\mu}_s = -2 \times \frac{e\hbar}{2m_e} \frac{\vec{S}_i}{\hbar} = -2\mu_B \frac{\vec{S}_i}{\hbar} \quad (2.13)$$

A Equação 2.13 reporta o momento magnético devido ao *spin* do elétron. Note que o fator multiplicativo 2 da Equação 2.13 é resultante da equivalência entre o momento angular e o *spin* oriundo da teoria quântica, não existindo portanto, um análogo na mecânica clássica (TIPLER; MOSCA, 2007).

2.3 Magnetização e Suscetibilidade Magnética

Vimos até agora que cada porção indivisível da matéria são fontes de magnetização devido ao movimento de suas partículas elementares. É natural pensar que a magnetização total de um objeto será as contribuições dos momentos magnéticos ou de *spin* de cada um desses elementos. Considere um determinado material magnetizado. Vamos definir agora uma quantidade macroscópica: a *magnetização* \vec{M} . Esta é definida como a soma vetorial de todos os *spins* presentes em um pequeno elemento de volume Δv , e então o resultado é dividido por Δv ; a quantidade resultante (REITZ; MILFORD; CHRISTY, 1982)

$$\vec{M} = \lim_{\Delta v \rightarrow 0} \frac{1}{\Delta v} \sum_i \vec{S}_i \quad (2.14)$$

é conhecida como **magnetização**. É importante ressaltar que o volume Δv é sempre pequeno com relação às dimensões do material em questão, porém, grande quando comparado com as dimensões atômicas (MARTINS, 1975). No estado desmagnetizado, a soma $\sum \vec{S}_i$ retorna valor zero como resultado da orientação aleatória dos *spins* \vec{S}_i , porém na presença de um campo externo de excitação, ou campo magnético externo (\vec{B}), o valor para \vec{M} dependerá dessa influência externa. Veremos que essa característica é muito importante para a verificação de transição de fases em materiais magnéticos.

Para resolver problemas do magnetismo, como classificação de materiais, é essencial haver uma relação entre campo externo aplicado (\vec{B}) e magnetização (\vec{M}). Essa relação é de chamada de **suscetibilidade magnética** e mede, de uma maneira geral, o grau de influência que o campo magnético externo exerce sobre as mínimas partes do material. A suscetibilidade pode ser expressa em termos de χ_m , sendo que este é um número adimensional:

$$\chi_m = \frac{M}{B} \quad (2.15)$$

Com esse conceito podemos definir as fases que os materiais magnéticos podem exibir. Como dito, toda matéria apresenta propriedades magnéticas quando sob a ação de um campo magnético. Entretanto cada material reage ao campo magnético de uma forma, de acordo com as características físicas de seus átomos e das interações entre estes, o que permite classificá-los conforme a sua suscetibilidade em:

- Ferromagnético e ferrimagnética ($\chi_m \gg 1$);

- Diamagnética ($\chi_m < 1$);
- Paramagnética ($\chi_m > 0$);
- Antiferromagnética (χ_m pequeno);
- Materiais supercondutores ($\chi_m = -1$).

Definições retiradas de (NETO, 2004). Em especial, neste trabalho será analisado o comportamento dos materiais paramagnéticos e ferromagnéticos.

2.4 Paramagnetismo

Os materiais **paramagnéticos** são os que apresentam suscetibilidade magnética positiva, muito pequena. O paramagnetismo é caracterizado fundamentalmente por: a) apresentar átomos com momentos magnéticos permanentes, b) possuir uma interação inter-atômica muito fraca e c) ter os *spins* dos átomos orientados ao acaso.

Na presença de um campo magnético externo, os *spins* tendem a se alinhar com o campo, mas essa tendência é contrariada pela tendência de os momentos terem orientações randômicas em virtude da agitação térmica. Portanto, a quantidade de *spins* alinhados com o campo magnético externo dependerá exclusivamente da intensidade do campo aplicado e da temperatura do sistema. Em temperaturas muito baixas a oscilação molecular pode ser considerada pequena, portanto as orientações randômicas dos *spins* dessas moléculas são amenizadas e quando sobre ação de um campo magnético aplicado, esses *spins* tendem a se alinhar quase que completamente ao campo.

A energia potencial de um *spin* \vec{S}_i pertencente a uma estrutura magnética qualquer sob ação de um campo magnético externo \vec{B} é dada por:

$$U_i = -S_i B \cos \theta = -\vec{S}_i \cdot \vec{B} \quad (2.16)$$

Note que, quanto mais alinhado com o campo ($\theta \rightarrow 0^\circ$) menor será a energia potencial entre o *spin* do átomo e o campo. Perceba também que essa energia de troca tende a aumentar quanto mais antiparalelo o *spin* estiver do campo magnético vetorial externo ($\theta \rightarrow 180^\circ$), e como a tendência natural de todo sistema físico é de buscar um estado energético mais favorável, ou de equilíbrio (entenda como menor energia), até mesmo nas substâncias paramagnéticas os *spins* dos átomos tendem a se alinhar com o campo visando

a diminuição da energia do sistema (esse comportamento é observado em menor escala devido sua baixa suscetibilidade). Vale frisar que a Equação 2.16 também é válida para descrever a interação *spin*-campo em materiais ferromagnéticos.

A expressão que descreve o comportamento da magnetização interna em um paramagneto devido ao campo externo e temperatura é dada pela Lei de Curie:

$$M = \frac{1}{3} \frac{m B_{ap}}{kT} M_s \quad (2.17)$$

onde B_{ap} representa o módulo do campo externo aplicado, k a constante de *Boltzman*, T a temperatura do sistema, m a energia do dipolo magnético analisado e M_s a energia de saturação do sistema. Note que a temperatura é inversamente proporcional à magnetização, portanto, altas temperaturas tendem a diminuir a magnetização interna, o que significa uma tendência de aleatoriedade nas direções dos *spins* dos átomos. Para que M se aproxime do valor de saturação M_s o campo aplicado B_{ap} deve ser suficientemente intenso para anular o efeito da temperatura T . Outro aspecto interessante é o completo estado de randomicidade que os *spins* do átomos apresentarão quando sobre nenhuma influência externa, ou seja, quando $B_{ap} = 0 \rightarrow M = 0$, lembrando que valores de magnetização M próximos de zero indica um maior nível de aleatoriedade que nas orientações dos *spins* e dipolos da substância, como previamente citado.

2.5 Ferromagnetismo

Os materiais **ferromagnéticos** são os que têm valores positivos, muito grandes, da suscetibilidade magnética χ_m . O ferromagnetismo se manifesta, por exemplo, no ferro, no cobalto e no níquel, puros, e nas ligas destes elementos entre si. As substâncias ferromagnéticas se distinguem das paramagnéticas pelo fato da magnetização adquirida pelo material não ser proporcional ao campo magnético que a produz. As propriedades magnéticas de uma amostra ferromagnética dependem tanto do campo externo aplicado quanto de todos os estados anteriores pelos quais o sistema passou. Esse tipo de material detém uma propriedade muito importante que é ter a capacidade de apresentar magnetização mesmo na ausência de um campo magnético externo.

Um material ferromagnético possui momento magnético espontâneo – mesmo em um campo magnético nulo. Um momento espontâneo sugere que os *spins* e os momentos magnéticos se distribuem de maneira regular ao longo de toda estrutura (KITTEL, 1982). Nesses materiais existem grandes interações entre os *spins* dos átomos. O resultado é tal

que um grande número de *spins* tendem a terem suas orientações apontadas para uma mesma direção mesmo na ausência de um campo magnético externo. Ainda à temperaturas normais o alinhamento é tão forte que as vibrações térmicas não podem destruí-lo. Essa forte interação nas vizinhanças dos átomos dos ferromagnetos promove o que chamamos de **domínios magnéticos**. Estes domínios são regiões do material onde os dipolos estão alinhados no mesmo sentido, como mostrado na Figura 3.

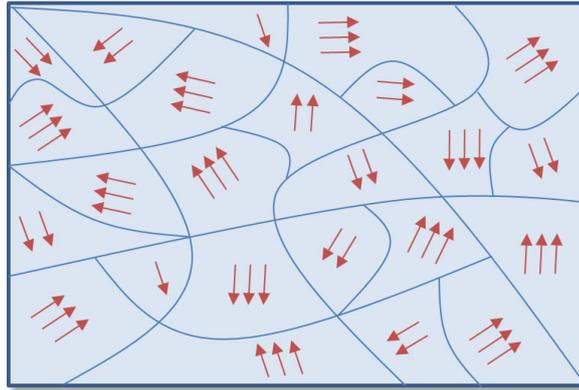


Figura 3: Domínios em um material ferromagnético: a forte interação de vizinhança promove um alinhamento local mesmo na ausência de campo magnético.

Note que são esses domínios que explicam o estado de magnetização nos ferromagnetos mesmo sob um campo magnético nulo: o alinhamento dentro dos domínios indica uma baixa grau de aleatoriedade nas orientações dos *spins*, e sabe-se que quanto mais ao acaso é esse direcionamento, mais o valor interno da magnetização tenderá a zero, sendo assim, a forte interação de vizinhança nos *spins* é responsável pelo estado de magnetização não nulo nas substâncias ferromagnéticas.

A expressão que define a interação de troca energética entre dois *spins* situados nos sítios i e j é dado por:

$$U_{ij} = -J \left[\vec{S}_i \cdot \vec{S}_j \right] \quad (2.18)$$

onde J (com $J > 0$) corresponde à intensidade de interação ferromagnética e dependerá exclusivamente do tipo do material em questão.

