

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

*Quantum Scenery: Um Novo Software de Química
Quântica Semiempírico Para Sistemas de Memória
Distribuída*

Júlio Daniel de Carvalho Maia

JOÃO PESSOA-PB

Junho/2018

Universidade Federal da Paraíba
Centro de Informática
Programa de Pós-Graduação em Informática

Quantum Scenery: Um Novo Software de Química Quântica
Semiempírico Para Sistemas de Memória Distribuída

Júlio Daniel de Carvalho Maia

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Informática da Universidade Federal da Paraíba como parte dos requisitos necessários para obtenção do grau de Mestre em Informática.

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Computação Distribuída

Nome do Orientador
Lucídio Dos Anjos Cabral

João Pessoa, Paraíba, Brasil

©Júlio Daniel de Carvalho Maia, 29/06/2018

Catálogo na publicação
Seção de Catálogo e Classificação

M217q Maia, Julio Daniel de Carvalho.

Quantum Scenery: Um Novo Software de Química Quântica
Semiempírico Para Sistemas de Memória Distribuída /
Julio Daniel de Carvalho Maia. - João Pessoa, 2018.
76 f. : il.

Orientação: Lucídio Cabral.

Coorientação: Gerd Rocha.

Dissertação (Mestrado) - UFPB/CI.

1. Modelagem Molecular. 2. Algoritmos Distribuídos. 3.
Química Quântica. 4. Métodos Semiempíricos. I. Cabral,
Lucídio. II. Rocha, Gerd. III. Título.

UFPB/BC



UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA



Ata da Sessão Pública de Defesa de Dissertação de Mestrado de Julio Daniel de Carvalho Maia, candidato ao título de Mestre em Informática na Área de Sistemas de Computação, realizada em 29 de maio de 2018.

1 Aos vinte e nove dias do mês de maio, do ano de dois mil e dezoito, às nove horas, no
2 Centro de Informática da Universidade Federal da Paraíba, em Mangabeira, reuniram-se os
3 membros da Banca Examinadora constituída para julgar o Trabalho Final do Sr. Julio Daniel
4 de Carvalho Maia, vinculado a esta Universidade sob a matrícula nº 2016100038, candidato
5 ao grau de Mestre em Informática, na área de "Sistemas de Computação", na linha de
6 pesquisa "Computação Distribuída", do Programa de Pós-Graduação em Informática, da
7 Universidade Federal da Paraíba. A comissão examinadora foi composta pelos professores:
8 Lucídio dos Anjos Formiga Cabral (PPGI-UFPB), Orientador e Presidente da Banca,
9 Claurton de Albuquerque Siebra (PPGI-UFPB), Examinador interno,
10 Gerd Bruno da Rocha (UFPB), examinador externo ao programa, Rafael de Cássio Bernardi
11 (University of Illinois), Examinador Externo à Instituição. Dando início aos trabalhos, o
12 Presidente da Banca, cumprimentou os presentes, comunicou aos mesmos a finalidade da
13 reunião e passou a palavra ao candidato para que o mesmo fizesse a exposição oral do
14 trabalho de dissertação intitulado "Quantum Scenery: Um Novo Software de Química
15 Quântica Semiempírico Para Sistemas de Memória Distribuída". Concluída a exposição, o
16 candidato foi arguido pela Banca Examinadora que emitiu o seguinte parecer: "**aprovado**".
17 Do ocorrido, eu, Alisson Vasconcelos de Brito, Vice-Coordenador do Programa de Pós-
18 Graduação em Informática, lavrei a presente ata que vai assinada por mim e pelos
19 membros da banca examinadora. João Pessoa, 29 de maio de 2018.

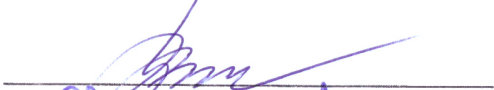

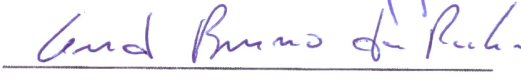
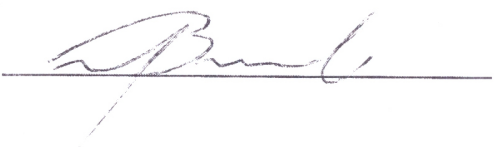

Prof. Dr. Alisson Vasconcelos de Brito

Prof. Dr. Lucídio dos Anjos Formiga Cabral
Orientador (PPGI-UFPB)

Prof. Dr. Claurton de Albuquerque Siebra
Examinador interno (PPGI-UFPB)

Prof. Dr. Gerd Bruno da Rocha
Examinador externo ao programa (UFPB)

Dr. Rafael de Cássio Bernardi
Examinador externo à instituição (University of Illinois)

Resumo

Modelos moleculares quânticos computacionalmente eficientes são extremamente importantes no tratamento de biomoléculas e sistemas químicos complexos. Os esforços de otimização desses modelos foram focados, na sua maioria, nos métodos de primeiros princípios, que apresentam uma complexidade computacional elevada. Os modelos quânticos semiempíricos, que introduzem um maior número de aproximações na sua concepção, e possuem uma complexidade computacional menos elevada, foram deixados de lado. Porém, a utilização de modelos quânticos para a previsão de reações químicas muito longas e de processos biológicos obriga a utilização de modelos aproximados durante o cálculo das propriedades desses sistemas. Nesse trabalho, foi proposto um novo *software* de química quântica computacional semiempírico num paradigma de memória distribuída SPMD para o cálculo de propriedades quânticas, utilizando a biblioteca de álgebra linear ScaLAPACK. Resultados preliminares mostram que o *software* proposto é capaz de reproduzir a energia total semiempírica dos sistemas quânticos de forma acurada. Além da energia, também foi proposto um módulo de otimização da geometria dos sistemas moleculares a partir do vetor gradiente da energia, que é capaz de prever geometrias que representam mínimos de energia para sistemas menores.

