

Thiago de Oliveira Quinelato

**Simulação Computacional Multi-CPU de Dinâmica
Molecular de Carbono Utilizando o Potencial
REBO2**

Juiz de Fora

2010

Thiago de Oliveira Quinelato

**Simulação Computacional Multi-CPU de Dinâmica
Molecular de Carbono Utilizando o Potencial
REBO2**

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

2010

Monografia submetida ao corpo docente 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

Agradecimentos

Agradeço primeiramente aos meus pais e à Thaiane, minha irmã mais querida, pelo total apoio e dedicação em todos os passos desta caminhada.

Agradeço também ao Diego, à Mila, à D. Arlete e ao Sr. José Clínio, por me acolherem como se eu fizesse parte da família.

Aos meus colegas do Grupo de Computação Gráfica e principalmente à Patrícia e ao Pedro, por colaborarem no desenvolvimento do programa de simulação deste trabalho.

É impossível não ser grato aos orientadores desse projeto pelo tempo e paciência dedicados.

Finalmente, agradeço ao CNPq pelo auxílio financeiro e à FAPEMIG pelos equipamentos disponibilizados.

Sumário

Resumo

1	Introdução	p. 7
1.1	Definição do Problema	p. 7
1.2	Objetivos	p. 8
2	Dinâmica de estruturas de carbono	p. 9
2.1	Potencial REBO2	p. 9
2.1.1	A função de ordem de ligação b_{ij}	p. 10
2.2	Dinâmica molecular	p. 13
2.3	Gradiente do Potencial REBO2	p. 14
3	Modelo Computacional	p. 17
3.1	Sistema de Comunicação	p. 17
3.1.1	Módulo Servidor	p. 17
3.1.2	Módulo Cliente	p. 18
3.2	Protocolo de Controle da Simulação	p. 19
3.2.1	Tipos de Processo	p. 19
3.2.2	Inicialização do Ambiente de Simulação	p. 20
3.2.3	Aplicativo Gráfico de Acesso e Controle	p. 20
3.2.4	Protocolo de Comunicação	p. 21
3.3	Particionamento de Espaço e Balanceamento de Carga	p. 21
3.3.1	Classificação de partículas	p. 21

4 Resultados Experimentais	p.27
4.1 Estudo de Caso	p.28
5 Conclusão	p.30

Referências Bibliográficas

Resumo

Esta monografia trata da simulação computacional distribuída de dinâmica molecular em nanoestruturas de carbono. Apresenta os fundamentos físicos necessários à compreensão do problema. Mostra, ainda, o potencial REBO2, que é a função usada neste trabalho para a avaliação da energia. Um esquema de particionamento e classificação para átomos em simulações distribuídas é proposto. Foi desenvolvido um sistema completo de simulação, comunicação e visualização cujas interações são apresentadas. O sistema de computação distribuída foi completamente desenvolvido e, portanto, nenhuma biblioteca clássica foi utilizada. Por fim, discute-se o desempenho dessa ferramenta sob algumas configurações.

Palavras-chave: Simulação Computacional, Nanotubos de Carbono, Computação Distribuída, Dinâmica Molecular.

1 Introdução

O estudo das propriedades de nanomateriais apresenta um vasto campo para pesquisa e desenvolvimento de novos materiais e tecnologias, úteis para diversas aplicações, desde a confecção de tecidos super-resistentes até a construção de componentes eletrônicos de alta eficiência. O interesse em realizar simulações computacionais para análise desses materiais é decorrente do alto custo relacionado à sua síntese e experimentação física.

Recentemente, um certo grupo de nanomateriais, os nanotubos de carbono, tem despertado a atenção de pesquisadores das mais variadas áreas. Há considerável foco em estudar a relação de dependência das características desses materiais à sua geometria, visto que determinadas propriedades desejáveis podem ser observadas quando os nanotubos de carbono apresentam-se em formas estruturais específicas.

O uso de uma máquina eficiente de simulação computacional de dinâmica molecular, associada a um modelo computacional que descreva adequadamente essas nanoestruturas, mostra-se como possível alternativa à dispendiosa análise em laboratório das propriedades desses materiais. Ainda, a utilização de computação gráfica torna as simulações mais intuitivas, evidenciando possíveis falhas de modelagem ou simulação. Tem-se, dessa forma, uma poderosa ferramenta para experimentação virtual de nanotubos de carbono.

1.1 Definição do Problema

Estruturas moleculares que apresentam propriedades interessantes frequentemente são formadas por dezenas ou até mesmo centenas de milhares de partículas. Como o esforço computacional necessário à realização de simulações de dinâmica molecular é diretamente proporcional ao número de partículas em análise, o estudo de estruturas formadas por muitos átomos pode tornar-se inviável. Este trabalho apresenta uma forma de processamento paralelo de um modelo de simulação de dinâmica molecular baseado no potencial REBO2 para redução do tempo de solução numérica desse modelo.

1.2 Objetivos

O objetivo primário deste trabalho é o estudo e desenvolvimento de um método para dinâmica molecular baseado no potencial REBO2 que atue de forma paralela e distribuída, utilizando um esquema de comunicação próprio, a fim de reduzir o tempo necessário à solução numérica do modelo computacional.

Como objetivos secundários, serão definidas estratégias para atingir um balanceamento de carga e um particionamento espacial adequados, de modo que a troca de mensagens entre as unidades de processamento tenha baixa influência no tempo de simulação.

2 *Dinâmica de estruturas de carbono*

2.1 Potencial REBO2

A função de energia potencial REBO2, apresentada em (BRENNER et al., 2002), foi construída de maneira empírica e fornece uma boa aproximação para a energia potencial em estruturas moleculares sólidas formadas por átomos de carbono e hidrogênio.

O potencial definido é de curto alcance e considera a quebra e formação de ligações covalentes, com a conseqüente mudança de hibridização dos átomos envolvidos. Por depender apenas do primeiro e do segundo átomos vizinhos mais próximos de cada átomo, a computação do potencial REBO2 apresenta um reduzido número de operações, quando comparada a abordagens baseadas em cálculos de primeiros princípios e funções de energia potencial semi-empíricas. Por esse motivo, é um potencial que tem sido largamente utilizado.

A forma geral da energia potencial total do sistema é definida pela Equação 2.1, conforme (ABELL, 1985).

$$E_b = \sum_i \sum_{j(>i)} \left[V^R(r_{ij}) - b_{ij} V^A(r_{ij}) \right] \quad (2.1)$$

Os termos $V^A(r_{ij})$ (Eq. 2.2) e $V^R(r_{ij})$ (Eq. 2.3) representam, respectivamente, a atração entre os elétrons de valência e a repulsão decorrente da interação entre núcleos. O valor escalar r_{ij} é a distância entre dois átomos vizinhos i e j . A função b_{ij} , que potencializa a ação do componente atrativo $V^A(r_{ij})$, é explicada Seção 2.1.1.

$$V^A(r_{ij}) = f^c(r_{ij}) \sum_{1 \leq n \leq 3} B_n e^{-\beta_n r_{ij}} \quad (2.2)$$

$$V^R(r_{ij}) = f^c(r_{ij})(1 + Q/r_{ij})Ae^{-\alpha r_{ij}} \quad (2.3)$$

O fator multiplicativo $f^c(r_{ij})$ (Eq. 2.4) define o raio de corte para a ação do potencial, garantindo assim que só serão consideradas ligações entre átomos que estejam a uma distância de 2Å ou menos ($1\text{Å} = 10^{-10}m$). É interessante observar que essa função varia suavemente de 1,0 (para $r_{ij} = 1,7\text{Å}$) até 0,0 (para $r_{ij} = 2,0\text{Å}$).

$$f^c(r_{ij}) = \begin{cases} 1,0 & r_{ij} < 1,7 \\ \frac{1 + \cos\left(\frac{\pi(r_{ij} - 1,7)}{0,3}\right)}{2} & 1,7 \leq r_{ij} \leq 2,0 \\ 0,0 & r_{ij} > 2,0 \end{cases} \quad (2.4)$$

Os valores para as constantes A , α , B_1 , B_2 , B_3 , β_1 , β_2 , β_3 e Q podem ser obtidos em (BRENNER et al., 2002).