Contudo, estamos interessados na energia de interação entre um *spin* arbitrário i e seus respectivos vizinhos. Para esse cálculo são tomados os m vizinhos mais próximos onde m estará diretamente relacionado com o grau de liberdade oferecido: uma, duas ou três dimensões. Bem como a topologia do problema em estudo. Considere $n = 3$, logo

em um sistema cúbico simples (uma das topologias tridimensionais), $j \in \{1, 2, \dots, 6\}$, a energia potencial de interação resultante será:

$$U_i = -J \sum_{j=1}^m \vec{S}_i \cdot \vec{S}_j \quad (2.19)$$

A Equação 2.19 representa a interação clássica entre o *spin* de um átomo em um material ferromagnético e seus respectivos primeiros vizinhos, e é comumente referenciada como **energia ferromagnética**.

2.6 Transição de Fase e Temperatura de Curie

Consideramos a transição de fase em uma determinada substância como sendo a sua transformação de uma fase para outra. Uma característica que distingue uma transição de fase é uma abrupta mudança em uma ou mais propriedades físicas do sistema analisado. Em sistemas magnéticos, o estudo de transição de fase se concentra basicamente na análise comportamental de materiais ferromagnéticos sob as mais diversas condições. Especialmente, consideramos que as oscilações termodinâmicas influenciam diretamente no comportamento magnético do sistema.

A certa temperatura os ferromagnetos apresentam um alinhamento local de seus *spins* promovido pela forte interação de vizinhança, e que é matematicamente representada pela expressão 2.19. Considere o sistema magnético em questão como sendo um sólido fechado e limitado e que não sofre nenhuma influência externa que não seja a temperatura. Se promovermos um aquecimento nesse sistema, é natural de se pensar que a oscilação atômica será proporcional à nova temperatura imposta. Esse movimento de vibração pode promover um desarranjo nos domínios magnéticos internos. Isso ocorre, pois, a orientação dos *spins* está relacionada com a agitação térmica dos átomos, portanto quanto mais intenso for esse movimento oscilatório, maior será a tendência desses *spins* adotarem direções aleatórias.

Quando os domínios magnéticos são quebrados, os *spins* adotam direcionamentos ao acaso, descaracterizando portanto, o comportamento usual de um material ferromagnético. Note que estruturas cujos *spins* estão orientados ao acaso são características fundamentais dos paramagnetos, e sendo assim, a uma determinada temperatura, consideramos que o sistema passou por uma **transição de fase**, onde o material ferromagnético passa a se comportar como um paramagnético. A temperatura crítica do sistema, é chamada

de **temperatura de Curie** (CAO et al., 2007). Logo, abaixo desse limiar o sistema pode ser considerado ferromagnético, e acima, paramagnético.

2.7 Energia do Sistema

Os pontos críticos (onde ocorre transição de fase) de sistemas ferromagnéticos sobre constante aquecimento podem variar. Esta variação está diretamente relacionada com fatores inerentes ao sistema como forma do objeto em questão (disposição espacial dos átomos e forma geométrica da estrutura), influências externas como temperatura e campo magnético, quantidade de *spins*, dentre outros. Os valores da temperatura de Curie do sistema são evidenciados principalmente pela análise comportamental das diversas propriedades físicas do sistema: a) calor específico, b) variação da energia, c) magnetização interna e d) a própria suscetibilidade magnética. Para que estes sejam calculados se faz necessário o cálculo da **energia total do sistema**.

A energia total do sistema, ou apenas **energia do sistema**, é dada, em uma de suas abordagens, pela soma das energias de interação entre os átomos pertencentes a esse sistema. Para tal, será necessário conhecer todos os elementos do sistema, portanto será apresentado o cálculo da energia potencial do sistema em sua versão discreta.

Já foi discutido previamente os aspectos de interação interatômica de curto alcance (Eq. 2.19) e de interação *spin*-campo (Eq. 2.16) em materiais ferromagnéticos. Contudo, para sistemas físicos constituídos de dipolos magnéticos, existe um tipo de interação específica de longo alcance e que também pode ser considerada. A energia dipolar entre dois dipolos i e j situados em posições \vec{r}_i e \vec{r}_j , respectivamente, é dada por:

$$\omega_{ij} = \frac{\vec{S}_i \cdot \vec{S}_j}{|\vec{r}_{ij}|^3} - 3 \frac{(\vec{S}_i \cdot \vec{r}_{ij})(\vec{S}_j \cdot \vec{r}_{ij})}{|\vec{r}_{ij}|^5} \quad (2.20)$$

A energia total de um sistema geralmente é referenciada com o seu **Hamiltoniano**. O Hamiltoniano que descreve o comportamento energético de um sistema ferromagnético em função de seus *spins* e suas respectivas posições, utilizando as interações vistas até agora (dipolo-dipolo (Eq. 2.20), interação com o campo magnético (Eq. 2.16) e energia ferromagnética (Eq. 2.19)), é dado por:

$$\mathcal{H} = A \sum_{\substack{i,j \\ i>j}}^n \omega_{ij} - J \sum_{\substack{i,k \\ i \neq k}}^n \vec{S}_i \cdot \vec{S}_k - \sum_{i=1}^n D(\vec{S}_i \cdot \vec{B}) \quad (2.21)$$

onde \vec{S}_i é o clássico *spin* de Heisenberg na posição $\vec{r}_i = (x_i, y_i, z_i)$ e $\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$ é o vetor posição que separa os átomos i e j . As constantes A , J e D correspondem às intensidades das interações dipolares, ferromagnéticas e campo externo, respectivamente. O segundo termo refere-se a interação ferromagnética entre os átomos.

3 *Modelo Computacional*

3.1 Simulação Computacional

Para se realizar uma simulação computacional, o potencial de interação adotado pode ser simples, como no caso do potencial gravitacional em interações de partículas cósmicas, ou possuir mais de um termo, tal qual na descrição de interações entre átomos e moléculas. Dessa forma, a simulação computacional nos permite extrair várias informações do sistema como um todo, permitindo, assim, o cálculo de diversas propriedades (SABINO et al., 2009).

A modelagem computacional nos oferece não apenas formas de resolver cálculos matemáticos, mas também e, sobretudo, a criação de laboratórios virtuais no qual estudos podem ser realizados de forma muito próxima à realidade. Assim, a simulação em ambientes computacionais permite que pesquisadores possam ser capazes de entender o comportamento da matéria em uma escala que não pode ser facilmente observada, ou ainda em situações físicas impossíveis de serem alcançadas em laboratórios (como, por exemplo, altas pressões ou temperaturas, número de partículas envolvidas, etc.). Dessa forma, a compreensão e a observação de um problema podem ser feitas de forma fidedigna e eficaz (SABINO et al., 2009).

Neste trabalho é de interesse a análise comportamental da interação atômica – mais especificamente, da interação entre os *spins* dos átomos – em materiais ferromagnéticos. Para simular o comportamento desse tipo de sistema, um método já consolidado para simulações físicas: método de Monte Carlo utilizando o algoritmo de Metrópolis (MMC-AM). Sabemos que o comportamento dos ferromagnetos pode variar dados fatores como campo magnético externo e temperatura do ambiente. O MMC-AM consegue reproduzir essa interação alcançando resultados numéricos satisfatórios usando como base a energia total do sistema que foi definida segundo a Equação 2.21. Alguns detalhes do método serão discutidos a seguir.

3.1.1 Método de Monte Carlo

O método de Monte Carlo (MMC) obtém aproximações numéricas para funções complexas através da conversão de um modelo físico ou matemático em um modelo estatístico. Os resultados são gerados sobre uma distribuição de probabilidades e a amostra obtida é usada para aproximar a função de interesse.

O MMC foi desenvolvido por Nicholas Metropolis e Stanislaw Ulam (METROPOLIS et al., 1953) durante a segunda guerra mundial para estudar a difusão de nêutrons durante o fenômeno de fissão nuclear. Em 1953, Nicholas Metropolis desenvolveu o algoritmo que leva seu nome com base no MMC a fim de estudar o comportamento estatístico dos gases. Desde então este algoritmo é muito utilizado para o estudo de vários sistemas físicos, buscando determinar propriedades de interesse.

De maneira geral, o algoritmo de Metropolis determina valores esperados de propriedades de um sistema em simulação calculando-se uma média sobre uma amostragem, que é obtida através da geração de números aleatórios. O algoritmo é concebido de modo a se obter uma amostragem que siga a distribuição de Boltzmann (METROPOLIS et al., 1953). Assim, o sistema a ser simulado deve se encontrar em temperaturas diferentes de zero e o valor da energia para cada partícula do sistema deve ser conhecido.

3.1.2 Implementação do algoritmo de Metropolis

De maneira geral, o MMC-AM é descrito da seguinte forma: a cada iteração, um número k é sorteado aleatoriamente. A partir daí o k -ésimo átomo do sistema tem a orientação de seu *spin* alterada também aleatoriamente. Para que esta interação de *spin* seja aceita o sistema deve satisfazer a seguinte probabilidade:

$$W(\{S_k^a\} \rightarrow \{S_k^b\}) = \begin{cases} e^{(-\Delta\mathcal{H}/Kt)} & \Delta\mathcal{H} > 0 \\ 1 & \Delta\mathcal{H} \leq 0 \end{cases} \quad (3.1)$$

portanto, para que o sistema transite do estado a para o estado b e conseqüentemente o novo *spin* gerado para o átomo k seja aceito, a condição da Equação 3.1 deve ser satisfeita. Com onde $\Delta\mathcal{H} = \mathcal{H}_b - \mathcal{H}_a$, ou seja a diferença de energia do sistema com a nova orientação do k -ésimo *spin* em relação à energia do sistema com a orientação antiga (YEOMANS, 1992), (JANKE, 1998). O fluxograma na Figura 4 detalha os passos necessários para a aceitação, ou não, de uma nova configuração de *spin* (PEÇANHA et al., 2008).