Palavras-chave: Modelagem molecular, Algoritmos distribuídos, Química quântica, Métodos semiempíricos

Abstract

Efficient quantum molecular models are of highly importance when treating biomolecules and complex molecular systems. Programming efforts in optimizing less demanding computational models were lacking for a very long time. However, adopting less accurate quantum models for calculating properties is fundamental when modeling chemical and biological processes. This work focuses on introducing a new quantum chemistry software based on a distributed, SPMD paradigm for quantum properties, using the linear algebra library ScaLAPACK. Early results show that this introduced software is accurate when predicting the semiempirical total energy of molecular systems. A module for optimizing molecular geometries is also introduced based on gradients from the total energy. This module is capable of predicting energy minima geometries for smaller systems.

Keywords: Molecular modeling, Distributed algorithms, Quantum chemistry, Semiempirical methods

Agradecimentos

Obrigado a todos que contribuíram de alguma forma com esse trabalho.

Sumário

1	Introdução	1
1.1	Motivação	3
1.2	Objetivos	5
1.2.1	Objetivo Geral	5
1.2.2	Objetivos Específicos	5
1.3	Metodologia	6
1.4	Publicações Relacionadas	8
1.5	Estrutura da Dissertação	8
2	Fundamentação Teórica	9
2.1	O Método Hartree-Fock	9
2.2	A Equação de Hartree-Fock-Roothaan	11
2.3	Métodos Semiempíricos	14
2.3.1	O Formalismo NDDO	15
2.4	Interface de Passagem de mensagens	20
2.5	Biblioteca de Álgebra Linear <i>ScaLAPACK</i>	23
2.6	Considerações Finais	26
3	Trabalhos Relacionados	28
3.1	<i>Softwares</i> de Estrutura Eletrônica Paralelizados	28
3.1.1	Programas semiempíricos	30
3.2	Considerações Finais	31
4	O Software <i>Quantum Scenery</i>	32
4.1	<i>Input</i> das Coordenadas Atômicas	33
4.2	Palpite da Primeira Matriz Densidade	34

4.3	Montagem da matriz de Fock	35
4.3.1	Integrais de repulsão de 2-elétrons	35
4.3.2	Operador de 1-elétron	38
4.3.3	Montagem do Operador de Fock	39
4.3.4	Operações de Álgebra linear e Critérios de Convergência	41
4.4	Gradiente Numérico da Energia e Otimização de Geometria	43
4.5	Considerações Finais	46
5	Resultados	51
5.1	Considerações Finais	59
6	Conclusão	61
	Referências Bibliográficas	68

Lista de Figuras

1.1	Exemplo de um sistema simples e de um sistema complexo do ponto de vista químico. À esquerda: um polipeptídeo formado por um ácido glutâmico e uma alanina; À direita: uma sub-unidade da ricina, a RTA , solvatada com moléculas de água numa caixa cúbica de simulação	2
2.1	Procedimento de Hartree-Fock-Roothaan simplificado em forma de fluxograma, iniciando com o <i>input das coordenadas atômicas</i> e finalizando com o cálculo das propriedades a partir da função de onda associada à geometria	13
2.2	Comparação entre os métodos de estrutura eletrônica. Mais à esquerda, os métodos <i>ab initio</i> representam os métodos mais correlacionados e com maior complexidade computacional. À direita, os métodos semiempíricos são os menos exatos, e com menor complexidade computacional. Os métodos DFT representam um meio termo entre exatidão e complexidade.	14
2.3	Blocos de uma matriz emergente do Metanal no formalismo NDDO. Cores diferentes representam blocos emergentes da interação entre átomos do sistema molecular.	17
2.4	Diferentes possibilidades de topologias de rede. A organização física dos nós computacionais é um fator importante quando é necessário passar mensagens de forma eficiente.	22
4.1	À esquerda, a geometria tridimensional da molécula de água e o seu arquivo de <i>input</i> representante à direita. As coordenadas tridimensionais dos átomos são especificadas utilizando o formato <i>xyz</i>	34
5.1	Correlação entre a energia encontrada a partir do MOPAC2012 e o QS para os sistemas contidos no conjunto de parametrização do RM1	52

5.2	Histograma com o número de estruturas em cada intervalo de RMSD da DL- Find. 148 sistemas obtiveram um valor de $\text{RMSD} \leq 0.1$. Apenas 3 sistemas obtiveram um $\text{RMSD} \geq 0.5$, considerado alto para sistemas pequenos. . . .	53
5.3	Superposição entre duas geometrias do 1,4-Dioxano, com $\text{RMSD} < 0.1$. Di- ferentes geometrias são representadas a partir das cores distintas nas ligações entre os carbonos.	54
5.4	Superposição das duas estruturas finais encontradas do fenil-nitrometano, com $\text{RMSD} \approx 0.73$. De forma análoga à figura 5.3, geometrias diferentes são representadas a partir de cores distintas nas suas ligações entre os carbonos	55
5.5	Superposição das duas geometrias em formato de fio do fenil-nitrometano, com $\text{RMSD} \approx 0.73$. Acima, a estrutura encontrada pelo MOPAC é represen- tada em vermelho, e a estrutura final do QS, em azul.	56
5.6	Tempos total do cálculo da energia para uma caixa esférica de 20\AA para um número de processos MPI $N=\{2,4,8, 16, 32 \text{ e } 48\}$	57

Capítulo 1

Introdução

A área de modelagem molecular representa o conjunto de algoritmos e técnicas computacionais de simulação e manipulação de estruturas moleculares. Em 1998, a Academia Sueca de Ciências reconheceu a importância dos métodos computacionais de modelagem molecular, concedendo o prêmio Nobel da química à Walter Kohn, pelo desenvolvimento da teoria do funcional da densidade (*density-functional theory* ou DFT) e John A. Pople, pelo desenvolvimento de métodos quânticos computacionais [Nobel 1998]. Novamente, em 2013, a Academia evidenciou a importância da modelagem molecular, premiando Martin Karplus, Michael Levitt, e Arieh Warshel, nos seus esforços ao desenvolvimento de modelos moleculares híbridos de mecânica quântica/clássica para sistemas químicos complexos [Nobel 2013]. Nos dias de hoje, é possível tratar sistemas moleculares com muitos átomos empregando arquiteturas computacionais de alta performance, bem como o de algoritmos paralelos que levam em consideração todo o poder computacional dessas plataformas.