2.1.1 A função de ordem de ligação b_{ij}

A função b_{ij} (Eq. 2.5) está relacionada à ordem da ligação entre cada par de átomos i e j e agrega também a influência exercida por átomos próximos na energia da ligação.

$$b_{ij} = \frac{b_{ij}^{\sigma-\pi} + b_{ji}^{\sigma-\pi}}{2} + b_{ij}^{RC} + b_{ij}^{DH} \quad (2.5)$$

Os valores $b_{ij}^{\sigma-\pi}$ e $b_{ji}^{\sigma-\pi}$ (Eq. 2.6) dependem do número de coordenação local e do ângulo de ligação respectivamente entre os átomos i e j e outros átomos que não participam da ligação.

$$b_{ij}^{\sigma-\pi} = \left[1 + \sum_{k(\neq i,j)} f_{ik}^c(r_{ik}) G_c(\cos(\theta_{ijk})) e^{\lambda_{ijk}} + P_{ij}(N_i^C, N_i^H) \right]^{-1/2}, \quad (2.6)$$

onde θ_{ijk} é o ângulo entre os vetores \vec{r}_{ij} e \vec{r}_{ik} . A função $G_c(\cos(\theta_{ijk}))$ (2.7) representa a contribuição que cada átomo vizinho à ligação $i - j$ tem à ordem de ligação empírica.

$$G_c(\cos(\theta_{ijk})) = \begin{cases} g_c(\cos(\theta_{ijk})) + Q(N_i^H) [\gamma_c - g_c(\cos(\theta_{ijk}))] & 0^\circ < \theta_{ijk} < 109,47^\circ \\ g_c(\cos(\theta_{ijk})) & 109,47^\circ < \theta_{ijk} < 180^\circ \end{cases} \quad (2.7)$$

A função $g_c(\cos(\theta_{ijk}))$ é aproximada pela *spline* polinomial definida pela Equação 2.8. Os valores de a_n variam conforme o ângulo θ_{ijk} e são exibidos na Tabela 2.1.

Tabela 2.1: Parâmetros utilizados para a *spline* $g_c(\cos(\theta_{ijk}))$, obtidos em (WANG, 2006).

Região	a_0	a_1	a_2	a_3	a_4	a_5
$0^\circ < \theta_{ijk} < 109,47^\circ$	0,27186	0,48922	-0,43286	-0,56140	1,27111	-0,03793
$109,47^\circ < \theta_{ijk} < 120^\circ$	0,69669	5,54440	23,4320	55,9476	69,8760	35,3117
$120^\circ < \theta_{ijk} < 180^\circ$	0,28160	1,06200	2,13400	2,53000	1,55200	0,38560

Tabela 2.2: Pontos conhecidos para aproximação da função γ_c .

$\theta_{ijk}(rad)$	γ_c
0	1
$\pi/3$	0,416335
$\pi/2$	0,271856

$$g_c(\cos(\theta_{ijk})) \approx \sum_{n=0}^5 a_n (\cos(\theta_{ijk}))^n \quad (2.8)$$

O valor de γ_c é aproximada por uma *spline* cúbica que tem os pontos conhecidos obtidos em (BRENNER et al., 2002) e exibidos na Tabela 2.2.

N_i^C (Eq. 2.9) e N_i^H (Eq. 2.10) são os números de conjugação de átomos de carbono e hidrogênio, respectivamente, do átomo i .

$$N_i^C = \sum_{k(\neq i,j)}^{\text{carbono}} f_{ik}^c(r_{ik}) \quad (2.9)$$

$$N_i^H = \sum_{l(\neq i,j)}^{\text{hidrogênio}} f_{il}^c(r_{il}) \quad (2.10)$$

Para estruturas de carbono em estado sólido, os valores de λ_{ijk} e da *spline* bicúbica P_{ij} são iguais a zero (BRENNER et al., 2002).

O valor de N_i^t é originalmente definido conforme a Equação 2.11. Porém, neste trabalho não são modelados átomos de hidrogênio e, portanto, o valor de N_i^H é zero.

$$N_i^t = N_i^C + N_i^H \quad (2.11)$$

A função $Q(N_i^t)$ é definida pela Equação 2.12.

$$Q(N_i^t) = \begin{cases} 1 & N_i^t < 3,2 \\ \frac{1 + \cos(2\pi(N_i^t - 3,2))}{2} & 3,2 < N_i^t < 3,7 \\ 0 & N_i^t > 3,7 \end{cases} \quad (2.12)$$

O termo b_{ij}^{RC} (Eq. 2.13) está relacionado com a existência ou não de caráter radical na ligação entre os átomos i e j e se a ligação é parte de um sistema conjugado. Seu valor foi calculado através de uma interpolação por *spline*, com os pontos conhecidos obtidos em (BRENNER et al., 2002) e (WANG, 2006).

$$b_{ij}^{RC} = F_{ij}(N_i^t, N_j^t, N_{ij}^{conj}) \quad (2.13)$$

A função N_{ij}^{conj} é dada por 2.14. A função de corte f^c seleciona os vizinhos do átomo i como os átomos k e os vizinhos do átomo j como átomos l .

$$N_{ij}^{conj} = 1 + \left[\sum_{k(\neq i,j)} f_{ik}^c(r_{ik}) H(x_{ik}) \right]^2 + \left[\sum_{l(\neq i,j)} f_{jl}^c(r_{jl}) H(x_{jl}) \right]^2 \quad (2.14)$$

O valor de x_{ik} é dado pela Equação 2.15 e o de $H(x_{ik})$ é dado pela Equação 2.16.

$$x_{ik} = N_k^t - f_{ik}^c(r_{ik}) \quad (2.15)$$

$$H(x_{ik}) = \begin{cases} 1 & x_{ik} < 2 \\ \frac{1 + \cos(2\pi(x_{ik} - 2))}{2} & 2 < x_{ik} < 3 \\ 0 & x_{ik} > 3 \end{cases} \quad (2.16)$$

O valor do termo b_{ij}^{DH} (Eq. 2.18) depende do ângulo diedral Θ_{ijkl} formado em ligações duplas entre os átomos de carbono (Fig. 2.1, obtida em (WANG, 2006)). O cosseno desse ângulo é calculado conforme a Equação 2.17.

$$\cos\Theta_{ijkl} = \frac{\vec{r}_{ik} \times \vec{r}_{ij}}{r_{ik}r_{ij}\text{sen}\theta_{kij}} \cdot \frac{\vec{r}_{jl} \times \vec{r}_{ji}}{r_{jl}r_{ji}\text{sen}\theta_{ijl}} = \left(\frac{\vec{r}_{jl} \cdot \vec{r}_{ik}}{r_{jl}r_{ik}} + \cos\theta_{kij}\cos\theta_{ijl} \right) \frac{-1}{\text{sen}\theta_{kij}\text{sen}\theta_{ijl}} \quad (2.17)$$

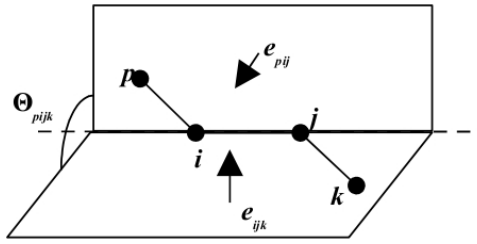


Figura 2.1: Ângulo diedral formado pelos átomos p , i , j e k .

Tabela 2.3: Valores para a função T_{ij} .

N_i^t	N_j^t	N_{ij}^{conj}	$T_{ij}(N_i^t, N_j^t, N_{ij}^{conj})$
2	2	1	-0,070280085
2	2	2, ... ,10	-0,00809675

$$b_{ij}^{DH} = T_{ij}(N_i^t, N_j^t, N_{ij}^{conj}) \left[\sum_{k(\neq i, j)} \sum_{l(\neq i, j)} (1 - \cos^2(\Theta_{ijkl})) f_{ik}^c(r_{ik}) f_{jl}^c(r_{jl}) \right] \quad (2.18)$$

Os valores para a função T_{ij} , obtidos em (BRENNER et al., 2002) e (WANG, 2006), são dados pela Tabela 2.3. Para valores não tabelados de N_i^t , N_j^t e N_{ij}^{conj} o valor de T_{ij} é zero.