Uma etapa importante do algoritmo é a geração de números aleatórios. Para garantir

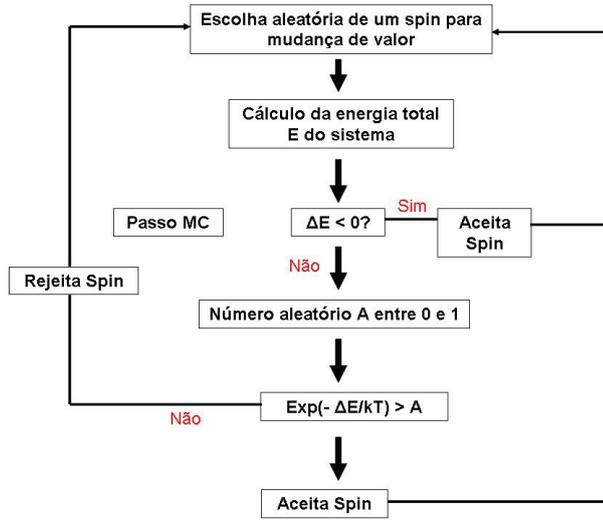


Figura 4: Fluxograma do algoritmo de Metropolis onde ΔE é a variação de energia obtida (antes e após a mudança do *spin*).

que a probabilidade de sortear cada átomo seja uniforme, foi utilizado o algoritmo *Mersenne Twister* cuja periodicidade é de 2^{19937} (MATSUMOTO; NISHIMURA, 1998). O átomo escolhido recebe um novo *spin* distribuído aleatoriamente no espaço de vetores unitários \mathbb{S}^2 .

3.2 Fundamentos

Seja \mathbb{U} um espaço de simulação discreto e finito provido de n átomos onde $|\mathbb{U}| = n$. Logo, $\mathbb{U} = \{a_1, \dots, a_n\}$ onde o i -ésimo átomo $a_i \in \mathbb{U}$ é um vetor $a_i = (\vec{r}_i, \vec{S}_i)$ tal que \vec{r}_i é a posição espacial $\vec{r}_i = (x_i, y_i, z_i)$ com $a_i \neq a_j$ se e somente se $\vec{r}_i \neq \vec{r}_j$ para $\forall i, j \in \{1, 2, \dots, n\}$ onde $i \neq j$, e $\vec{S}_i = (S_i^x, S_i^y, S_i^z)$ é o *spin* da partícula i com $\vec{S}_i \in \mathbb{R}^3$.

Sabemos que a energia associada a um átomo $a_i \in \mathbb{U}$ é dada por:

$$\mathcal{E}_i = \frac{A}{2} \sum_{\substack{j=1 \\ i \neq j}}^n \omega_{ij} - J \sum_{\substack{k=1 \\ i \neq k}}^n \vec{S}_i \cdot \vec{S}_k - D(\vec{S}_i \cdot \vec{B}), \quad (3.2)$$

e ω_{ij} é definido por:

$$\omega_{ij} = \frac{\vec{S}_i \cdot \vec{S}_j}{|\vec{r}_{ij}|^3} - 3 \frac{(\vec{S}_i \cdot \vec{r}_{ij})(\vec{S}_j \cdot \vec{r}_{ij})}{|\vec{r}_{ij}|^5}. \quad (3.3)$$

onde \vec{S}_i denota o *spin* da partícula analisada, \vec{S}_j o *spin* de um átomo $a_j \in \mathbb{U}$ e $\vec{r}_{ij} = \vec{r}_i -$

\vec{r}_j é o vetor posição que separa os átomos i e j , desta forma $i \neq j$.

O espaço \mathbb{U} pode ser completamente particionado em subconjuntos que chamaremos de células. O conjunto de células do espaço \mathbb{U} será denotado por $\mathcal{P}(\mathbb{U})$, tal que $\mathcal{P}(\mathbb{U}) = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_l\}$, tal que $|\mathcal{C}_l| = hl$. Contudo, para que esse conjunto de células seja viável, estas devem satisfazer a seguinte condição:

$$\begin{cases} \mathcal{C}_1 \cap \mathcal{C}_2 \cap \dots \cap \mathcal{C}_l = 0 \\ \mathcal{C}_1 \cup \mathcal{C}_2 \cup \dots \cup \mathcal{C}_l = \mathbb{U} \end{cases} \quad (3.4)$$

onde, $\mathcal{C}_1 = \{a_1, a_2, \dots, a_{h1}\}$, $\mathcal{C}_2 = \{a_{h1+1}, a_{h1+2}, \dots, a_{h2}\}$, ..., $\mathcal{C}_l = \{a_{(h(l-1))+1}, a_{(h(l-1))+2}, \dots, a_{hl}\}$, para células com número de átomos não nulo e arbitrário.

A energia de um i -ésimo átomo $a_i \in \mathbb{U}$ é dada pela (Eq. 3.2). Portanto a energia total (E_t) do sistema será o somatório de todas as energias $a_i \in \mathbb{U}$:

$$E_t = \sum_{i=1}^l \mathcal{E}_i, \quad \forall i \in \mathbb{N}$$

onde

$$i \leq l. \quad (3.5)$$

com l sendo o número total de células.

De forma análoga podemos definir a energia de uma l -ésima célula $\mathcal{C}_l \in \mathcal{P}(\mathbb{U})$ como o somatório das energias de todos os átomos $a_i \in \mathcal{C}_l$.

Seja G uma função definida por $G : \mathcal{P}(\mathbb{U}) \rightarrow \mathbb{R}$:

$$G(\mathcal{C}_l) = \sum_{i=(h(l-1))+1}^{hl} \mathcal{E}_i, \quad (3.6)$$

A energia total do sistema poderá ser calculada somando as energias dos átomos presentes em cada célula:

$$E_t = G(\mathcal{C}_1) + G(\mathcal{C}_2) + \dots + G(\mathcal{C}_l), \quad (3.7)$$

A partir disso podemos afirmar que:

$$E_t = \sum_{i=1}^l \mathcal{E}_i = G(\mathcal{C}_1) + G(\mathcal{C}_2) + \dots + G(\mathcal{C}_l). \quad (3.8)$$

Uma vez que as energias das células somadas retornam a energia total do sistema, concluímos que essas energias podem ser calculadas computacionalmente de forma paralela. Esta observação abre a possibilidade para que a computação da energia total do sistema seja dividida utilizando uma abordagem análoga ao modelo de computação mestre-escravo (WILKINSON; ALLEN, 1999). Um fluxo de execução principal poderia criar novos fluxos de execução (*threads*) que ficariam responsáveis por calcular as energias de determinadas células do espaço de simulação.

Nesta abordagem cada processador disponível na máquina é responsável por calcular a energia de determinadas células do espaço. Contudo, é de interesse que no final da divisão de tarefas entre as *threads* o número final de átomos por processador seja o mais uniforme possível, para que as *threads* estejam balanceadas.

Para garantir uma divisão coerente, será necessário o uso de algoritmos de particionamento espacial. Este será usado visando assegurar um bom mapeamento do volume de simulação, e posteriormente uma divisão de tarefas homogênea.

3.3 Particionamento Espacial

Para realizar operações geométricas eficientes, ou de visualização sobre regiões espaciais com um grande volume de dados, é preciso buscar formas estruturadas de se organizar esses dados. Nessas situações é muito comum o uso de árvores de particionamento espacial, no qual os dados são dispostos hierarquicamente, permitindo com isso, um acesso muito mais eficiente aos dados no volume de simulação de interesse. Quando falamos em particionamento espacial, três principais estruturas devem ser citadas:

- *Octree*: o conceito de particionamento baseado em *Octrees* ou *Quadtree* (representação bidimensional) é simples: o volume de simulação é dividido recursivamente em octantes (ou quadrantes para duas dimensões) até que uma determinada condição – por exemplo, profundidade da árvore ou granularidade de vértices na suas folhas – seja alcançada (AYALA et al., 1985). Esta árvore de particionamento espacial é alinhada por eixos.
- *Kd-trees*: uma *kd-tree* é uma conhecida estrutura de dados usada para mapear hierarquicamente um conjunto de pontos em k -dimensões, onde as partições são realizadas usando planos alinhados aos eixos (ZHOU et al., 2008). O problema de construção da melhor *kd-tree* possível para uma dada estrutura, é um problema

combinatório e que por isso, não conveniente à todas as aplicações.

- Árvores BSP: Uma árvore BSP (*binary spatial partition* – partição espacial binária) oferece um grau de liberdade a mais no particionamento espacial se comparada com as estruturas anteriores. A criação da árvore também é feita usando planos, contudo não existe a restrição de particionamento seguindo a direção dos eixos principais, ou seja, os planos que de fato dividem o espaço podem adotar quaisquer direções (SAMET; WEBBER, 1988). Como na estrutura do item anterior, a criação desta pode ser cara, o que inviabiliza o seu uso em muitos casos.

Para o problema abordado nesse trabalho, será usado a árvore de partição *Octree*. Esta será escolhida por ser suficientemente capaz de realizar o mapeamento hierárquico de forma eficiente sem demandar um alto custo computacional em sua construção, dado os já conhecidos outros aspectos computacionais do problema em questão que também demandam uma intensiva computação de dados.