A problemática aqui é que os algoritmos convencionais dos métodos de química quântica possuem complexidade elevada no tempo de resolução de suas equações. Do mais simples, que é $O(N^2)$ até o mais complexo que chega a $O(N^7)$, onde N é o número de funções matemáticas usadas na solução do problema e quase sempre N cresce quando o número de átomos da molécula aumenta. Assim, as implementações convencionais tornam esses métodos proibitivos para tratar sistemas com muitos átomos.

Hoje em dia, com o advento da computação de alto desempenho e a disponibilidade de sistemas computacionais de memória distribuída de grande porte, é possível coordenar vários nós computacionais em um mesmo cálculo de química quântica, permitindo o tratamento de

sistemas cada vez maiores e mais complexos.

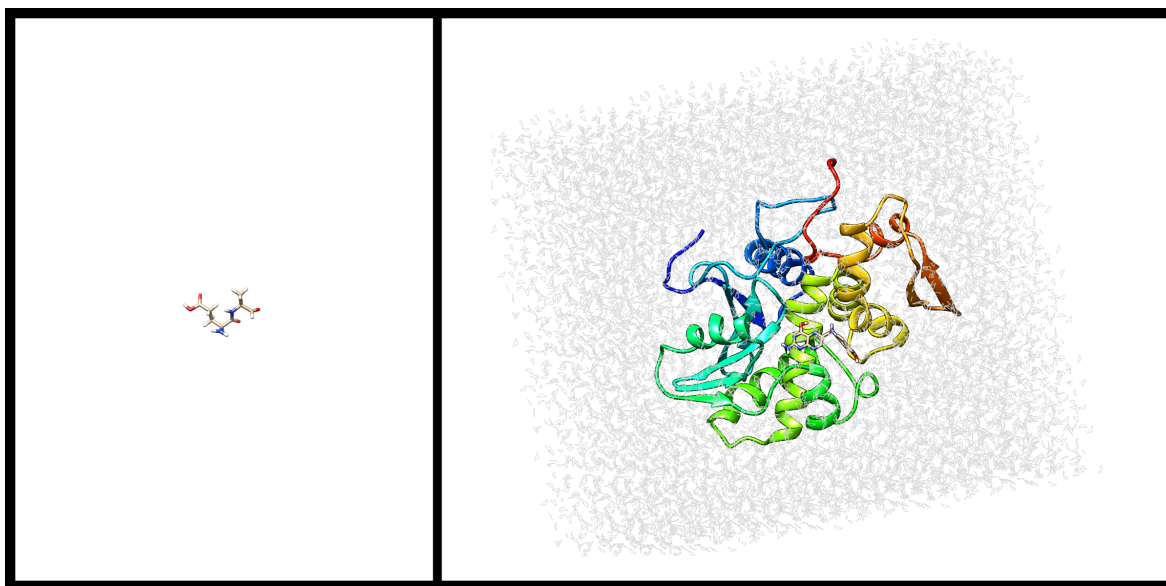


Figura 1.1: Exemplo de um sistema simples e de um sistema complexo do ponto de vista químico. À esquerda: um polipeptídeo formado por um ácido glutâmico e uma alanina; À direita: uma sub-unidade da ricina, a **RTA**, solvatada com moléculas de água numa caixa cúbica de simulação

A partir da figura 1.1, é possível perceber a diferença entre um sistema molecular simples e um complexo. O polipeptídeo representado à esquerda possui apenas 24 átomos, e suas propriedades são calculáveis até com os métodos mais acurados. É impossível utilizar essa abordagem para o sistema molecular da direita, que representa uma proteína solvatada por moléculas de água numa caixa cúbica de simulação, contendo 36698 átomos. O tratamento quântico desse tipo de sistema exige a utilização de métodos menos acurados e mais eficientes do ponto de vista computacional.

Porém, ainda existem dificuldades na utilização de sistemas de memória distribuída. A maioria dos esforços de desenvolvimento e de otimização de softwares para sistemas de memória distribuída foram voltados para os métodos mais exatos. Os métodos semiempíricos, apesar de possuírem uma complexidade computacional mais baixa, foram deixados de lado. O software de química quântica semiempírico mais utilizado pela comunidade, o MOPAC [Stewart 2014], idealizado em 1981, permaneceu por muito tempo um código sequencial,

sem nenhuma otimização. Apenas em meados de 2012, Maia e colaboradores [Maia et al. 2012] introduziram as primeiras otimizações nesse código semiempírico, paralelizando alguns trechos computacionalmente custosos para plataformas com muitos núcleos e unidades de processamento gráfico (GPUs).

Nesse trabalho, será proposto um novo código de química quântica, semiempírico, paralelo, para memória distribuída, e que seja eficiente do ponto de vista computacional.

1.1 Motivação

Como vimos anteriormente, é de fundamental importância que os programas de química quântica sejam ajustados para que a sua utilização em grandes sistemas computacionais seja possível. Os esforços de desenvolvimento, como esperado, focaram na otimização dos métodos mais exatos e de maior complexidade. Os programas computacionais que executam cálculos semiempíricos, apesar de ainda muito utilizados, não sofreram grandes alterações, já que esses métodos são conceitualmente mais simplificados e menos acurados que outros métodos quânticos, tornando possível a sua execução em um tempo computacional menor. Porém, para sistemas com muitos átomos, como a proteína vista na fig. 1.1, muitas vezes uma abordagem quântica não é utilizada para o cálculo das propriedades. A dimensão desses sistemas torna o tratamento quântico inviável, sendo necessário a utilização de várias aproximações durante os cálculos.

Uma abordagem muito usada nesse caso é a *mecânica molecular*, que descreve sistemas moleculares a partir do uso de mecânica clássica, ao invés de mecânica quântica, para que o tratamento de macromoléculas e sistemas muito complexos seja possível. A mecânica molecular desconsidera os elétrons dos sistemas moleculares, e utiliza equações advindas da mecânica clássica para descrever as propriedades e o comportamento de sistemas macromoleculares com o tempo.

Nesses tipos de simulação, são empregadas equações clássicas de movimento para descrever a forma como os biopolímeros se movem [Phillips et al.]:

$$m_{\alpha} \ddot{r}_{\alpha} = - \frac{\partial}{\partial r_{\alpha}} U_{total}(r_1, r_2, \dots, r_n) \quad (1.1)$$

sendo α um átomo pertencente ao conjunto molecular, r_{α} sua posição no espaço, e U_{total} a

energia potencial total, que depende da posição de todos os átomos do sistema molecular.