2.2 Dinâmica molecular

A simulação de dinâmica molecular é um processo determinista que fornece a evolução temporal do sistema mediante a utilização das leis do movimento de Newton. A força \vec{F}_i atuante em cada átomo i é a soma da influência de todas as ligações feitas por esse átomo e pode ser dada pelo negativo do gradiente da energia potencial no átomo (Eq. 2.19). O cálculo desse gradiente é explicado na seção seguinte.

$$\vec{F}_i = -\vec{\nabla}_i E_b \quad (2.19)$$

Aplicando operações algébricas sobre a equação da Segunda Lei de Newton ($\vec{F}_i = m\vec{a}_i$), obtemos a expressão para o vetor aceleração \vec{a}_i em cada partícula, dada pela Equação 2.20, onde m é a massa de um átomo de carbono (aproximadamente $1,99442341 \times 10^{-26}$ kg).

$$\vec{a}_i = \frac{\vec{F}_i}{m} \quad (2.20)$$

De posse do vetor aceleração é possível realizar a integração das equações de movimento. Neste trabalho foi utilizado o modelo preditor-corretor de Nordsieck para a integração. Este modelo é dividido em duas fases (predição e correção), exigindo que o gradiente da energia em cada átomo i seja calculado duas vezes por iteração. Apesar de demandar um maior esforço computacional quando comparado com outros integradores (*Leapfrog*, por exemplo), o modelo de Nordsieck apresenta uma aproximação numérica melhor que a obtida com integradores de uma fase.

Para controlar a energia do sistema um termostato simples foi desenvolvido: a cada iteração o vetor velocidade de todos os átomos é reescalado.

2.3 Gradiente do Potencial REBO2

O gradiente da energia potencial em cada partícula, utilizado para calcular a força resultante e realizar a dinâmica molecular, é obtido pela Equação 2.21.

$$\vec{\nabla}_i E_b = \sum_j \left[\vec{\nabla}_i V^R(r_{ij}) - V^A(r_{ij}) \vec{\nabla}_i b_{ij} - b_{ij} \vec{\nabla}_i V^A(r_{ij}) \right] \quad (2.21)$$

O termo $\vec{\nabla}_i V^R(r_{ij})$ é mostrado pela Equação 2.22.

$$\vec{\nabla}_i V^R(r_{ij}) = (1 + Q/r_{ij}) A e^{-\alpha r_{ij}} \vec{\nabla}_i f^c(r_{ij}) - f^c(r_{ij}) (\alpha + \alpha Q/r_{ij} + Q/r_{ij}^2) A e^{-\alpha r_{ij}} \vec{\nabla}_i r_{ij} \quad (2.22)$$

O gradiente do termo atrativo é dado pela Equação 2.23.

$$\vec{\nabla}_i V^A(r_{ij}) = \sum_{n=1}^3 B_n e^{-\beta_n r_{ij}} \vec{\nabla}_i f^c(r_{ij}) - f^c(r_{ij}) \left[\sum_{n=1}^3 \beta_n B_n e^{-\beta_n r_{ij}} \right] \vec{\nabla}_i r_{ij} \quad (2.23)$$

A função de corte $f^c(r_{ij})$ tem o gradiente dado pela Equação 2.24.

$$\vec{\nabla}_i f^c(r_{ij}) = \begin{cases} 0,0 & r_{ij} < 1,7 \\ -\text{sen}\left(\frac{\pi(r_{ij}-1,7)}{0,3}\right) \frac{\pi}{0,6} \vec{\nabla}_i r_{ij} & 1,7 \leq r_{ij} \leq 2,0 \\ 0,0 & r_{ij} > 2,0 \end{cases} \quad (2.24)$$

O gradiente $\vec{\nabla}_i r_{ij}$ é calculado conforme a Equação 2.25.

$$\vec{\nabla} r_{ij} = -\frac{\vec{r}_{ij}}{|\vec{r}_{ij}|} \quad (2.25)$$

O valor de $\vec{\nabla}_i b_{ij}$ é dado pela Equação 2.26.

$$\vec{\nabla}_i b_{ij} = \frac{\vec{\nabla}_i b_{ij}^{\sigma-\pi} + \vec{\nabla}_i b_{ji}^{\sigma-\pi}}{2} + \vec{\nabla}_i b_{ij}^{RC} + \vec{\nabla}_i b_{ij}^{DH} \quad (2.26)$$

Os termos $\vec{\nabla}_i b_{ij}^{\sigma-\pi}$ e $\vec{\nabla}_i b_{ji}^{\sigma-\pi}$ são calculados de acordo com a Equação 2.27.

$$\begin{aligned} \vec{\nabla}_i b_{ij}^{\sigma-\pi} = & \frac{-(b_{ij}^{\sigma-\pi})^3}{2} \left\{ \vec{\nabla} P_{ij} \right. \\ & \left. + \sum_{k(\neq i,j)} \left[\sum_{n=0}^5 a_n (\cos \theta_{ijk})^n e^{\lambda_{ijk}} \vec{\nabla}_i f_{ik}^c + \sum_{n=1}^5 n a_n (\cos \theta_{ijk})^{n-1} e^{\lambda_{ijk}} f_{ik}^c \vec{\nabla} \cos \theta_{ijk} \right] \right\} \end{aligned} \quad (2.27)$$

onde

$$\vec{\nabla} \cos \theta_{ijk} = \vec{\nabla} \left[\frac{\vec{r}_{ik} \cdot \vec{r}_{ij}}{r_{ik} r_{ij}} \right] = \frac{\vec{\nabla} (\vec{r}_{ik} \cdot \vec{r}_{ij})}{r_{ik} r_{ij}} - \frac{\vec{r}_{ik} \cdot \vec{r}_{ij}}{(r_{ik})^2 r_{ij}} \vec{\nabla} r_{ik} - \frac{\vec{r}_{ik} \cdot \vec{r}_{ij}}{(r_{ij})^2 r_{ik}} \vec{\nabla} r_{ij}.$$

O gradiente $\vec{\nabla}_i b_{ij}^{RC}$ é dado pela Equação 2.28.

$$\vec{\nabla}_i b_{ij}^{RC} = \vec{\nabla}_i F_{ij} (N_i^t, N_j^t, N_{ij}^{conj}) \quad (2.28)$$

O termo $\vec{\nabla}_i b_{ij}^{DH}$ é obtido pela Equação 2.29.

$$\begin{aligned} \vec{\nabla}_i b_{ij}^{DH} = & \left[\sum_{k(\neq i,j)} \sum_{l(\neq i,j)} (1 - \cos^2 (\Theta_{ijkl})) f_{ik}^c (r_{ik}) f_{jl}^c (r_{jl}) \right] \vec{\nabla}_i T_{ij} (N_i^t, N_j^t, N_{ij}^{conj}) \\ & + T_{ij} (N_i^t, N_j^t, N_{ij}^{conj}) \left[\sum_{k(\neq i,j)} \sum_{l(\neq i,j)} (1 - \cos^2 (\Theta_{ijkl})) f_{ik}^c (r_{ik}) \vec{\nabla}_i f_{jl}^c (r_{jl}) \right] \\ & + T_{ij} (N_i^t, N_j^t, N_{ij}^{conj}) \left[\sum_{k(\neq i,j)} \sum_{l(\neq i,j)} (1 - \cos^2 (\Theta_{ijkl})) f_{jl}^c (r_{jl}) \vec{\nabla}_i f_{ik}^c (r_{ik}) \right] \\ & + T_{ij} (N_i^t, N_j^t, N_{ij}^{conj}) \left[\sum_{k(\neq i,j)} \sum_{l(\neq i,j)} f_{ik}^c (r_{ik}) f_{jl}^c (r_{jl}) (-2 \cos (\Theta_{ijkl})) \vec{\nabla} \cos (\Theta_{ijkl}) \right] \end{aligned} \quad (2.29)$$

O vetor $\vec{\nabla} \cos(\Theta_{ijkl})$ é calculado conforme a Equação 2.30.

$$\begin{aligned}
 \vec{\nabla} \cos \Theta_{ijkl} &= \vec{\nabla} \left[\left(\frac{\vec{r}_{jl} \cdot \vec{r}_{ik}}{r_{jl} r_{ik}} + \cos \theta_{kij} \cos \theta_{ijl} \right) \frac{-1}{\text{sen} \theta_{kij} \text{sen} \theta_{ijl}} \right] \\
 &= \left[\vec{\nabla} \left(\frac{\vec{r}_{jl} \cdot \vec{r}_{ik}}{r_{jl} r_{ik}} \right) + \cos \theta_{ijl} \vec{\nabla} \cos \theta_{kij} + \cos \theta_{kij} \vec{\nabla} \cos \theta_{ijl} \right] \frac{-1}{\text{sen} \theta_{kij} \text{sen} \theta_{ijl}} \\
 &\quad + \cos \Theta_{ijkl} \left[\frac{\cos \theta_{ijl}}{(\text{sen} \theta_{ijl})^2} \vec{\nabla} \cos \theta_{ijl} + \frac{\cos \theta_{kij}}{(\text{sen} \theta_{kij})^2} \vec{\nabla} \cos \theta_{kij} \right]
 \end{aligned} \tag{2.30}$$

3 *Modelo Computacional*

Uma abordagem frequentemente utilizada para o desenvolvimento de simulações distribuídas de dinâmica molecular é o particionamento espacial da estrutura de interesse em blocos de partículas e a divisão desses blocos entre as unidades de processamento disponíveis. Torna-se necessário, portanto, que simulações computacionais desse tipo apresentem estratégias para a manutenção da comunicação entre as unidades de processamento envolvidas, assim como um esquema de particionamento eficiente do espaço, de forma a promover um bom balanceamento da carga de trabalho.