3.3.1 Octree

A forma no qual os dados estão dispostos no espaço que se deseja representar, influencia diretamente na escolha da estrutura de particionamento que se deseja utilizar. Portanto a distribuição dos dados – uniformes ou não, esparsos, disjuntos – devem ser levados em conta nessa escolha (LATTARI, 2007).

Uma das estruturas mais utilizadas quando os objetos estão distribuídos uniformemente ao longo do espaço são as *Octrees*. Seu esquema de representação é regular, baseado no alinhamento de eixos.

O processo de refinamento espacial usando *Octree* é basicamente constituído dos seguintes passos: a) amostragem da estrutura a ser espacialmente mapeada (Fig. 5a), b) representação do objeto escolhido no espaço de simulação (Fig. 5b) e c) refinamento do objeto no espaço em questão usando *Octree* (Fig. 5c).

A *Octree* representa um objeto em três dimensões como sendo uma união disjunta de células de tamanhos variáveis. O domínio de interesse é dividido recursivamente em octantes, que no escopo deste trabalho, são preenchidos (ou não) por átomos. Sendo assim, cada octante (célula) será representado como um nó da árvore e um determinado conjunto de átomos como folhas. É importante notar que cada nó da estrutura contém exatamente oito filhos. Contudo, alguns desses filhos podem ser considerados nulos quando não preenchidos por pelo menos um átomo (critério de simplificação). Uma

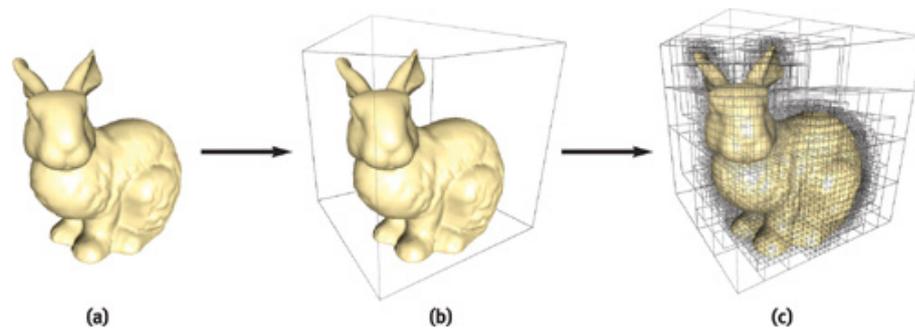


Figura 5: Etapas para realização do mapeamento espacial usando *Octree*.

definição formal, juntamente com vários algoritmos que implementam esse conceito pode ser encontrado em (YAMAGUCHI et al., 1984).

O processo de construção da árvore, no contexto deste trabalho, constitui uma das etapas mais importantes do processo. A representação da árvore para uma estrutura simplificada é apresentada na Figura 6. A criação da árvore funciona da seguinte maneira: a partir da raiz (que representa todo o domínio), o espaço é particionado em octantes e é verificado se o critério de parada foi alcançado, em caso afirmativo, o algoritmo cessa retornando a representação encontrada, caso contrário, cada filho da raiz é explorado e a partir deste, novos filhos são criados até que a representação esteja concluída.

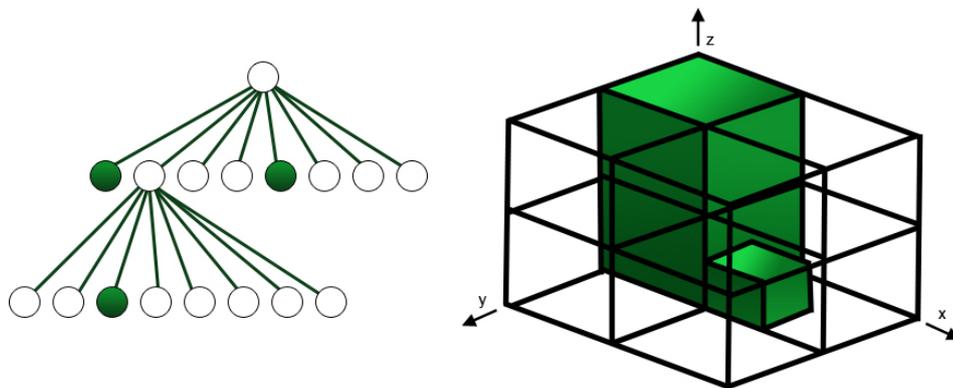


Figura 6: Representação de uma estrutura simples utilizando *Octree*.

O critério que verifica se um nó da árvore é folha, está associado com a granularidade do nó (conta quantos vértices, ou pontos estão presentes naquele nó), ou profundidade da árvore. O processo de construção recursivo termina quando todo o espaço for mapeado seguindo um critério de parada conveniente ao problema. Usando essa abordagem, os nós que não tiverem filhos, e não forem folhas, neste caso que não representem nenhum dado, são podados com a finalidade de otimizar uma futura busca na árvore.

Note na Figura 6 que as partições são alinhadas pelos eixos cartesianos x, y, z , o que

caracteriza a *Octree* como uma árvore de particionamento regular, ao contrário da árvore BSP.

Para exemplificar o critério de parada no particionamento da árvore, considere a Figura 7. Nesta figura é apresentado um espaço de simulação contendo onze átomos, dispostos bidimensionalmente, portanto, a árvore de representação será a *Quadtree*. É necessário mapear todo esse espaço com o objetivo de otimizar buscas por átomos específicos quando for necessário. O critério de parada na construção da árvore é **granularidade um**, o que significa que, se um determinado nó conter no máximo um átomo, este será uma folha, e caso contrário, um novo particionamento será realizada, até que esta condição seja aceita.

Nesta árvore, o nó *R* rotula o ponto de partida da árvore, ou seja, sua raiz. Os nós brancos marcados com a letra *N* indicam os nós intermediários que contêm filhos não nulos, e os marcados com os números variando de 1 a 13 representam as folhas da árvore. Note que os nós 2 e 8 (quadrados) são marcados de preto, pois estes não contêm nenhuma informação relevante para a aplicação (região do espaço vazia).

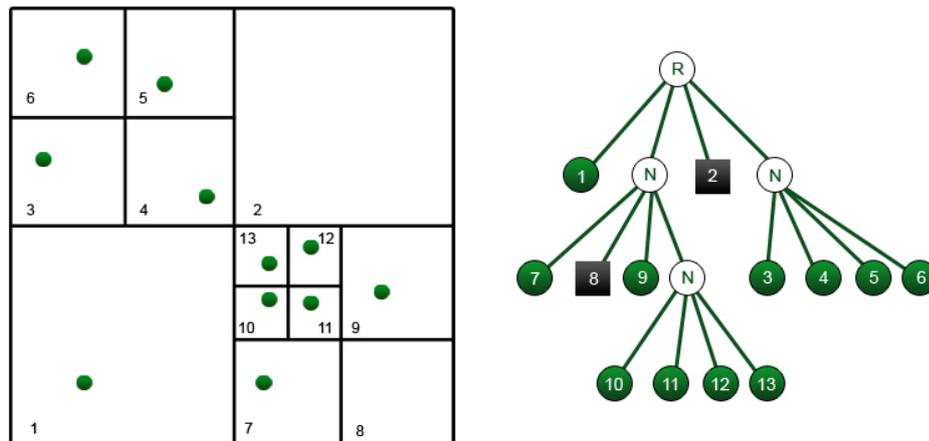


Figura 7: Mapeamento espacial de átomos dispersos no plano. As folhas da árvore são compostas de no máximo um átomo.

Um pseudocódigo para criação de uma *Octree*, tendo como critério de parada a granularidade nas folhas, é representada pela função **ConstroiOctree**.

No contexto dessa monografia, a estrutura de dados *Octree* foi escolhida para mapear o espaço de simulação pelos seguintes fatores:

- a *Octree* gera uma partição espacial hierárquica bem definida;
- provê uma estrutura compacta para um conjunto volumoso de dados;

Function ConstroiOctree(*CELULA* *filho, *INT* profundidade)

```

1 pai = *filho;
2 numAtomos = verificaGranularidade(pai )
3 if numAtomos ≤ MAX_GRANULARIDADE then
4   DefineFilhos(pai );
5   for i = 1 to 8 do
6     if pai → filho [i] ≠ NULL then
7       ConstroiOctree(pai → filho [i] , profundidade + 1);
8     else
9       return 0 ;
10    end
11  end
12 end

```

- o organizado mapeamento espacial dessa estrutura pode fornecer rápidas informações de vizinhanças no objeto amostrado;
- se ajusta à forma do objeto garantindo um refinamento eficiente;
- algoritmos de manipulação geométrica podem ser implementados de forma fácil, em cima dessa estrutura;
- o conceito da *Octree* é fácil de implementar.

Evitar o acesso à regiões do espaço não preenchidos por átomos é um dos maiores problemas encontrados quando a otimização do cálculo da interação de *spins* em estruturas magnéticas é necessária. Com o mapeamento espacial, esse problema é resolvido sendo necessário, a partir de então um esquema para divisão de tarefas entre múltiplos fluxos de execução. O particionamento espacial com *Octree* será uma das abordagens utilizadas para realizar essa divisão.

3.4 Divisão de Tarefas Entre *Threads*

O modelo físico proposto, e que foi discutido anteriormente, pode ser constituído de um número muito grande de átomos. Contudo, computacionalmente, o problema deve ser tratado de maneira discreta. Portanto, quanto maior o número de átomos, mais realistas serão os resultados obtidos. A principal desvantagem é que o tempo para se obter os resultados é proporcional ao tamanho do sistema. Nesta seção serão apresentadas formas diferentes de se paralelizar o modelo físico proposto. O principal objetivo é permitir que

a resolução computacional da Equação 3.2 seja obtida em menor tempo e, consequentemente, que sistemas maiores possam ser simulados em intervalos de tempo que possam ser obtidos com computadores atuais.