Em simulações longas, a equação de movimento 1.1 precisa ser reavaliada milhares, ou até milhões, de vezes. Isso torna a eficiência do cálculo da energia potencial total crucial durante procedimentos de dinâmica molecular, onde o sistema será propagado no tempo, com o objetivo de simular a dinâmica dos átomos, considerando as suas naturezas e interações intra-molecular e intermolecular [Karplus e McCammon 2002]. Como estamos lidando com sistemas atômicos, o formalismo correto para se calcular a energia potencial total é o formalismo estabelecido pela mecânica quântica. Porém, a complexidade computacional dos algoritmos envolvidos nos cálculos quânticos torna difícil a simulação de sistemas macromoleculares.

A mecânica molecular introduz o conceito de *campos de força*, definindo as forças que atuam em um átomo como sendo funções dos outros átomos do sistema, e que podem ser calculadas a partir de equações advindas da mecânica clássica. Essa aproximação permite que a avaliação da energia potencial seja realizada de forma muito mais eficiente que a avaliação pelo formalismo quântico, permitindo a simulação de sistemas com milhões de átomos. Porém, por não considerar os elétrons na sua formulação, a mecânica molecular inviabiliza a modelagem de alguns fenômenos físicos e químicos, como os fenômenos fotoquímicos, o cálculo de estados excitados, e interações do sistema molecular com campos magnéticos.

1. Tratar todo o sistema molecular de forma quântica;
2. Utilizar uma abordagem híbrida de QM/MM, onde uma parte do sistema molecular é enxergada pela mecânica quântica e outra pela mecânica molecular;

Como já dito anteriormente, a complexidade computacional dos algoritmos envolvidos impossibilita o tratamento de macromoléculas inteiras com a abordagem quântica. O método QM/MM (*quantum mechanics/molecular mechanics*) foi proposta primeiramente por Warshel e Levitt [Warshel e Levitt 1976], e posteriormente formalizada por Field e colaboradores [Field, Bash e Karplus 1990].

No trabalho introdutório, Field argumenta que a metodologia mais comum para o cálculo das propriedades na dinâmica molecular é a abordagem de mecânica molecular. Porém, como o tratamento clássico dos sistemas impossibilita a modelagem de alguns fenômenos, os autores então propõem uma metodologia híbrida para calcular as propriedades: A maior

parte do sistema é calculada com mecânica molecular, e a parte *reativa* do sistema deve ser calculada com mecânica quântica. Os autores utilizaram aproximações semiempíricas para calcular o trecho de QM do sistema molecular, que por ser mais eficiente que métodos *ab initio*, permite dinâmicas com trajetórias mais longas.

A partir do trabalho de Field e colaboradores, é possível notar a importância dos métodos quânticos menos acurados. Em cálculos de dinâmica molecular com tempos de simulação muito longos, é importante que os métodos que modelam a parte QM em métodos híbridos sejam capazes de calcular as propriedades desejadas sem se tornar um gargalo computacional. Ao otimizar esses tipos de método, o limite superior de dinâmicas quânticas é consequentemente aumentado, permitindo a simulação de fenômenos mais longos e de complexos moleculares maiores.

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo principal desse trabalho é o desenvolvimento de um novo programa de química quântica semiempírico, que se beneficie de sistemas de memória distribuída, e que seja capaz de calcular propriedades de sistemas supramoleculares e biomoleculares, tais como: zeólitas, proteínas, entre outros.

1.2.2 Objetivos Específicos

1. Estudo dos softwares existentes de química quântica na literatura e aprendizagem do procedimento do cálculos quânticos envolvidos;
2. Avaliação de bibliotecas de álgebra linear candidatas a serem utilizadas no código;
3. Aprendizagem das bibliotecas e de programação distribuída utilizando a interface de passagem de mensagens MPI;
4. Implementação do software utilizando MPI e as bibliotecas de álgebra linear escolhidas;

5. Validação das propriedades calculadas pelo software proposto a partir de resultados experimentais e de softwares já existentes;
6. Utilização do software proposto em clusters computacionais e avaliação do desempenho do mesmo com relação ao número de nós computacionais envolvidos no cálculo;

1.3 Metodologia

O fluxo de implementação do trabalho orientar-se-á a partir das equações e procedimentos que definem o cálculo da energia total semiempírica de um sistema molecular. Porém, antes da implementação do software, é necessário primeiramente definir uma biblioteca de álgebra linear capaz de realizar as operações necessárias durante esse cálculo, que são:

- A diagonalização de uma matriz real e simétrica. A partir de uma matriz nesse formato, é necessário que a biblioteca seja capaz de encontrar os autovalores e o seu conjunto de autovetores associados;
- A multiplicação de uma matriz pela sua transposta. Um procedimento de multiplicação de matrizes geral também pode ser utilizado, mas é necessário que a transposta da matriz trabalhada seja calculada de antemão;

Após a escolha da biblioteca de álgebra linear, é possível iniciar a implementação das equações que descrevem a energia total do sistema. O primeiro passo é escolher um formato de *input* das coordenadas atômicas a partir de um arquivo. Dentro desse arquivo também é possível inserir um conjunto de "opções" que controlarão o tipo de cálculo a ser realizado pelo *software*, como: (i) Cálculo da energia total para uma geometria fixa, (ii) processo de otimização de geometria que seja capaz de encontrar a conformação atômica que represente a menor energia total possível, (iii) emprego de métodos de aceleração de convergência, (iv) impressão da estrutura eletrônica do sistema no final do cálculo, etc. Essas opções geralmente são dadas a partir de um conjunto de palavras-chave, e podem ser informadas no cabeçalho do arquivo de *input*.

A partir do *input* das coordenadas atômicas, o processo iterativo do cálculo da energia total pode iniciar. Serão necessárias propostas de algoritmos distribuídos que representam

gargalos computacionais durante esse cálculo, como: A montagem dos operadores envolvidos, as operações de álgebra linear necessárias nesses operadores, e a avaliação da energia total do sistema.