Neste capítulo são apresentadas as decisões de projeto adotadas na construção do simulador de dinâmica molecular no que diz respeito ao controle da simulação, à comunicação entre os processos participantes da computação e ao particionamento espacial da estrutura molecular.

3.1 Sistema de Comunicação

O objetivo do desenvolvimento de um sistema de comunicação próprio é prover um mecanismo para troca de mensagens pela rede de forma simples e que minimize o *overhead* associado à comunicação. Assim, o sistema de comunicação foi desenvolvido utilizando orientação a objetos, de maneira a encapsular as chamadas de sistema responsáveis pelo estabelecimento, gerenciamento, utilização e finalização de conexões com múltiplos nós de forma independente. Isso simplifica a implementação de sistemas baseados em troca de mensagens.

O sistema desenvolvido segue o paradigma cliente-servidor, sendo dividido em dois módulos, que são apresentados a seguir.

3.1.1 Módulo Servidor

O módulo servidor possibilita que um nó aguarde por conexões e trate as mensagens recebidas de outros nós. Este módulo é formado por três classes, como ilustra a Figura 3.1. A

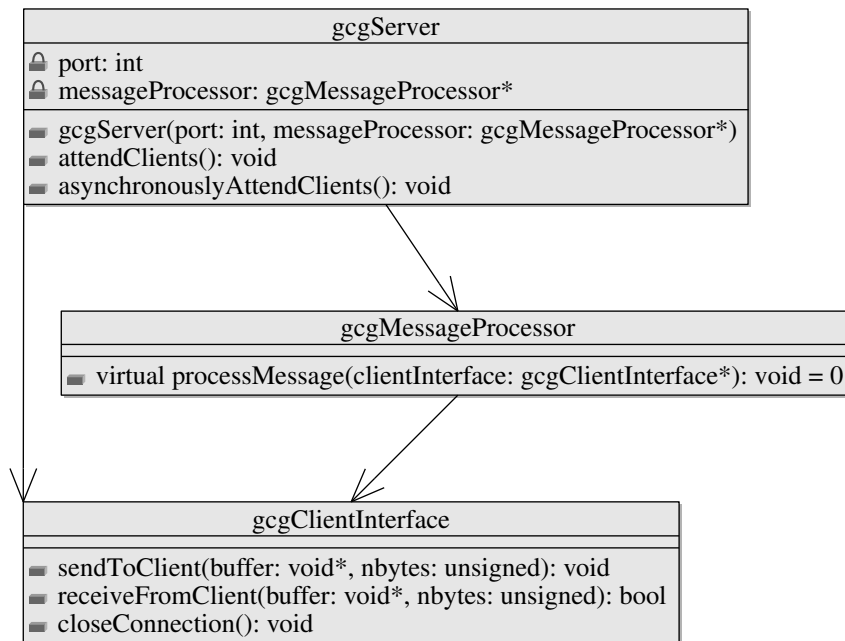


Figura 3.1: Diagrama de classes para o módulo servidor. Alguns métodos e classes foram omitidos, por simplificação.

classe *gcgServer* é responsável por reservar uma porta junto ao sistema operacional para a conexão dos clientes e por gerenciar o processo de criação e término de conexões. Para cada instância da classe *gcgServer* são associadas uma porta que receberá as conexões dos clientes e uma instância de uma classe que herde de *gcgMessageProcessor*. Essa classe deve implementar o método abstrato *processMessage* com a lógica necessária ao tratamento das mensagens recebidas pelo servidor.

Quando um cliente inicia uma conexão na porta “escutada” pelo servidor, uma instância da classe *gcgClientInterface* é criada. Esta é fornecida como parâmetro para o método *processMessage* da instância de *gcgMessageProcessor* associada ao objeto *gcgServer* que recebeu a conexão. A classe *gcgClientInterface* expõe métodos para a troca de mensagens com o cliente que realizou a conexão, bem como uma maneira simplificada para provocar a interrupção da conexão pelo servidor.

3.1.2 Módulo Cliente

O módulo cliente permite que um nó inicie uma conexão em um servidor e as mensagens sejam trocadas. É formado pelas classes *gcgClient* e *gcgClientGroup* (Fig. 3.2). Cada instância da classe *gcgClient* representa uma conexão com um servidor. A classe implementa rotinas para a troca de mensagens entre o cliente e o servidor, de forma similar à disponibilizada pela classe *gcgServer*.

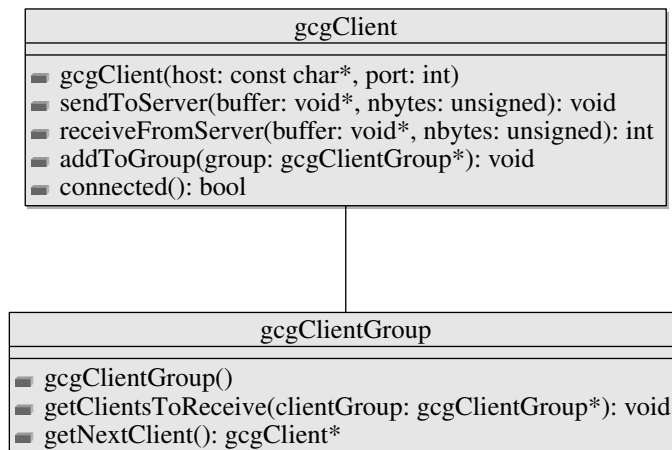


Figura 3.2: Diagrama de classes para o módulo cliente. Alguns métodos foram omitidos, por simplificação.

Existem situações em que é necessário enviar mensagens a diversos servidores e aguardar a resposta de todos. Porém, é possível que esses servidores apresentem diferentes tempos de resposta. Neste caso, realizar o tratamento sequencial das respostas recebidas é inviável. Torna-se interessante, então, prover a funcionalidade de aguardar simultaneamente pela resposta de todos os servidores. Os objetos da classe *gcgClient* podem ser agrupados em coleções, representadas por instâncias da classe *gcgClientGroup*. Dessa forma, é possível obter conjuntos de objetos da classe *gcgClient* que já receberam as mensagens de resposta e fazer o tratamento dessas mensagens antes mesmo do recebimento das mensagens de todos os servidores.

3.2 Protocolo de Controle da Simulação

3.2.1 Tipos de Processo

O sistema foi construído seguindo a hierarquia mestre-escravo. Nessa abordagem há um conjunto de processos idênticos e especializados (chamados *processos escravos*) que têm seu funcionamento determinado por um processo mestre. Neste trabalho o processo mestre atende requisições de um terceiro tipo de processo, que faz a interface com o usuário. A Figura 3.3 mostra uma visão geral dos componentes do sistema e indica as instâncias das classes *gcgClient* e *gcgServer*. Cabe dizer que cada processo pode ter sua execução realizada em uma unidade de processamento diferente.

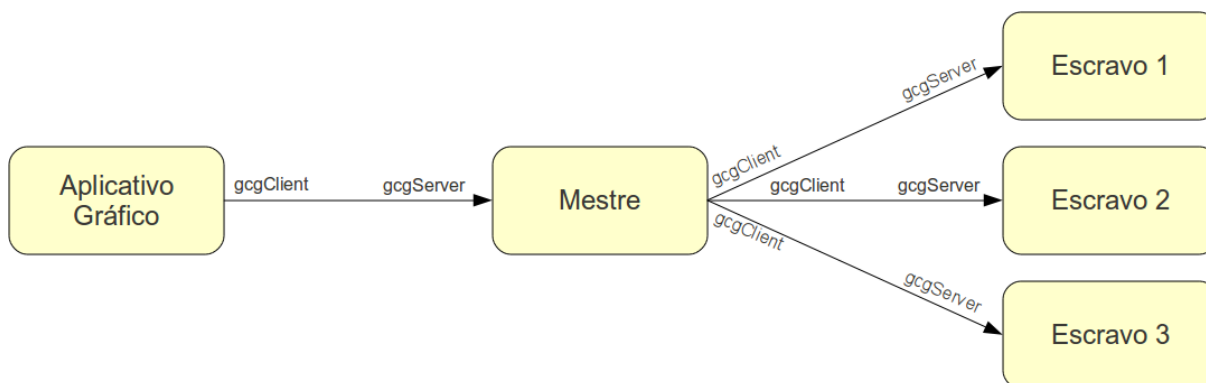


Figura 3.3: Arquitetura do sistema. O sentido das setas indica o sentido do estabelecimento das conexões.