A divisão de tarefas entre *threads*, ou seja, a divisão dos átomos entre os fluxos de execução, é uma parte importante do processo de paralelização do problema. Sabemos que quanto mais homogênea for essa divisão, maior será o desempenho do algoritmo.

Em especial podemos destacar 3 diferentes formas de se subdividir o espaço visando a paralelização dos cálculos. A Figura 8 ilustra graficamente cada uma dessas abordagens. A primeira representação indica o particionamento espacial dinâmico baseado em planos: os planos do espaço são definidos em tempo de execução, e então alocados para cada *thread*. A segunda indica um particionamento cego baseado em regiões do espaço. Podemos chamar de particionamento cego, pois, nessa abordagem o espaço é subdividido em regiões estáticas e não levam em consideração os átomos presentes na estrutura. Esta divisão apresenta uma grande deficiência quando uma dessas partições contiver uma concentração de átomos muito superior do que as demais. E a terceira representação indica o particionamento espacial hierárquico baseado em *Octree*, como discutido na seção anterior..

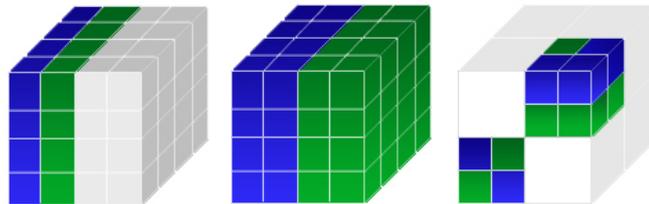


Figura 8: Representações das possíveis formas de se particionar o espaço: primeira representação baseada em planos, segunda baseada em particionamento uniforme do espaço, e terceira, divisão hierárquica usando *Octree*.

Em especial será apresentado formas de paralelização baseado apenas na primeira e na terceira representação da Figura 8.

3.4.1 Divisão de Tarefas Usando Planos

As especificações dessa subseção foram retiradas de (PEÇANHA et al., 2009). Para dividir o trabalho entre os fluxos, será apresentado nessa seção um esquema de mapeamento dinâmico do espaço. A ideia é escolher planos alinhados com eixos, onde cada fluxo fica responsável por calcular as energias dos átomos localizados nesses planos. A energia do

i -ésimo átomo é dada pela Equação 3.2 e o somatório de todas as energias de interação de todos os átomos nos dá a energia total do sistema.

Resumidamente, nossa implementação segue os seguintes passos:

- Para P fluxos disparados, cada um fica responsável pelo cálculo da energia de um dos diferentes planos do espaço;
- Ao término da computação da energia de um determinado plano, o P -ésimo fluxo soma o seu resultado a uma variável global;
- Se ainda existirem planos a serem computados, um deles é direcionado a este P -ésimo fluxo, conforme ilustrado na Figura 9;
- Ao término do cálculo da energia de todos os planos, a variável global com o somatório dos valores da energia de cada plano é retornada;
- Esses passos são seguidos a cada iteração.

Algumas restrições devem ser observadas para que a solução paralela funcione corretamente: a) fluxos diferentes devem computar energias de planos diferentes, b) a divisão dos planos deve seguir a uma das direções adotadas pelos eixos XY, YZ ou ZX e c) um plano cuja energia já tenha sido computada não deve ser acessado novamente.

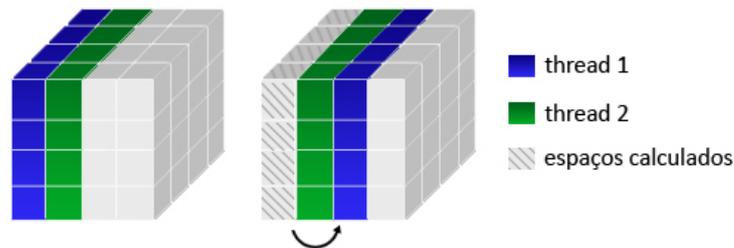


Figura 9: Na primeira figura, cada fluxo é responsável pelo cálculo da energia de um plano. Na segunda figura, o fluxo 1 (*thread 1*) termina o cálculo da energia do primeiro plano, busca um novo plano e marca o espaço já calculado como visitado.

O número de fluxos a ser utilizado na execução da aplicação pode variar. Em aplicações do tipo *CPU bound*, ou seja, que fazem uso intenso da CPU, costuma-se utilizar um número de fluxos igual ao número de processadores ou núcleos disponíveis na máquina. Deve-se ressaltar que um número de fluxos maior do que o número de processadores ou

núcleos existentes na máquina pode ser escolhido, contudo, poderá haver perda de desempenho, já que haverá disputa entre os fluxos pelo uso dos processadores. Em um cenário ideal, o ganho de desempenho será proporcional ao número de processadores disponíveis, levando-nos a obter a chamada aceleração linear (*linear speedup*). Assim, no momento de se determinar a quantidade de fluxos que serão utilizados na execução, deve-se observar a quantidade de recursos disponíveis na máquina, e não o número de planos na qual o espaço de simulação foi dividido.

Poder-se-ia ainda adotar uma abordagem onde o espaço de simulação é previamente particionado, como mostrado na segunda representação da Figura 8. Contudo, essa abordagem foi descartada, visto que o objeto a ser simulado pode adotar geometrias variadas e, caso uma destas seja assimétrica e as partições realizadas no espaço sejam regulares, células diferentes podem conter números de átomos diferentes. A discrepância entre o número de átomos por partição pode provocar desbalanceamento entre os fluxos, limitando o desempenho obtido. Isso decorre do fato de fluxos diferentes realizarem cálculos a mais (ou a menos) que outras, portanto, fluxos cujas tarefas já tenham sido realizadas permanecerão em espera até que todos os outros fluxos tenham finalizado suas tarefas.

A grande vantagem do modelo dinâmico implementado é que ele garante que um fluxo não ficará ocioso, a não ser que as energias de todos os planos já tenham sido calculadas. Uma das desvantagens observadas é o grande número de acessos a regiões de seção crítica.

Para implementar o modelo apresentado, usamos o conceito de *pool de threads*. Neste os fluxos são criados apenas uma vez a cada execução do programa. No início do algoritmo é indicado o número máximo de fluxos que o processo principal poderá disparar. Este número determinará o número de fluxos N no *pool*. Os fluxos ficam adormecidos até que alguma tarefa seja requisitada. Ao término dessa tarefa, o fluxo retorna ao *pool* e fica em estado de espera até que a solicitação de uma nova tarefa seja feita. O conceito de *pool* se contrapõe a implementação convencional, onde o fluxo é criado sob demanda e, ao término de seu trabalho, este é sincronizado e posteriormente destruído. O uso do *pool de threads* evita que fluxos sejam criados a todo momento, tornando o processo computacionalmente menos custoso.

3.5 Divisão de Tarefas Usando *Octree*

O método de divisão de tarefas entre *threads* apresentado na seção anterior se mostra eficiente para estruturas geométricas que ocupam uma grande parcela do espaço, assim

como esferas e cubos. Contudo, para geometrias vazadas, como tubos e cascas esféricas, o método apresenta uma falha: o algoritmo, tanto sequencial quanto paralelo usando divisão por planos, percorre todo o espaço matricial em busca de elementos que pertençam à geometria do objeto.

Neste trabalho, os dados estão representados na forma de uma matriz tridimensional tal que cada célula dessa matriz pode ter (ou não) um átomo localizado em seu interior (Fig. 10). Existe, portanto, apenas um átomo por elemento de volume. Quando desejamos modelar uma estrutura mais elaborada, a geometria do objeto é definida de forma implícita (VELHO; GOMES; FIGUEIREDO, 2002). Portanto, na construção da matriz que representará o espaço juntamente com o objeto, será atribuída valor 1 aos *spins* pertencentes àquela geometria e 0 aos não pertencentes. Note que para geometrias vazadas, teremos uma matriz esparsa, onde a maioria dos dados que a constituem são informações não significativas.

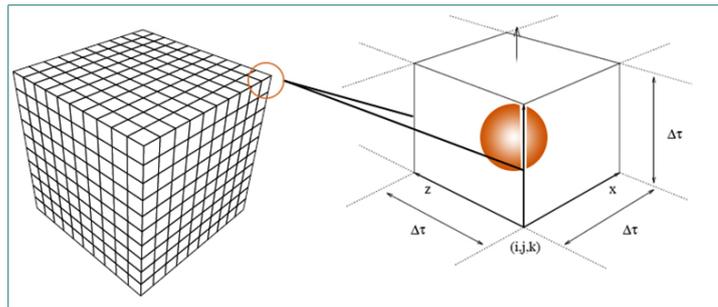


Figura 10: Representação dos átomos computacionalmente.

Ambos os algoritmos, sequencial e de divisão por planos, percorrem toda matriz de simulação em busca de elementos que pertençam à geometria que está sendo simulada. Algoritmos de particionamento espacial hierárquico servem justamente para refinar esse espaço de forma a minimizar o tempo gasto para acessar elementos significativos (regiões do espaço que contenham átomos). Veremos na seção de resultados que algoritmos que usam esse conceito conseguem obter resultados numéricos de forma muito mais rápida.

Com o espaço completamente disposto de forma hierárquica, usando a representação baseada em *Octree*, nos resta agora dividir as partições espaciais entre as *threads*. A ideia principal gira em torno da montagem de listas contendo células dos espaços onde cada um dos fluxos de execução será responsável pela computação de uma dessas listas.