Com um algoritmo distribuído responsável pela avaliação da energia a partir de um conjunto de coordenadas, será possível calcular outras propriedades interessantes do ponto de vista químico, como as cargas atômicas, as energias dos orbitais de fronteira, e o calor de formação final do sistema. Também é possível enxergar a função que calcula a energia total como uma função-custo a ser minimizada, e as coordenadas tridimensionais dos átomos como os parâmetros dessa função que devem ser otimizados. A derivada da energia quântica pode ser encontrada a partir de um procedimento numérico, permitindo otimizar os parâmetros da função-custo utilizando métodos de otimização *quasi-newton*, como o método do gradiente conjugado [Hestenes e Stiefel 1952], o método de Broyden-Fletcher-Goldfarb-Shanno (Ou BFGS) [Li e Fukushima 2001], entre outros.

Finalmente, o trabalho será realizado a partir das seguintes atividades:

1. Escolha do formato de matrizes trabalhado;
2. Revisão bibliográfica e escolha da biblioteca de álgebra linear a ser utilizada durante o cálculo da energia do sistema molecular;
3. Implementação da leitura das coordenadas atômicas via arquivo de *input*;
4. Implementação das seguintes operações em memória distribuída:
 - (a) Palpite dos primeiros coeficientes dos orbitais moleculares e montagem da matriz densidade;
 - (b) Cálculo do operador de 1-elétron a partir das integrais de *overlap*;
 - (c) Cálculo das integrais repulsão de 2-elétrons;
 - (d) Montagem do operador de Fock a partir do operador de 1-elétron e das integrais de 2-elétrons;
 - (e) Avaliação da energia do sistema semiempírico a partir do operador de Fock;
5. Cálculo dos gradientes numéricos a partir das coordenadas dos átomos;

6. Implementação de algoritmos de otimização *Quasi-Newton* com a informação da derivada para a otimização da geometria a partir da energia;
7. Publicação dos resultados encontrados;

1.4 Publicações Relacionadas

1.5 Estrutura da Dissertação

O presente trabalho será organizado da seguinte forma:

Capítulo 2: Nesse capítulo será introduzida a fundamentação teórica presente no trabalho: Um resumo dos modelos físicos baseados na mecânica quântica envolvidos no cálculo e os aspectos computacionais relativa ao processo de escrita dos algoritmos distribuídos;

Capítulo 3: Serão apresentados trabalhos que já abordam o uso de computação de alto desempenho e arquiteturas distribuídas na resolução de problemas da química quântica;

Capítulo 4: Descreve o programa *Quantum Scenery* em detalhes: Como cada parte foi codificada e como os algoritmos distribuídos funcionam;

Capítulo 5: Descreve os resultados obtidos até o presente momento;

Capítulo 6: Contém as considerações parciais acerca dos resultados obtidos até o presente momento, e quais as perspectivas dos objetivos alcançados;

Capítulo 2

Fundamentação Teórica

2.1 O Método Hartree-Fock

O método de Hartree-Fock [Seaton 1977] é um método computacional para se encontrar uma função de onda que descreve um sistema molecular em questão, e é a base para a maioria das técnicas em química quântica computacional. A equação de Hartree-Fock é um problema de autovalores e autovetores, e a sua solução provê uma forma de encontrar a energia de um sistema com n elétrons. A equação de Hartree-Fock é da forma:

$$f |\chi_\alpha\rangle = \varepsilon_\alpha |\chi_\alpha\rangle \quad (2.1)$$

onde f é o operador de Fock, χ_α é um spin-orbital (produto de duas funções, uma que depende das coordenadas do elétron e a outra que depende da sua coordenada de spin) e ε_α é a energia desse spin-orbital. Nessa equação, os spin-orbitais são os autovetores do operador de Fock e as suas energias associadas, os autovalores. Cada spin-orbital é uma função mono-eletrônica centrada na posição de cada núcleo atômico e que descreve a posição e o spin de um elétron ao redor deste núcleo. Essas funções atômicas mono-eletrônicas podem ser combinadas linearmente de forma a construir uma função mono-eletrônica molecular, são os chamados orbitais moleculares φ_i , e esse procedimento de construção é conhecido como (MO-LCAO, do inglês, Molecular Orbital as a Linear Combination of Atomic Orbitals). Os orbitais moleculares, por sua vez, são usados na montagem da função de onda completa da molécula via determinante de Slater, pois é um requisito de todo sistema de partículas

fermiônicas (aquele composto de partículas que tem spin semi-inteiro) ser descrito por uma função de onda antissimétrica. O determinante segue:

$$\Psi(x_1, x_2, \dots, x_n) = \frac{1}{\sqrt{N!}} \begin{bmatrix} \varphi_1(x_1) & \varphi_2(x_1) & \dots & \varphi_N(x_1) \\ \varphi_1(x_2) & \varphi_2(x_2) & \dots & \varphi_N(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1(x_N) & \varphi_2(x_N) & \dots & \varphi_N(x_N) \end{bmatrix} \quad (2.2)$$

O princípio da antissimetria estabelece que a função de onda deve ser antissimétrica com relação à permutação de quaisquer dois elétrons, visto que os elétrons são indistinguíveis uns dos outros. Qualquer permutação (que aqui pode ser feita trocando duas colunas) acarreta apenas numa mudança no sinal do determinante. A energia de uma função de onda qualquer pode ser calculada através da integral mostrada na equação a seguir.

$$\langle \Psi | f | \Psi \rangle = E^1 \quad (2.3)$$

Dessa forma a energia de um sistema molecular passa a depender de dois fatores: (1) o conjunto de funções matemáticas usadas para representar um spin-orbital atômico e (2) os coeficientes da expansão linear na composição de cada orbital molecular. Em se tratando do primeiro fator existem funções matemáticas que são mais adequadas ao problema molecular. Essas funções matemáticas são as STOs (Slater-type Orbitals) e GTOs (Gaussian-type Orbitals). Em se tratando do segundo fator precisamos encontrar quais coeficientes lineares tornam a energia da equação 2.3 um mínimo quando temos um conjunto de spins-orbitais pré-definidos. Isso pode ser conseguido tendo como base o princípio variacional.