3.2.2 Inicialização do Ambiente de Simulação

O processo mestre é iniciado de forma não automatizada e disponibiliza, no início de sua execução, uma porta para que o processo com a interface gráfica possa se conectar. De forma semelhante, cada processo escravo disponibiliza uma porta à qual o processo mestre se conecta para fazer o controle da simulação (isto é, cada processo escravo atua como um servidor de processamento). A porta a ser associada a cada processo escravo é definida por parâmetro durante sua inicialização. Para automatizar a inicialização dos processos escravos foi criado um *script* que inicializa um conjunto de processos escravos em *background* e define a porta a ser monitorada por cada processo.

Ao iniciar a aplicação gráfica o usuário deve indicar o arquivo, no formato PLY, com a posição de cada átomo da estrutura de interesse. O formato PLY é um formato de armazenamento de modelos geométricos. A posição inicial de todas as partículas da estrutura é enviada para o processo mestre, que estabelece a conexão com os processos escravos e faz o particionamento espacial da estrutura. Cada partição de espaço tem seus átomos delegados a um processo escravo. O processo mestre pode, então, iniciar a simulação de dinâmica molecular.

3.2.3 Aplicativo Gráfico de Acesso e Controle

Durante a execução da simulação o usuário pode utilizar a interface gráfica para visualizar a distribuição dos átomos na estrutura. O aplicativo fornece a funcionalidade de atualização da posição dos átomos, que deve ser utilizada com parcimônia para não comprometer os recursos computacionais utilizados pelo processo mestre. É permitido ao usuário desconectar a aplicação de controle do processo mestre durante a simulação. Ao se reconectar ao processo mestre, a aplicação obtém a posição dos átomos da estrutura utilizada e atualiza a visualização. O usuário

pode interromper a simulação quando julgar necessário.

3.2.4 Protocolo de Comunicação

Todas as mensagens seguem um protocolo de comunicação que exige que elas sejam formadas por um *byte* que indique o tipo da mensagem, seguido por uma quantidade arbitrária de dados. O tratamento das mensagens recebidas por um processo é feito de forma sequencial e sem acesso concorrente, isto é, somente uma mensagem é lida do *buffer* por vez. O processo mestre conduz a simulação enviando mensagens a todos os escravos e aguardando a resposta de cada um deles. Dessa maneira, a comunicação entre o processo mestre e os processos escravos é sincronizada com barreiras a cada passo de simulação e todos os processos escravos realizam o tratamento da mesma estrutura de dados (definida pelo tipo da mensagem) simultaneamente. Essa sincronia entre os processos escravos é importante pois algumas etapas da simulação exigem que aconteçam trocas de dados entre os processos escravos, por intermédio do processo mestre.

3.3 Particionamento de Espaço e Balanceamento de Carga

O particionamento da estrutura é feito por planos perpendiculares ao eixo de maior variação da posição dos átomos. Esses planos definem tantas partições de espaço quantos forem os processos escravos disponíveis para computação. As partições de espaço têm aproximadamente as mesmas dimensões.

O particionamento da estrutura e a classificação das partículas (Seção 3.3.1) são recalculados a cada duas iterações de dinâmica molecular para diminuir os erros causados por vizinhanças atômicas desatualizadas.

Esta é uma solução de implementação simplificada pois está fora do escopo deste trabalho o desenvolvimento de um particionamento espacial que forneça um balanceamento de carga ótimo. Entretanto, esse modelo fornece um balanceamento de carga aceitável nos casos em que as unidades de processamento utilizadas pelos processos escravos são semelhantes e o número de átomos está igualmente distribuído entre os processos.

3.3.1 Classificação de partículas

O cálculo da função de energia potencial REBO2 em um átomo depende da posição dos seus vizinhos de primeiro e segundo níveis. São considerados *vizinhos de primeiro nível* (ou

primeiros vizinhos) os átomos $n_1(i)$ localizados a uma distância de 2\AA ou menos do átomo i . Os *vizinhos de segundo nível* $n_2(i)$ de um átomo i são os vizinhos de primeiro nível $n_1(j)$ de pelo menos um átomo j que seja vizinho de primeiro nível de i : $j \in n_1(i)/i$. Desse modo, para calcular a energia potencial de um átomo é importante conhecer a posição de todos os átomos que estejam a uma distância de 4\AA ou menos desse átomo. Mostra-se necessário, assim, que cada processo escravo tenha acesso não só à posição dos átomos que têm a dinâmica sob sua responsabilidade, mas também à posição de todos os átomos que estejam a uma distância de até 4\AA dos limites do volume de átomos a ele designado.

A seguinte classificação define a relação entre um átomo e um processo escravo:

- Classe 0: átomos que estão fora da área de atuação do processo;
- Classe 1: átomos delegados ao processo e localizados a uma distância de mais de 4\AA das fronteiras do volume. Visíveis somente pelo processo corrente;
- Classe 2: átomos delegados ao processo e localizados a uma distância de 4\AA ou menos das fronteiras do volume. Visíveis por outros processos;
- Classe 3: átomos vistos pelo processo corrente mas delegados a outro processo.

É fácil notar que, com essa classificação, garante-se que, para cada átomo, independente do tipo de particionamento, acontece uma das duas situações:

- O átomo é da classe 1 para algum processo escravo e é da classe 0 para todos os outros ou
- O átomo é da classe 2 para algum processo escravo e é da classe 3 para pelo menos um outro processo escravo.

A Figura 3.4 ilustra uma região de fronteira entre dois processos escravos. Os átomos nas regiões “A” e “B” têm sua função de energia avaliada pelo processo escravo P_1 e sofrem dinâmica molecular por ação desse processo. Já os átomos nas regiões “C” e “D” têm sua função de energia avaliada pelo processo escravo P_2 e sofrem dinâmica molecular por ação desse processo. Os átomos na região “A” são classificados como átomos de classe 1 pelo processo P_1 e não têm cópia no processo P_2 . De maneira análoga, os átomos na região “D” são classificados como átomos de classe 1 pelo processo P_2 e não têm cópia no processo P_1 . Os átomos na região “B” são classificados como átomos de classe 2 pelo processo P_1 e como átomos de classe 3 pelo processo P_2 . O processo P_1 é responsável pela avaliação da energia e pela dinâmica

molecular desses átomos, notificando as alterações de posição ao processo mestre, que por sua vez notifica o processo P_2 . Os átomos na região “C” são classificados como átomos de classe 2 pelo processo P_2 e como átomos de classe 3 pelo processo P_1 . O processo P_2 é responsável pela avaliação da energia e pela dinâmica molecular desses átomos, notificando as alterações de posição ao processo mestre, que por sua vez notifica o processo P_1 .