Considere o espaço representado pela Figura 11 com sua respectiva árvore ilustrada na Figura 12. Será analisada a divisão de tarefas para duas *threads*. Na Figura 12 as

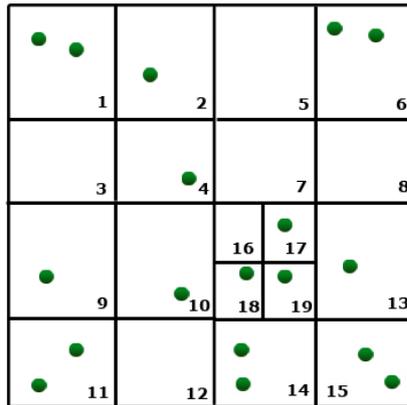


Figura 11: Região espacial particionada usando *Quad-tree*.

células alocadas para cada um das *threads* estão representadas pelas cores verde (*thread* 1) e azul (*thread* 2). Após o particionamento espacial, essa divisão de tarefa entre as *threads* pode seguir o seguinte esquema:

- A alocação de células para cada processador consiste na montagem de filas de processamento, onde cada *thread* ficará restrita à computação das energias de átomos presentes nessas células específicas;
- Dada uma folha, verifique se ela é válida. Em caso afirmativo, coloque-a na fila de células a serem computadas pelo processador 1, caso contrário, descarte-a e procure pela próxima;
- Após a alocação de uma célula para o fluxo de execução 1, procure por uma folha válida (que contenha átomos) e a coloque na fila da *thread* 2.
- Enfileire uma folha de cada vez para cada *thread*, até que toda árvore tenha sido percorrida, e todas as células do espaço divididas entre os fluxos.

Nesse esquema de divisão, a *thread* 1 ficará responsável pelo cálculo das energias dos átomos presentes nas células 1,4,9,11,14,17 e 19, enquanto a *thread* 2 pela computação das células 2,6,10,13,15 e 18.

Esse esquema de divisão de tarefas se mostra eficiente, sobretudo para regiões espaciais esparsas. Uma característica interessante do método apresentado é que ele garante a melhor divisão possível para geometrias simétricas com um número de *threads* divisível por 8.

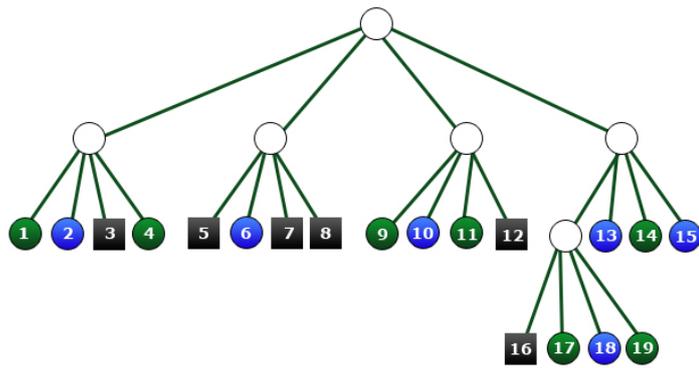


Figura 12: Representação da árvore de particionamento do espaço mostrado na figura anterior.

Assim como no método de divisão por planos apresentado anteriormente, a divisão usando *Octree* também foi implementada usando o conceito de *pool* de *threads*. Na seção de resultados será discutida a eficiência desse esquema, se comparado à implementação tradicional.

4 Experimentos

Para testar a eficiência dos métodos de paralelização e particionamento espacial, foi desenvolvida uma aplicação na linguagem C/C++. Essa aplicação é composta basicamente por um núcleo numérico, responsável por realizar os cálculos referentes às interações atômicas, e um visualizador implementado usando a biblioteca gráfica OpenGL.

Os módulos de computação paralela desenvolvidos, e que implementam os métodos de divisão de tarefas por planos e *Octree* propostos foram codificados usando a biblioteca *pthread* (BUTENHOF, 1997). Os resultados dessas implementações serão discutidas neste capítulo.

A interface de visualização tem como objetivo expor graficamente os estados dos átomos ao longo da região de simulação sobre determinadas temperaturas e campos magnéticos (Fig. 13). Com esse visualizador é possível analisar o comportamento dos átomos e comprovar se a situação de fato condiz com a realidade. Para tal, cada átomo foi representado como uma esfera, onde em seu interior foram posicionadas setas que representam as orientações dos *spins*. As cores dos átomos indicam suas energias individuais, onde cores próximas do azul assinalam menor energia, e vermelho maior.

O conceito de particionamento espacial adaptativo usando *Octree*, foi implementado usando os dois critérios de parada apresentados, profundidade e granularidade. A Figura 14 mostra uma casca esférica mapeada por uma *Octree* baseada em profundidade. Note que as regiões esparsas do volume de simulação são consideravelmente reduzidas a medida que a profundidade aumenta.

Teoricamente, temos que quanto menor o número de *spins* por folha, ou seja, um cenário próximo da árvore com profundidade 5 na Figura 14, mais homogêneo será a divisão de tarefas entre as *threads*. Contudo, é importante frisar que uma árvore com poucos *spins* por folha não é a melhor escolha devido aos seguintes fatores: o particionamento que gera células com quantidades de *spins* próximos à um, possui uma árvore resultante grande, o que aumenta consideravelmente a complexidade de acesso a essa estrutura; como a árvore

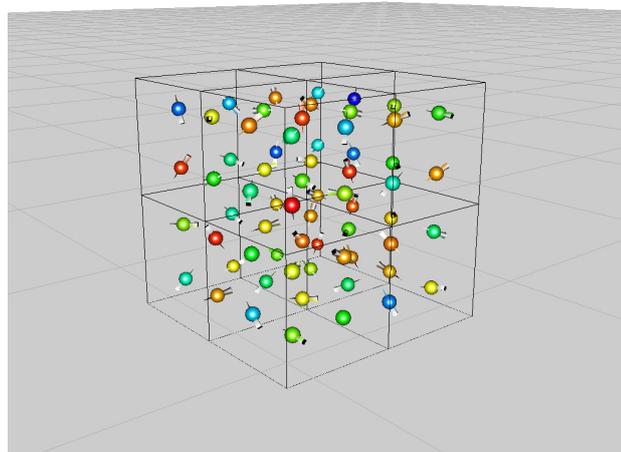


Figura 13: Região espacial simples composta de 64 átomos dispostos regularmente ao longo de uma matriz tridimensional.

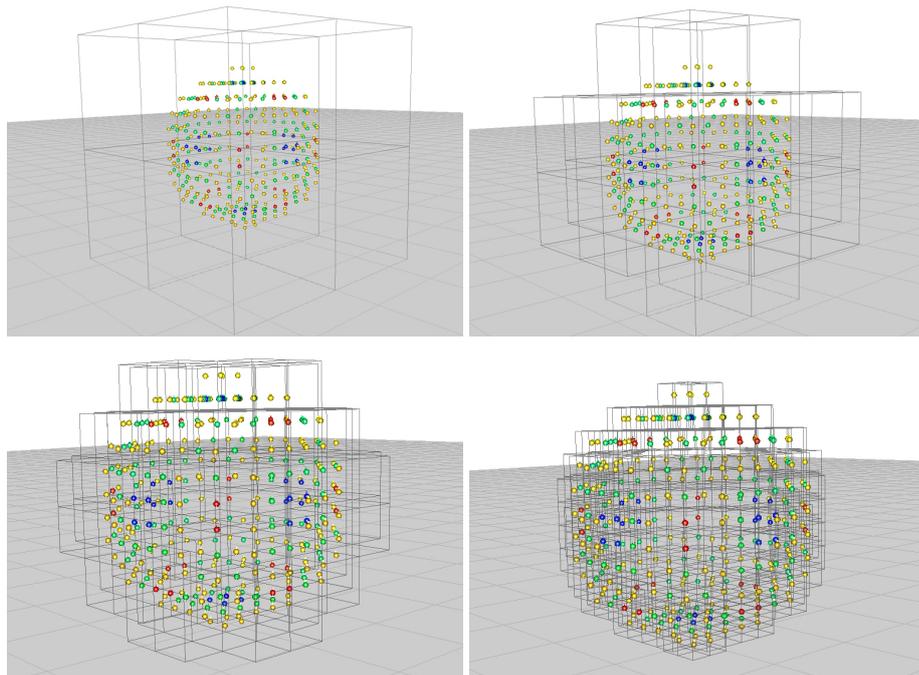


Figura 14: Refinamento espacial usando *Octree*: progressão do particionamento para profundidades 2,3,4 e 5 respectivamente

tem valência fixa oito, os nós esparsos (que não contem átomos), mesmo não sendo considerados na visualização e simulação, ocupam espaços desnecessários na memória; muitas vezes é de interesse simular estruturas com uma grande quantidade de átomos, logo a memória disponível no ambiente computacional pode ser insuficiente para a geração de determinadas *Octrees*; assumindo que *spins* presentes em uma mesma célula estão localizados próximos uns dos outros na memória do computador, a criação de árvores com

muitos *spins* por folha evita saltos de memória, quando as *threads* estiverem percorrendo as listas formadas pelo algoritmo de divisão de tarefas.