O princípio variacional afirma que, se conseguirmos uma função de onda normalizada, o valor esperado do operador de Fock é um limite inferior para a energia exata do estado fundamental. A equação 2.4 explicita o princípio variacional.

$$\langle \Psi_{exata} | f | \Psi_{exata} \rangle = E_{exata} \quad (2.4)$$

Se possuírmos uma função de onda exata (Ψ_{exata}), o valor esperado do operador de Fock corresponde à energia exata do sistema molecular. O objetivo do método Hartree-Fock é justamente combinar o princípio variacional com ideia que a função de onda que descreve

¹Valor esperado da energia de um estado quântico utilizando a notação bra-ket proposta por Paul Dirac.

o sistema molecular é um determinante de Slater. O objetivo é encontrar a função de onda mais próxima da exata possível, e isso é feito através de uma minimização do funcional da energia com respeito aos coeficientes da expansão dos spin-orbitais atômicas através de um método análogo aos multiplicadores de Lagrange, o método iterativo do *self-consistent field* (SCF), discutido adiante.

2.2 A Equação de Hartree-Fock-Roothaan

A formulação matricial do método HF leva em consideração a expansão dos orbitais moleculares em termos de um conjunto de funções-base conhecidas, representadas aqui por $g_v(r)$. Os orbitais moleculares são escritos da forma:

$$\phi_p(r) = \sum_{v=1}^k C_{vp} g_v(r) \quad (2.5)$$

onde k é o número de orbitais atômicos, e os C_{vp} são os coeficientes que serão determinados durante o método. Expressar a função de onda através dessa forma leva à uma equação de autovalores generalizada, a equação de Hartree-Fock-Roothaan [Janssen e Nielsen 2008], representada em 2.6:

$$FC = SC\varepsilon \quad (2.6)$$

A equação 2.6 recebe o nome de equação de Hartree-Fock-Roothaan, e é uma equação de autovalores e autovetores, onde F é a matriz de Fock, C são os coeficientes dos orbitais moleculares, S são as integrais de *overlap* e ε representa as energias dos orbitais moleculares.

O conjunto de funções de base é conhecido. Então, o problema de encontrar as energias dos orbitais moleculares limita-se a encontrar os coeficientes que formarão a matriz C . Esses coeficientes podem ser encontrados ao diagonalizar a matriz F , já que a equação 2.6 é um problema de autovalores e autovetores. É necessário então definir uma expressão para a matriz F em termos de integrais de 1 e 2 elétrons que envolva o conjunto de funções de base que nós conhecemos. No método HFR, o operador de Fock pode ser definido a partir da

equação 2.7:

$$F_{\mu\nu} = H_{\mu\nu}^{core} + \sum_{\lambda\sigma} P_{\lambda\sigma} \left[(\mu\nu|\lambda\sigma) - \frac{1}{2}(\mu\lambda|\nu\sigma) \right] \quad (2.7)$$

A matriz P pode ser calculada a partir da equação 2.8:

$$P_{\mu\nu} = 2 \sum_{\alpha}^{\frac{N}{2}} C_{\mu\alpha} C_{\nu\alpha}^* \quad (2.8)$$

Podemos notar que a montagem da matriz de Fock depende dos coeficientes C dos orbitais moleculares. Esses coeficientes, segundo a equação 2.6, são obtidos diagonalizando a própria matriz de Fock, o que leva a uma dependência circular. Esse problema é resolvido iterativamente, através de um procedimento chamado self-consistent field (SCF). O SCF é um procedimento iterativo, e resolve as equações de Roothaan da seguinte forma:

1. Especifica-se as características do sistema: posições dos núcleos, números atômicos e o número de elétrons;
2. Escolhe-se um conjunto de funções de base do sistema (spin-orbitais atômicos);
3. Calcula-se as matrizes S e H, e o conjunto de integrais de um-elétron e dois-elétrons;
4. Realiza-se um palpite para uma primeira matriz densidade que servirá como ponto de partida para o cálculo;
5. Calcula-se a matriz F a partir da matriz P, S e H;
6. Diagonaliza-se a matriz F para obter novos C e energias associadas aos orbitais ε
7. A partir de C, monta-se uma nova matriz P. A matriz P for semelhante a matriz P dentro de um certo parâmetro estabelecido, o cálculo converge. Se o procedimento não convergir, volta-se ao procedimento 5;
8. Se o procedimento convergir, podemos calcular a energia eletrônica associada ao sistema através da equação 2.3, já que conseguimos chegar numa função de onda "boa o suficiente";

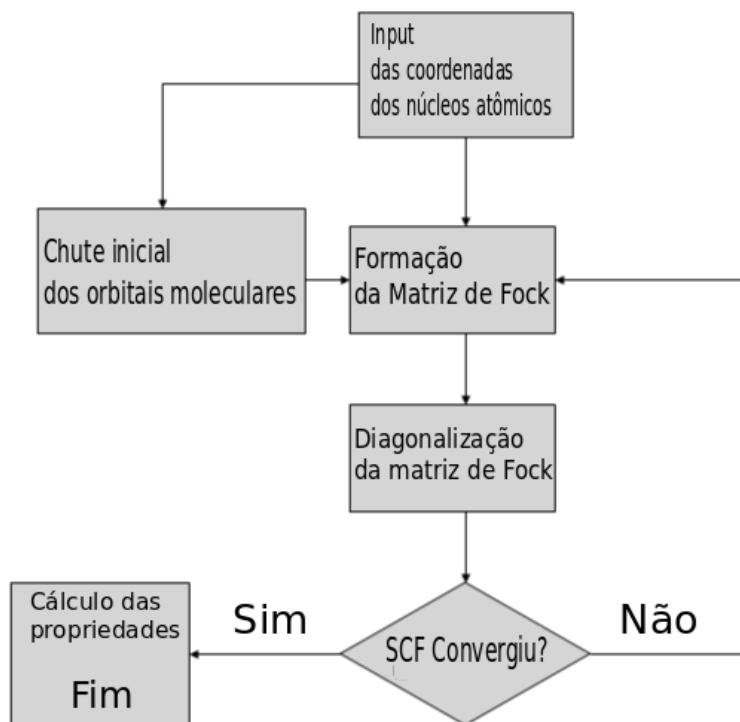


Figura 2.1: Procedimento de Hartree-Fock-Roothaan simplificado em forma de fluxograma, iniciando com o *input das coordenadas atômicas* e finalizando com o cálculo das propriedades a partir da função de onda associada à geometria