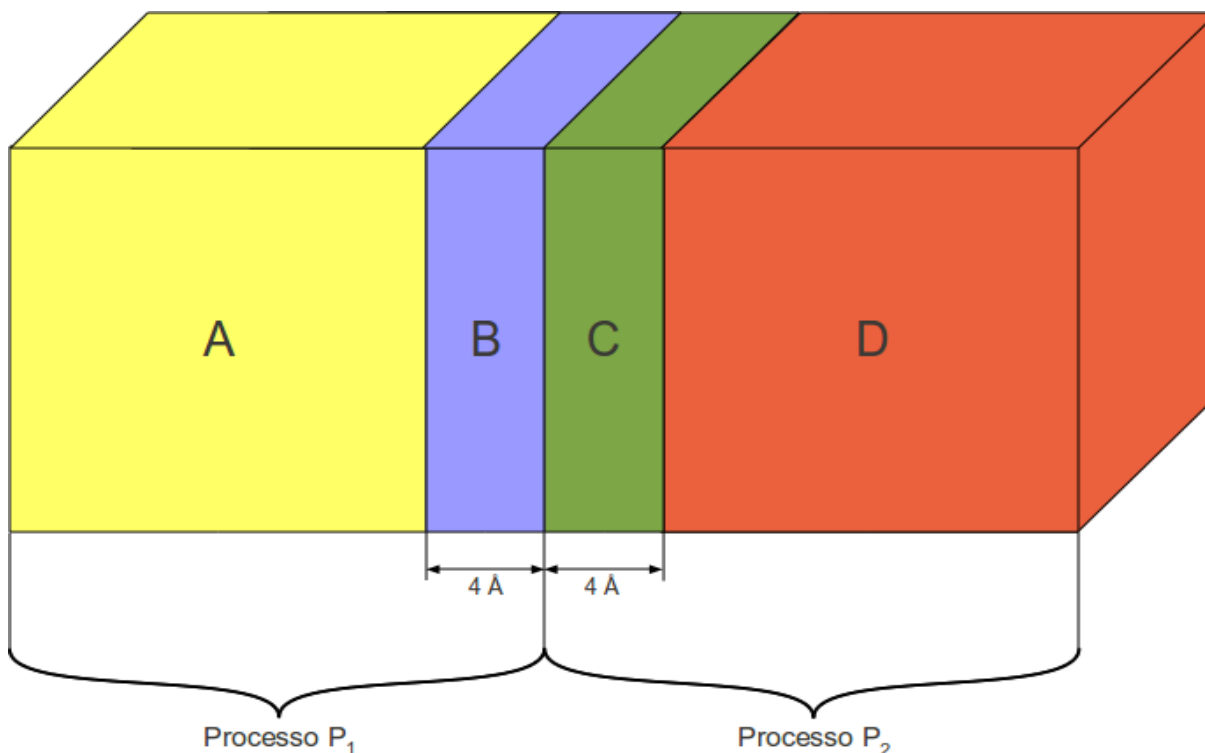


Figura 3.4: Exemplo da região de fronteira entre dois processos escravos, mostrando os volumes que definem diferentes classificações para os átomos em cada processo escravo.

A classificação inicial de todos os átomos é realizada pelo processo mestre, quando a aplicação gráfica envia a esse processo as informações de posição inicial dos átomos. O processo mestre envia a cada processo escravo apenas as informações dos átomos classificados como de classe 1, 2 ou 3 para o processo escravo. A cada passo de iteração o processo mestre obtém dos processos escravos os novos limites espaciais para a estrutura e refaz o particionamento, redefinindo para cada processo escravo as regiões que contêm átomos de classe 1, 2 e 3. Cada processo escravo reclassifica todos os átomos antes classificados como de classe 1 ou 2 e notifica (somente para átomos que sofreram mudança de classe) essas modificações ao processo mestre. As notificações dos processos escravos e a reação do processo mestre a cada notificação são exibidas pelas Figuras 3.5 a 3.10.

Todas as alterações nos átomos de classe 2 de cada processo escravo são notificadas ao processo mestre. O processo mestre se encarrega de notificar os processos nos quais o átomo é de classe 3 seu novo estado. A Figura 3.11 ilustra esse caso.

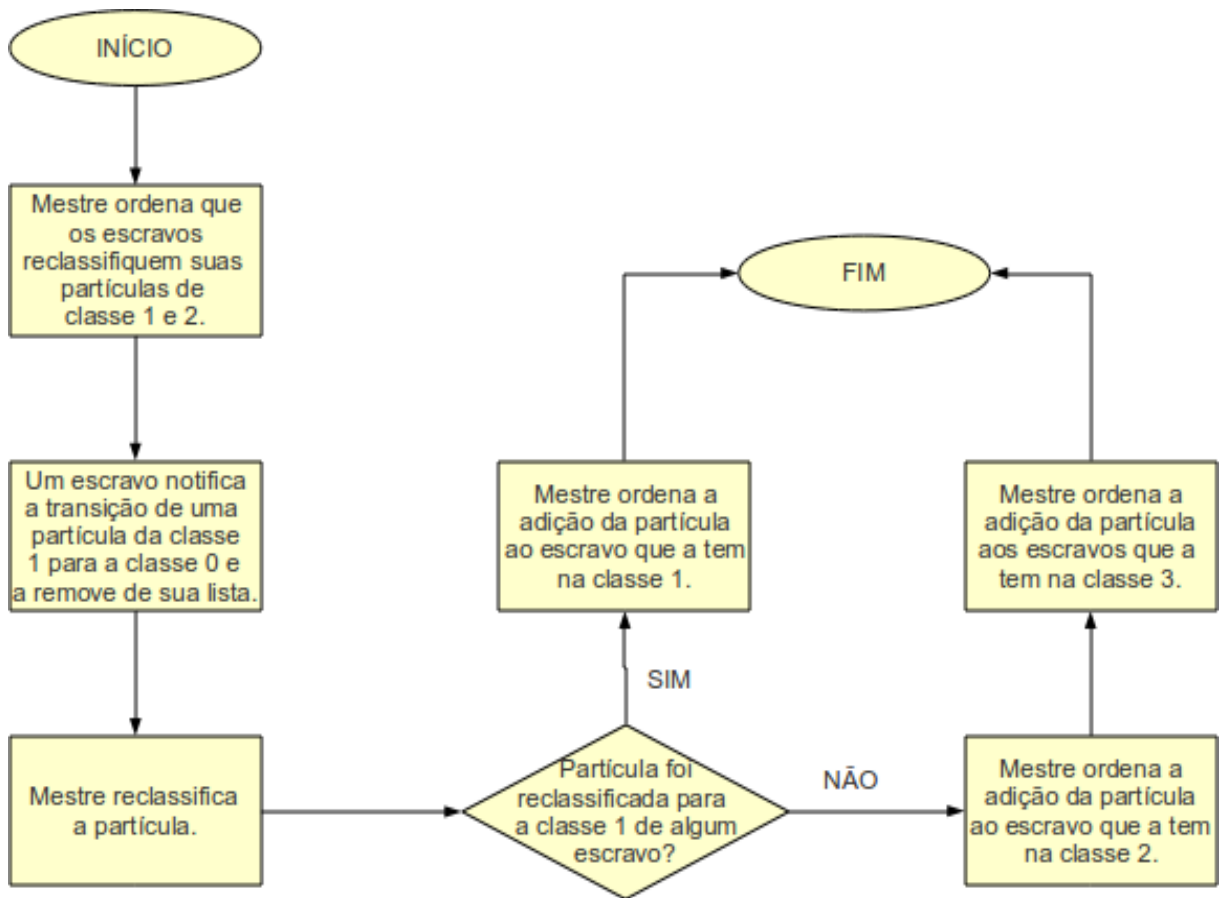


Figura 3.5: Reclassificação de um átomo da classe 1 para a classe 0 em um processo escravo.

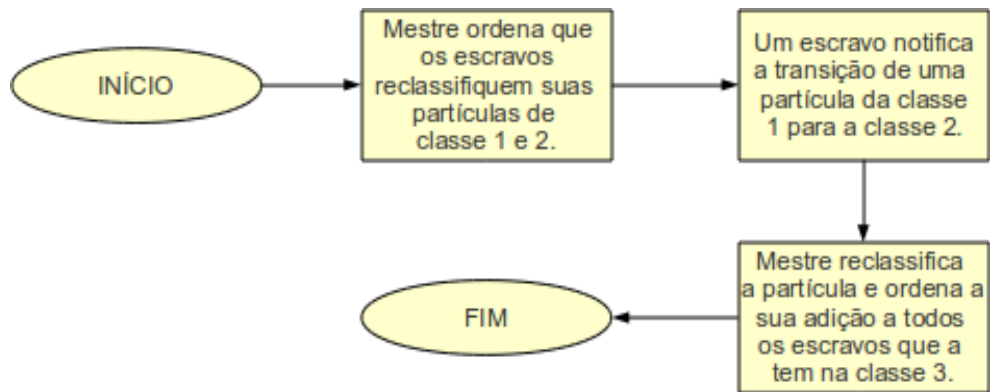


Figura 3.6: Reclassificação de um átomo da classe 1 para a classe 2 em um processo escravo.

Esta classificação tornou-se efetiva e é uma contribuição desta monografia.