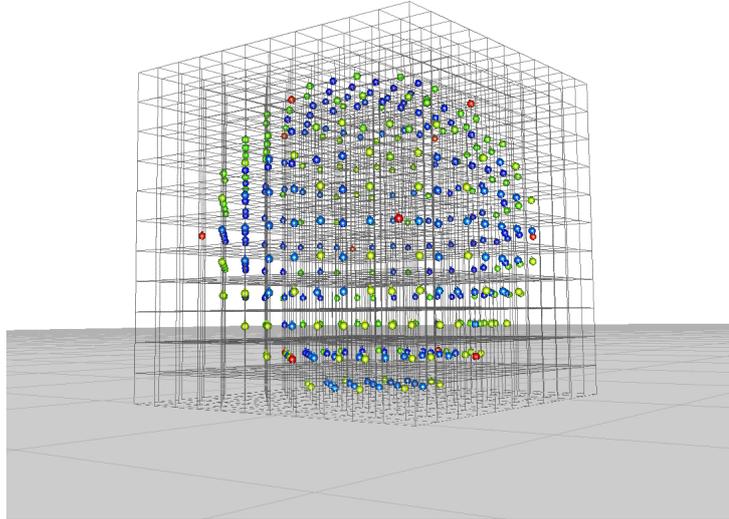


Figura 15: Representação convencional de uma casca esférica usando a representação matricial sem particionamento espacial.

Portanto, existe a necessidade de se encontrar um número de *spins* por folha que garanta uma divisão homogênea, e que leve em consideração os fatores citados.

A Figura 15 mostra a representação convencional de uma estrutura com geometria arbitrária, no caso, uma esfera, usando uma matriz tridimensional. Note na imagem que existem muitos elementos vazios na matriz, e que ainda assim serão acessados pelos algoritmos convencionais. Como mencionado, a ideia fundamental do algoritmo de particionamento espacial é evitar ao máximo acessos à regiões não significativas do espaço.

4.1 Testes de Desempenho

Nesta seção são apresentados os resultados obtidos nas simulações do modelo computacional descrito neste trabalho. As simulações foram executadas em uma máquina Dual Xeon E5410 Quad Core de 2.33Ghz com 4 GB de memória RAM, 12 MB de memória cache L2, placa-mãe Intel Workstation Board modelo S5000XVN e sistema operacional Linux Ubuntu 9.04 64 bits. Portanto 8 núcleos de processamento estão disponíveis nesta máquina. Nos testes realizados o critério de parada escolhido foi 5000 passos de Monte Carlo. Para assegurar uma maior confiabilidade nos dados obtidos, os testes dos três

algoritmos, sequencial, divisão por planos e *Octree*, foram executados 10 vezes para cada configuração (número de *threads*, número de átomos e geometria do objeto), e assim a média desses valores é obtida.

O primeiro teste realizado teve como objetivo verificar a eficiência do esquema de particionamento espacial por planos para regiões completamente preenchidas por átomos. A Figura 16 mostra a aceleração obtida em função do número de *threads*. O espaço de simulação usado foi um cubo de lado 16, totalmente preenchido. Observando a figura, podemos notar que o ganho relativo é o máximo (igual ao número de *threads* disparadas) para um número de *threads* que divida perfeitamente o espaço de simulação. Como explicação tomemos a situação onde são usadas 7 *threads*. Nesse caso, teremos sempre 7 *threads* realizando cálculos de forma concorrente, contudo, como esse número não divide o espaço perfeitamente, e como existe a restrição de apenas uma *thread* por plano, sobrarão dois planos que serão computados por apenas dois fluxos de execução. Com isso, os outros 5 fluxos ficarão em espera para realizar a sincronização, logo, a arquitetura paralela de *hardware* disponível não é aproveitada em seu máximo nessa situação.

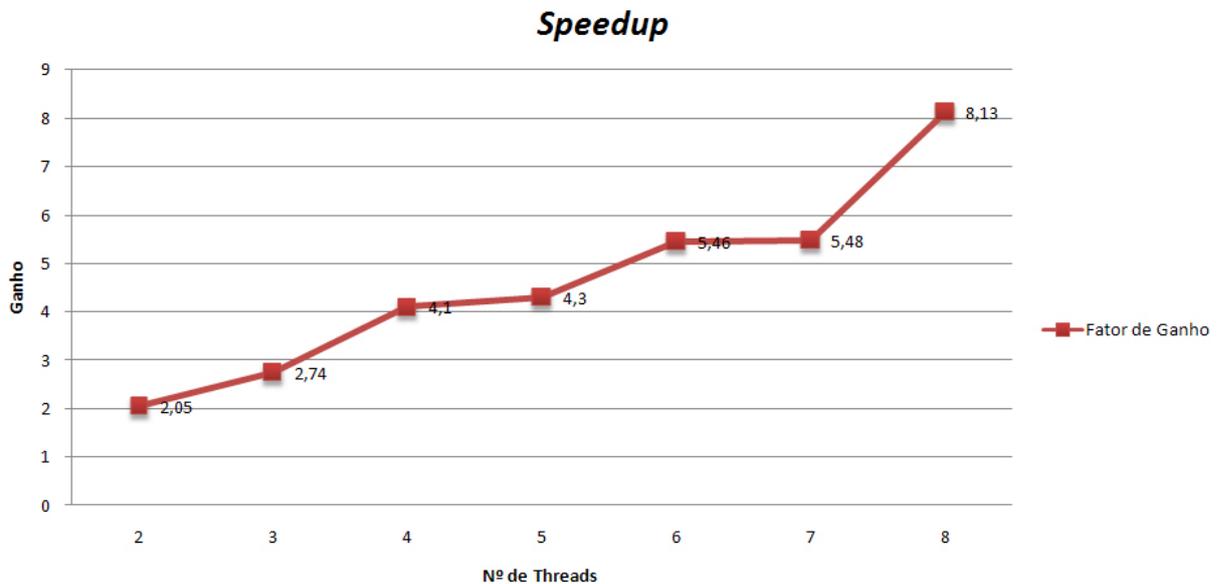


Figura 16: *Speedup* para uma região preenchida com 4096 átomos.

Foram realizados testes para análise de comportamento do algoritmo de divisão por planos para cubos preenchidos com dimensões diferentes. A Figura 17 ilustra os resultados obtidos. Note que o tempo de execução do algoritmo paralelo foi muito menor, o que comprova sua eficiência para regiões espaciais completamente preenchidas por átomos.

Contudo, sabemos que o ganho desse algoritmo pode ser prejudicado pela alta concentração de átomos em determinados planos. Isso ocorre, por exemplo, na estrutura da

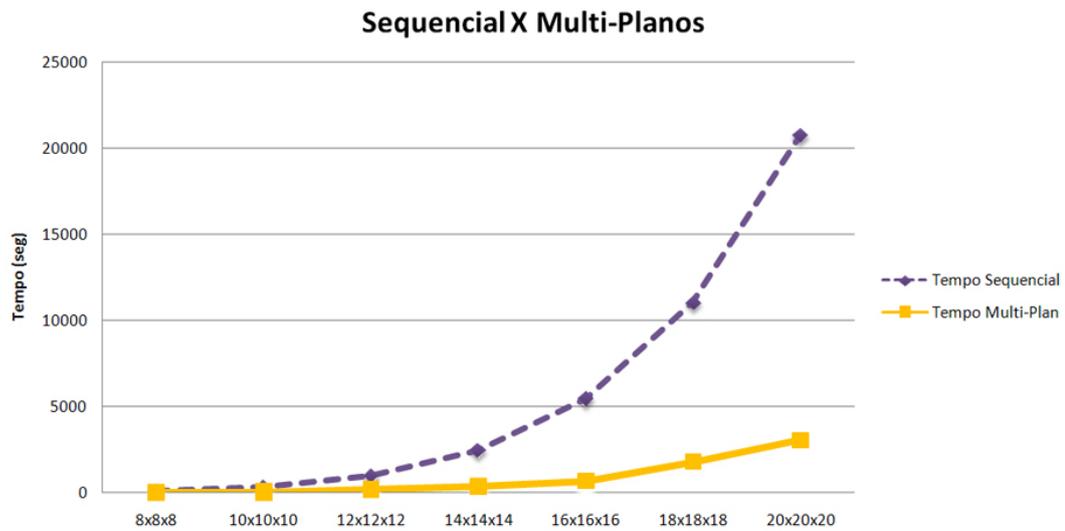


Figura 17: Comparação do tempo de execução em função da dimensão do espaço. Nessa situação foram usadas 8 *threads*.

Figura 15. Nesta situação, planos mais internos à esfera são compostos por mais átomos do que nas bordas do objeto. Essa deficiência pode ser observada no gráfico da Figura 18. O gráfico mostra experimentos realizados usando os algoritmos de particionamento por planos e *Octree* sobre uma casca esférica de raio 10. A linha pontilhada, representa o *speedup* relativo obtido pelo algoritmo de divisão por planos. Note que, embora o resultado possa ser considerado satisfatório, o ganho para 8 *threads* para essa geometria não foi tão bom quanto no teste anterior para o cubo.