A figura 2.1 ilustra uma simplificação do procedimento SCF. É possível, a partir desse processo, encontrar a energia total para o sistema. É importante notar que os procedimentos de montagem da matriz de densidade e diagonalização da matriz de Fock tornam-se gargalos durante o SCF, e são bons candidatos à paralelização. A montagem da matriz P é nada mais que uma multiplicação da matriz dos coeficientes pela matriz dos coeficientes, de forma transposta, e o procedimento de diagonalização de um operador simétrico está presente na maioria das distribuições da LAPACK. Isso significa que os gargalos computacionais podem ser facilmente paralelizados se uma biblioteca de álgebra linear paralela for adotada durante o procedimento do SCF.

2.3 Métodos Semiempíricos

Atualmente, existem 3 categorias de métodos que representam níveis de abstração diferentes para cálculos de estrutura eletrônica. Métodos com maior acurácia são conhecidos como *ab initio*, ou de primeiros princípios, e são capazes de reproduzir características calculadas por meios experimentais de forma consistente. Existem também os métodos baseados em DFT (*Density functional theory*), que possuem uma complexidade computacional mais baixa que os de primeiros princípios, ao custo de menos exatidão nas características computadas. Hoje em dia, métodos DFT são os métodos mais utilizados na química computacional [Thiel 2014], devido ao seu bom balanceamento entre nível de exatidão e eficiência, permitindo a simulação de sistemas mais complexos que os tratados por métodos de primeiros princípios, e mantendo um nível de exatidão aceitável.

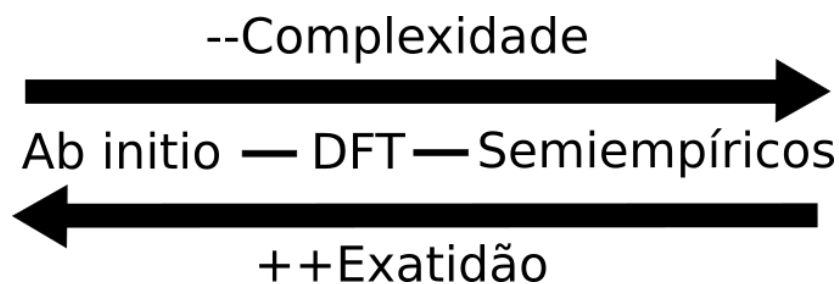


Figura 2.2: Comparação entre os métodos de estrutura eletrônica. Mais à esquerda, os métodos *ab initio* representam os métodos mais correlacionados e com maior complexidade computacional. À direita, os métodos semiempíricos são os menos exatos, e com menor complexidade computacional. Os métodos DFT representam um meio termo entre exatidão e complexidade.

Por último, existem os métodos chamados semiempíricos, que introduzem mais simplificações nos cálculos de estrutura eletrônica, possuindo menor complexidade computacional. Esses métodos se baseiam no formalismo dos métodos de primeiros princípios, mas apresentam aproximações nas equações que montam os operadores e parametrizações numéricas que os tornam muito menos custosos computacionalmente. Os erros introduzidos na função de onda a partir das aproximações são minimizados com a introdução de parâmetros numéricos "calibrados" com resultados experimentais.

Os métodos semiempíricos levam em consideração aproximações durante o cálculo da

energia total. Primeiramente, esses métodos tratam apenas os elétrons da camada de valência de cada átomo durante a montagem do operador de Fock, representando os mesmos através de uma base mínima. Elétrons de próximos ao núcleo são representados apenas por uma carga nuclear reduzida. Uma das aproximações introduzida no formalismo semiempíricos é a eliminação de integrais de 3 e 4 centros, que aumentam muito a complexidade computacional do procedimento se levadas em consideração.

Conceitualmente, os métodos semiempíricos são bem menos custosos computacionalmente que os métodos *ab-initio*. A eliminação de integrais de 3 e 4 centros reduz a complexidade computacional do cálculo de repulsão de 2-elétrons de $O(N^4)$ para $O(N^2)$ durante a montagem da matriz de Fock, que é o gargalo computacional dos cálculos *ab-initio*. O gargalo agora passa a ser as operações de álgebra linear durante o cálculo, como a diagonalização do operador de Fock, e a montagem da matriz densidade a partir dos coeficientes dos orbitais moleculares.

2.3.1 O Formalismo NDDO

Existem alguns formalismos adotados nos métodos semiempíricos com diferentes níveis de aproximação. O formalismo mais adotado atualmente é o *Neglect of Diatomic Differential Overlap* (NDDO) [Rocha G.B.; Simas 2007]. A maioria dos métodos semiempíricos atualmente utilizam aproximações do formalismo NDDO, e geralmente modificam o seu conjunto de parâmetros numéricos para alcançar resultados mais acurados em propriedades quânticas desejadas.

As aproximações empregadas no NDDO são:

- Os elétrons próximos ao núcleo são ignorados, e representados por uma carga média. Os elétrons da valência são representados por um conjunto mínimo de orbitais de Slater fixo;
- A matriz de *overlap* envolvida na equação 2.6 é uma matriz identidade, e os elementos da mesma podem ser enxergados como um delta de Kronecker [Dewar e Thiel 1977];
- As integrais de repulsão de 1-elétron são representadas através de parâmetros numéricos, e apenas as de 2 centros são calculadas a partir da geometria molecular. O formalismo das equações que regem a montagem dos operadores exige apenas a avaliação

de integrais $(\mu\nu|\lambda\sigma)$ onde os pares de orbitais estão centrados nos mesmos átomos $(\mu\nu \in Atom_1)$ e $(\lambda\sigma \in Atom_2)$;

As matrizes envolvidas nos cálculos semiempíricos possuem dimensões definidas a partir do número de orbitais atômicos do sistema. Desse modo, cada átomo possui um certo número de orbitais, e cada orbital representa uma linha e uma coluna das matrizes envolvidas nos cálculos. Podemos então enxergar a matriz como sendo *blocada*, e os blocos que representam interações entre os átomos. Um elemento que possui as linhas e colunas como orbitais de um mesmo átomo define um elemento de 1-Centro. Já elementos que representam linhas e colunas de orbitais de átomos distintos definem blocos de 2-Centros. A blocagem de uma matriz nos métodos semiempíricos pode ser verificada a partir da figura 2.3.