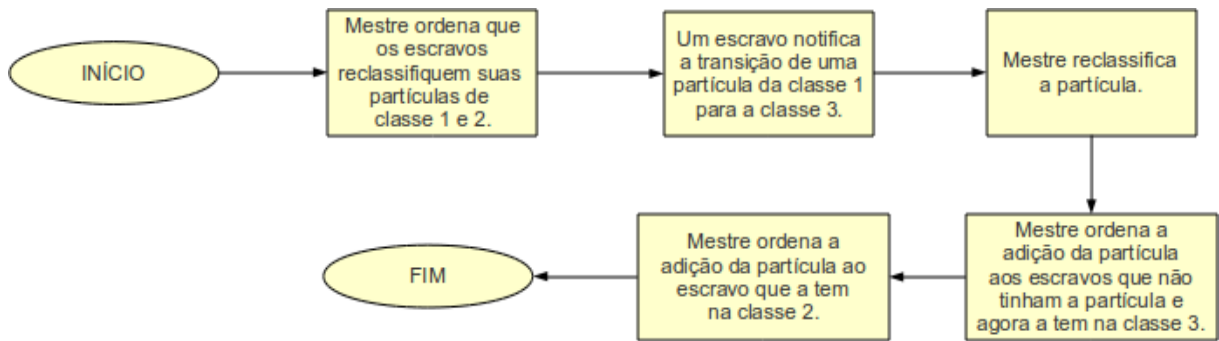


Figura 3.7: Reclassificação de um átomo da classe 1 para a classe 3 em um processo escravo.

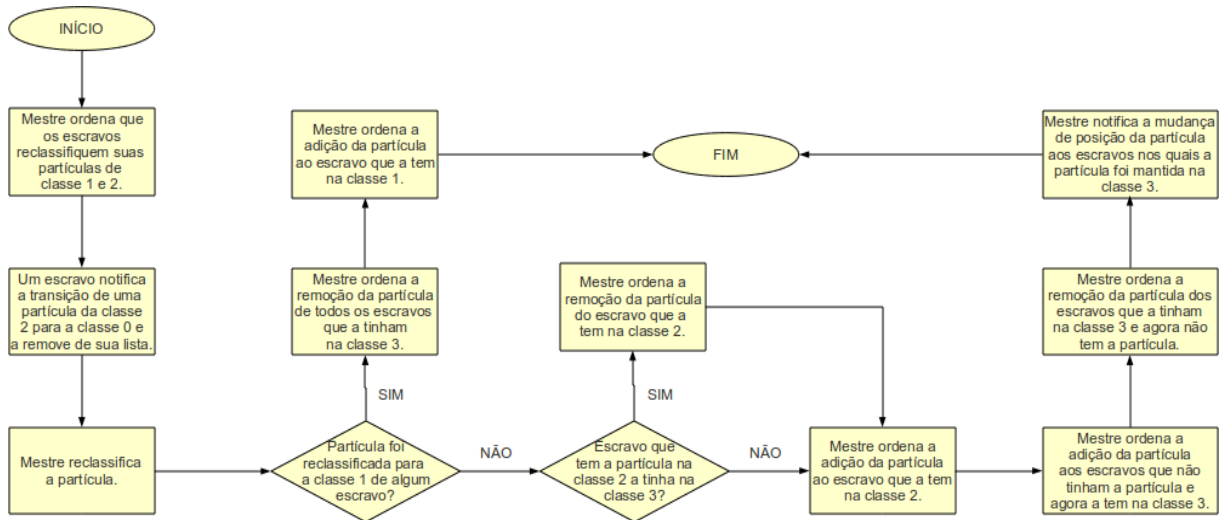


Figura 3.8: Reclassificação de um átomo da classe 2 para a classe 0 em um processo escravo.

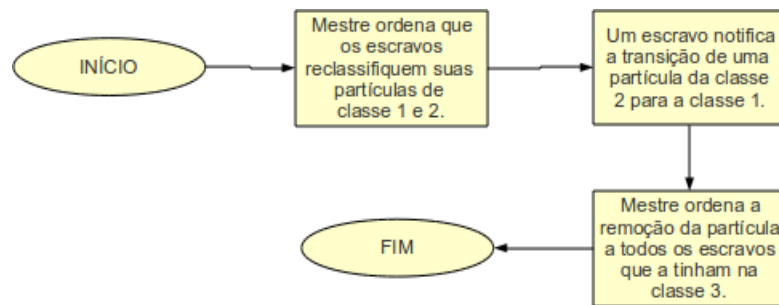


Figura 3.9: Reclassificação de um átomo da classe 2 para a classe 1 em um processo escravo.

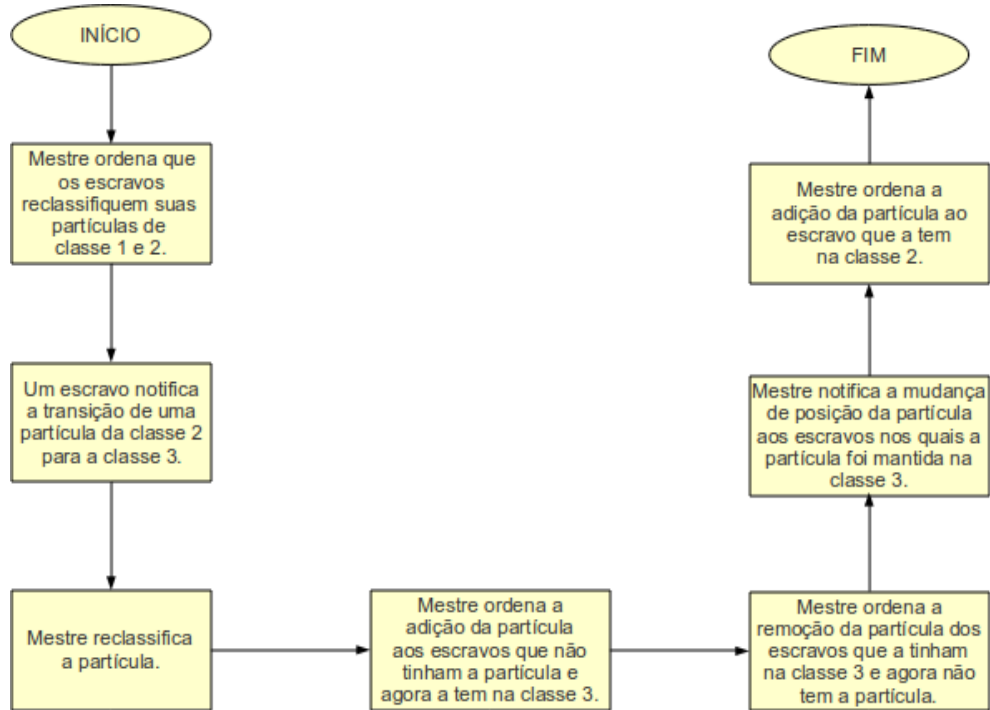


Figura 3.10: Reclassificação de um átomo da classe 2 para a classe 3 em um processo escravo.

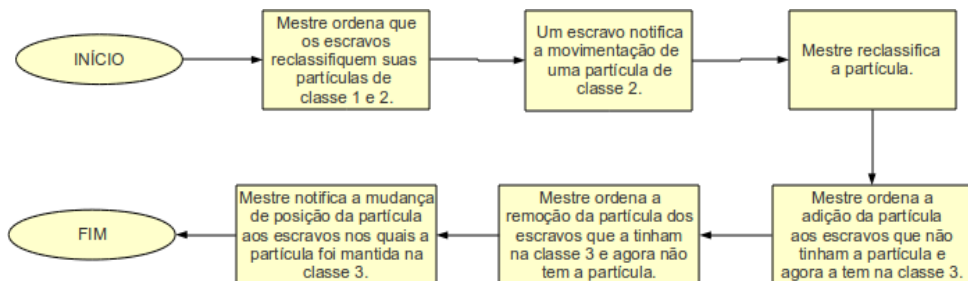


Figura 3.11: Notificação da movimentação de um átomo da classe 2 em um processo escravo.

4 *Resultados Experimentais*

Neste capítulo são apresentados os resultados obtidos com o programa desenvolvido durante este trabalho. A Figura 4.1 mostra a interface gráfica da aplicação durante uma simulação de dinâmica molecular de um nanotubo com 100.100 átomos utilizando 16 processos escravos.