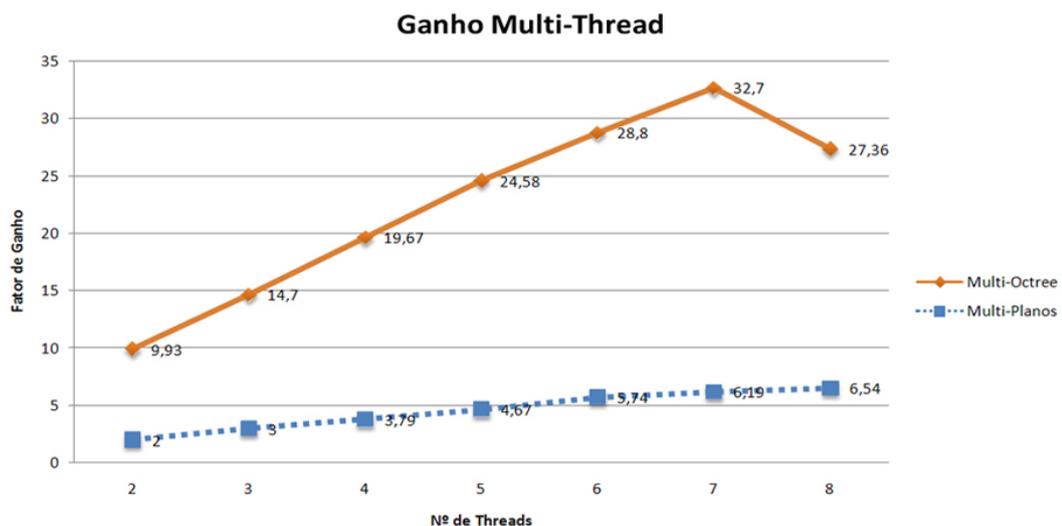


Figura 18: Fator de ganho obtido usando simulações baseadas em planos e *Octree* em função do número de *threads*.

O gráfico da Figura 18 também mostra o desempenho do algoritmo de particionamento

espacial por *Octree*. Neste experimento foi usada uma árvore com profundidade limite igual a 4. Os resultados obtidos comprovam todas as vantagens que caracterizam esse método de divisão, e que foram discutidos ao longo desse trabalho. Um deles é a divisão de átomos praticamente homogênea entre os processadores que foi conseguida. Isso é evidenciado pelo comportamento linear nos fatores de ganho do algoritmo entre 2 e 7 *threads*. Os motivos que levaram a um decaimento no desempenho do algoritmo para 8 *threads* ainda está sendo estudado.

Melhores resultados podem ser obtidos quando temos uma geometria ainda mais esparsa. Como demonstração, foram executados experimentos usando um objeto com muitas regiões vazias no espaço: um tubo de parede única, com altura 40 e raio 8. Na Figura 19 são mostrados os resultados. O desempenho do algoritmo baseado em *Octree* superou todas as expectativas alcançando um fator de ganho de até 44 vezes, se comparado com o algoritmo sequencial. Então, podemos dizer que o desempenho do algoritmo está diretamente relacionado com a geometria do objeto simulado e o número de processadores/núcleos disponíveis no ambiente computacional.

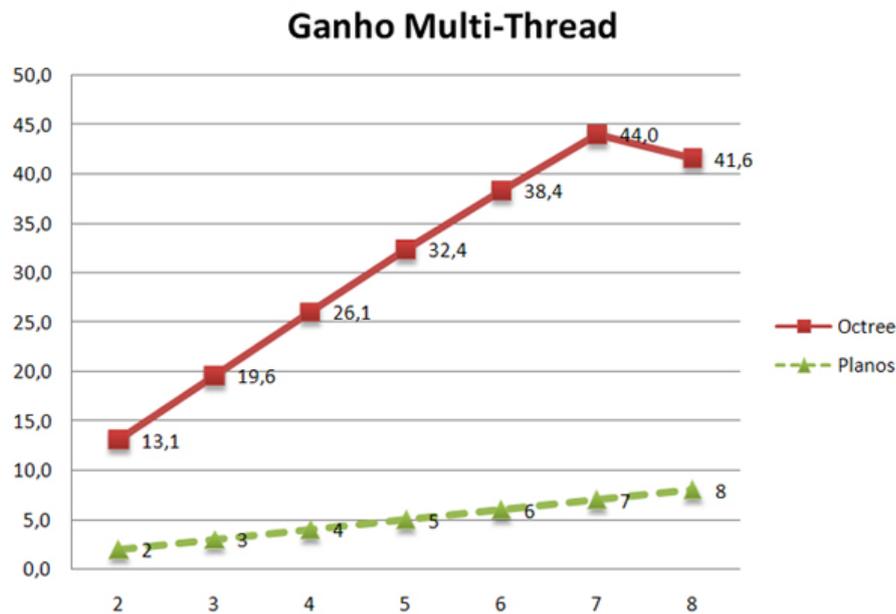


Figura 19: Gráfico do fator de ganho dos algoritmos propostos executados sobre um cilindro vazado de parede única.

Devemos frisar que o desempenho do algoritmo de divisão por planos, no último experimento realizado (Fig. 19), alcançou os melhores resultados possíveis dada a sua implementação, obtendo o que chamamos de *speedup* linear.

5 Conclusão

Neste trabalho foi apresentado um modelo físico e computacional para simulação da interação interatômica em nanoestruturas magnéticas usando particionamento espacial. Foi visto que processo de paralelização se fez necessário pela observação de que a simulação dos modelos envolve um alto custo computacional, diretamente associado à quantidade de elementos (*spins*) presentes no sistema, o que, em última instância, limita o tamanho do sistema simulado.

Os modelos de paralelização apresentados basearam-se no conceito de particionamento espacial dinâmico (usando planos) e estático (*Octree*). Juntamente com o processo de particionamento foram apresentados métodos para divisão de tarefas entre *threads*.

O método de paralelização baseado em planos se apresentou muito robusto e eficiente para espaços com uma alta densidade de átomos. Usando essa abordagem foi possível alcançar bons resultados, onde foi obtido um *speedup* de até 8 vezes se comparado com a versão sequencial.

O outro método de paralelização proposto, baseado em particionamento usando *Octree*, alcançou excelentes resultados chegando a realizar o processo de simulação até 44 vezes mais rápido do que a versão sequencial. O alto desempenho desse algoritmo está diretamente relacionado com a geometria do objeto simulado e com o número de processadores/núcleos disponíveis.

Duas questões que ainda estão em aberto é o motivo do decaimento no desempenho do algoritmo baseado *Octree* para 8 *threads* (Fig. 18 e Fig. 19) e a profundidade, ou granularidade, ideal para a montagem da *Octree*.

Referências

- AYALA, D. et al. Object representation by means of nonminimal division quadtrees and octrees. *ACM Trans. Graph.*, ACM, New York, NY, USA, v. 4, n. 1, p. 41–59, 1985. ISSN 0730-0301.
- BUTENHOF, D. R. *Programming with POSIX Threads*. 1. ed. MA, USA: Addison-Wesley Reading, 1997.
- CAO, L. et al. Size and shape effects on curie temperature of ferromagnetic nanoparticles. *Transactions of Nonferrous Metals Society of China*, v. 17, n. 6, p. 1451 – 1455, 2007. ISSN 1003-6326.
- JANKE, W. Nonlocal monte carlo algorithms for statistical physics applications. *Mathematics and Computers in Simulation*, v. 47, n. 2-5, p. 329 – 346, 1998. ISSN 0378-4754.
- KITTEL, C. *Introdução à Física do Estado Sólido*. 5. ed. Rio de Janeiro, RJ: Editora Guanabara Dois S. A., 1982.
- LATTARI, L. G. *Gerenciamento e Visualização de Ambientes Virtuais*. [S.l.]: Universidade Federal de Juiz de Fora, 2007.
- MARTINS, N. *Introdução à Teoria da Eletricidade e do Magnetismo*. 2. ed. São Paulo, BRA: Edgard Blücher, 1975.
- MATSUMOTO, M.; NISHIMURA, T. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, ACM, New York, NY, USA, v. 8, n. 1, p. 3–30, 1998. ISSN 1049-3301.
- METROPOLIS, N. et al. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, v. 21, p. 1087–1092, 1953.
- NETO, J. dos S. C. *Estudo Sistemático das Propriedades Termodinâmicas e criticalidade de Filmes Finos e Super-Redes Magnéticas*. [S.l.]: Universidade Federal de São Carlos, 2004.
- PEÇANHA, J. et al. Simulação computacional da interação de spins em sistemas magnéticos usando múltiplos fluxos de execução. *XII Encontro de Modelagem Computacional*, 2009.
- PEÇANHA, J. et al. Um modelo computacional para simulação de interação de spins em elementos e compostos magnéticos. *XI Encontro de Modelagem Computacional*, 2008.
- REITZ, J. R.; MILFORD, F. J.; CHRISTY, R. W. *Fundamentos da Teoria Eletromagnética*. 1. ed. Rio de Janeiro, BRA: CAMPUS, 1982.

- SABINO, T. et al. Simulação de dinâmica molecular usando o potencial de lennard-jones. *XII Encontro de Modelagem Computacional*, 2009.
- SAMET, H.; WEBBER, R. E. Hierarchical data structures and algorithms for computer graphics. *IEEE Comput. Graph. Appl.*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 8, n. 4, p. 59–65, 67–75, 1988. ISSN 0272-1716.
- TIPLER, P. A.; MOSCA, G. *Physics - For Scientists and Engineers*. 6. ed. New York, NY, USA: W. H. Freeman, 2007.
- VELHO, L.; GOMES, J.; FIGUEIREDO, L. H. de. *Implicit Objects in Computer Graphics*. 1. ed. New York, LLC: Springer-Verlag, 2002.
- WILKINSON, B.; ALLEN, M. *Parallel programming: techniques and applications using networked workstations and parallel computers*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1999. ISBN 0-13-671710-1.
- YAMAGUCHI, K. et al. Octree-related data structures and algorithms. *IEEE Comput. Graph. Appl.*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 4, n. 1, p. 53–59, 1984. ISSN 0272-1716.
- YEOMANS, J. M. *Statistical Mechanics of Phase Transitions*. [S.l.]: Oxford University Press, 1992.
- ZHOU, K. et al. Real-time kd-tree construction on graphics hardware. In: *SIGGRAPH Asia '08: ACM SIGGRAPH Asia 2008 papers*. New York, NY, USA: ACM, 2008. p. 1–11.