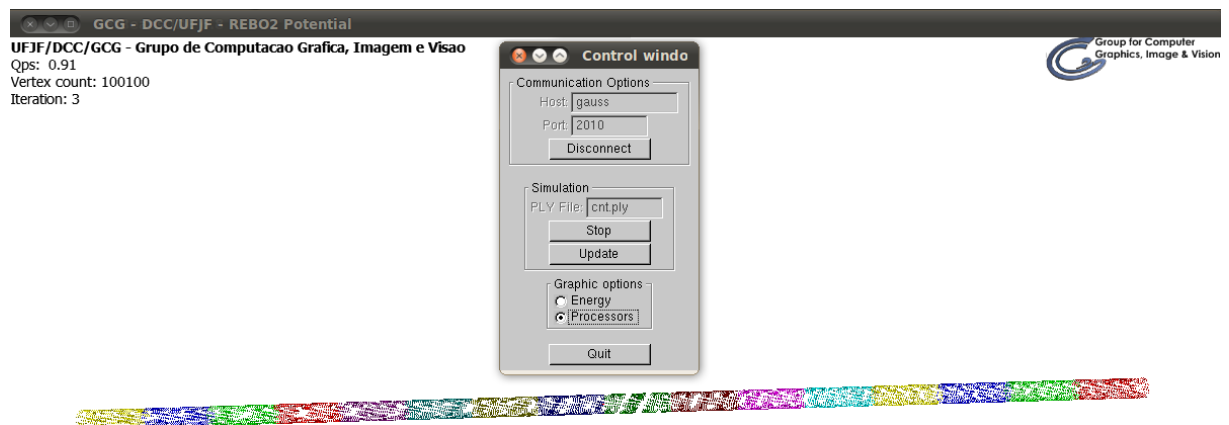


Figura 4.1: Interface gráfica do simulador. A estrutura exibida tem 100.100 átomos. As cores indicam o particionamento espacial que divide a estrutura entre 16 processos escravos.

Para a análise de desempenho foram utilizados quatro computadores com especificações técnicas diferentes (Tabela 4.1). A versão do sistema operacional utilizado em todos os computadores é Ubuntu Desktop Edition 10.04 64bits.

Tabela 4.1: Comparação entre os computadores utilizados para análise de desempenho.

Computador	Processador	Memória RAM
1	2 x Intel Xeon Quad Core E5410 Ativo 2.33GHz 12MiB 1.333MHz	4GiB/DDR2/1.066MHz
2	Intel Core i7-940 2.93GHz 8MiB 1.333MHz	4GiB/DDR3/1.600MHz
3	Intel Core2 Quad Q8200 2.33GHz 4MiB 1.333MHz	4GiB/DDR2/1.066MHz
4	Intel Core2 Quad Q6600 2.40GHz 8MiB 1.066MHz	4GiB/DDR2/667MHz

Tabela 4.2: Número de processos escravos utilizados em cada computador, para cada cenário, no caso de processamento distribuído.

Cenário	Computador 1	Computador 2	Computador 3	Computador 4	Total
1	8	0	0	0	8
2	8	8	0	0	16
3	8	8	4	0	20
4	8	8	4	4	24

Tabela 4.3: Tempo médio de execução de uma iteração de simulação para um nanotubo de 100.000 átomos (em segundos).

	Cenário 1	Cenário 2	Cenário 3	Cenário 4
Tempo	31,16	16,15	13,51	13,86
<i>Speedup</i>	15,05	29,03	34,70	33,83

4.1 Estudo de Caso

Um conjunto de cenários foi definido com o objetivo de observar o desempenho do sistema sob diferentes condições. A análise é baseada no tempo médio de execução de uma iteração de dinâmica molecular. No caso sequencial foram computadas 15 iterações, enquanto nos casos em que há processamento distribuído foram usadas 100 iterações. O particionamento espacial e a classificação das partículas, para os casos com processamento distribuído, são recalculados a cada duas iterações. Em todos os casos a vizinhança de cada átomo foi recalculada a cada iteração. A métrica utilizada para avaliar o desempenho é o *speedup* em relação à execução sequencial, que aconteceu utilizando o computador 2. Em todos os cenários de computação distribuída a execução do processo mestre foi realizada pelo computador 1. A Tabela 4.2 mostra o número de processos escravos utilizados no processamento distribuído em cada computador para os cenários avaliados.

A estrutura utilizada conta com 100.100 átomos de carbono e comprimento de 1.500Å. O processamento sequencial teve média de 468,9 segundos por iteração. A Tabela 4.3 compara o tempo médio por iteração e o *speedup* alcançado para os 4 cenários definidos.

É importante ressaltar que os resultados apresentados são resultados instantâneos, obtidos com uma execução de cada cenário.

Os resultados mostram que o *speedup* alcançado no cenário 4 (24 processos escravos) foi menor que o *speedup* no cenário 3 (20 processos escravos). Para esses dois cenários foram realizadas mais duas execuções com 100 iterações cada. Os resultados são exibidos na Tabela 4.4.

Tabela 4.4: Tempo médio de execução de uma iteração de simulação para um nanotubo de 100.000 átomos (em segundos) – execuções extras para os cenários 3 e 4.

	Cenário 3	Cenário 4
Execução 1	13,51	13,86
Execução 2	13,54	13,88
Execução 3	13,54	13,66
Média	13,53	13,80
Desvio padrão	0,02	0,12
<i>Speedup</i>	34,66	33,98

Algumas hipóteses podem ser consideradas para tentar explicar esse comportamento: influência do número de mensagens trocadas, balanceamento de carga irregular ou, ainda, sobrecarga de trabalho no processo mestre. Entretanto, como não foram realizadas muitas execuções com esses cenários, não se pode afirmar que as amostras analisadas fornecem o comportamento médio do sistema.

5 *Conclusão*

Este trabalho apresenta uma estratégia para computação distribuída de simulação de dinâmica molecular em nanoestruturas de carbono usando a função de energia potencial REBO2. Todos os módulos necessários ao funcionamento do sistema (visualização, comunicação e simulação) foram implementados.

O modelo computacional elaborado é baseado na hierarquia mestre-escravo. Uma aplicação gráfica fornece a posição dos átomos da estrutura ao processo mestre. A divisão do trabalho entre os processos escravos ocorre por meio de particionamento espacial da estrutura de interesse. A simulação de dinâmica molecular acontece de forma sincronizada entre os processos escravos, sendo coordenada pelo processo mestre.

O esquema utilizado para particionamento espacial é o de corte simples por planos. A cada processo escravo é atribuído um conjunto de átomos e cada átomo tem sua dinâmica molecular computada por somente um processo escravo, a cada iteração. Como o potencial utilizado é de curto alcance e depende de informações de vizinhança atômica, é necessário manter uma cópia das informações dos átomos localizados próximo à fronteira entre o volume de dois processos escravos vizinhos.

Uma classificação de átomos para simulações distribuídas de dinâmica molecular foi proposta por este trabalho e tornou-se efetiva. As alterações da classe dos átomos são propagadas para os processos escravos que dependem das informações dessas partículas. A classificação proposta pode ser aplicada em outros sistemas que utilizem função de energia de curto alcance. Esta é uma contribuição desta monografia.

Foram realizados experimentos para análise de desempenho do simulador. Tais experimentos mostraram que o uso da estratégia de computação distribuída causou redução significativa no tempo de simulação. O aumento do número de processos escravos, entretanto, a partir de um certo limite, causa redução do *speedup*. Algumas hipóteses que poderiam explicar esse fato são o tempo gasto com comunicação, o balanceamento de carga utilizado ou a necessidade de trocar informações de partículas de classe 2 entre processos escravos. Uma explicação mais

precisa exigiria uma investigação maior, que não pôde ser feita por conta do tempo exíguo.

Para trabalhos futuros pretende-se modelar o comportamento de estruturas de carbono com adição de outros tipos de átomos, tais como hidrogênio e oxigênio. Observar o efeito do uso de uma função de energia potencial de longo alcance também seria interessante. Podem ainda ser implementados mecanismos mais bem-elaborados para promover o equilíbrio da energia. Outra proposta é a realização de estimativas do tráfego na rede de maneira a verificar qual a influência da troca de mensagens entre os processos no tempo necessário à simulação. Pretende-se, ainda, elaborar estratégias mais complexas de particionamento espacial para tentar promover um melhor balanceamento de carga entre os processos escravos.

Referências Bibliográficas

ABELL, G. C. Empirical chemical pseudopotential theory of molecular and metallic bonding. *Phys. Rev. B*, American Physical Society, v. 31, n. 10, p. 6184–6196, Maio 1985.

BRENNER, D. W. et al. A second-generation reactive empirical bond order (REBO) potential energy expression for hydrocarbons. *Journal of Physics: Condensed Matter*, v. 14, n. 4, p. 783, 2002. Disponível em: <<http://stacks.iop.org/0953-8984/14/i=4/a=312>>.

WANG, Z. *Reactive empirical bond-order (REBO) potential*. [S.l.], 01 2006.