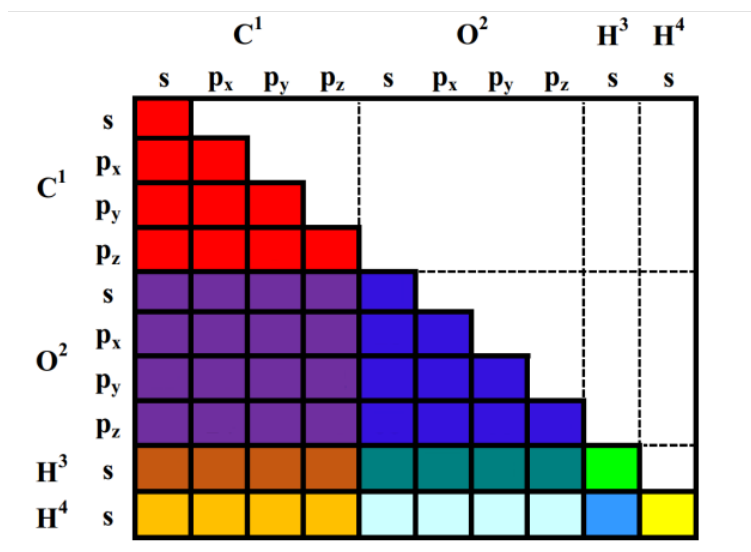


Figura 2.3: Blocos de uma matriz emergente do Metanal no formalismo NDDO. Cores diferentes representam blocos emergentes da interação entre átomos do sistema molecular.

A figura 2.3 representa o metanal (CH_2O). O carbono e o oxigênio contém 4 orbitais (s , p_x , p_y e p_z), e o hidrogênio só contém um orbital s . Os elementos pertencentes aos blocos das interações entre o (C^1-C^1), (O^2-O^2) (H^3-H^3), e (H^4-H^4) são de 1-centro. Esses blocos são simétricos, então apenas o triângulo inferior precisa ser armazenado. Os blocos (O^2-C^1), (H^3-C^1), (H^3-O^2), (H^4-C^1), (H^4-O^2), e (H^4-H^3) são blocos de 2-centros, e não são simétricos.

O termo de 1-elétron (H_{core}) pode ser entendido como o operador que representa os termos da energia cinética do elétron, e a energia potencial de atração entre o elétron e o núcleo. Esse operador pode ser representado como uma matriz, e possuem expressões que regem os seus valores, discutidas a seguir.

Nos métodos baseados em NDDO, e consequentemente nos métodos empregados nesse trabalho [Dewar e Thiel 1977] [Rocha et al. 2006], os termos na diagonal da matriz que representa o operador de 1-elétron representam a interação entre os elétrons e os núcleos de cada átomo, e podem ser definidos a partir da seguinte equação:

$$H_{\mu\mu}^{core} = U_{\mu} - \sum_{B \neq A} Z_B (\mu\mu | s_b s_b) \quad (2.9)$$

onde A e B são átomos distintos, μ é um orbital centrado no átomo A, e $(\mu\mu|s_b s_b)$ é o valor da atração entre o orbital μ e o núcleo do átomo B, representado como dois orbitais s fictícios na operação da integral.

Para elementos fora da diagonal, existem duas possíveis equações que regem os valores do operador de 1 elétron, a equação 2.10:

$$H_{\mu\nu}^{core} = \sum_{B \neq A} Z_B (\mu\nu|s_b s_b) \quad (2.10)$$

onde μ e ν são orbitais pertencem ao átomo A, e os outros termos são análogos à eq. 2.9, rege os elementos que estão fora da diagonal, mas representam blocos de 1-Centro. Já a equação 2.11:

$$H_{\mu\nu}^{core} = \frac{1}{2}(\beta_\mu + \beta_\nu) S_{\mu\nu} \quad (2.11)$$

rege os elementos que estão fora da diagonal e representam blocos de 2-centros. Nessa equação, μ e ν são orbitais de átomos distintos, os termos β_μ e β_ν são parâmetros numéricos, e um termo da matriz de overlap S, montada a partir das integrais de *overlap* entre os dois orbitais. A partir das equações 2.9, 2.10, e 2.11, é possível construir a matriz de 1-elétron apropriada.

O operador de Fock pode ser construído de forma análoga. Elementos da diagonal podem ser calculados a partir da equação a seguir:

$$F_{\mu\mu} = H_{\mu\mu}^{core} + \sum_{\nu \in A} P_{\nu\nu} [(\mu\mu|\nu\nu) - \frac{1}{2}(\mu\nu|\mu\nu)] + \sum_B \sum_{\lambda \in B} \sum_{\sigma \in B} P_{\lambda\sigma} (\mu\mu|\lambda\sigma) \quad (2.12)$$

Para elementos de 1-centro fora da diagonal, o seu valor pode ser calculado de forma análoga à da equação 2.12:

$$F_{\mu\nu} = H_{\mu\nu}^{core} + P_{\mu\nu} [\frac{3}{2}(\mu\nu|\mu\nu) - \frac{1}{2}(\mu\mu|\nu\nu)] + \sum_B \sum_{\lambda \in B} \sum_{\sigma \in B} P_{\lambda\sigma} (\mu\nu|\lambda\sigma) \quad (2.13)$$

Elementos de 2-centros são calculador a partir da equação a seguir:

$$F_{\mu\nu} = H_{\mu\nu}^{core} - \frac{1}{2} \sum_{\lambda \in A} \sum_{\sigma \in B} P_{\lambda\sigma} (\mu\lambda|\nu\sigma) \quad (2.14)$$

A energia eletrônica do sistema envolvido a partir do operador de 1-elétron e do operador de Fock nos métodos NDDO da seguinte forma:

$$E_{elec} = \frac{1}{2} \sum_{\mu=1}^{n_{orbs}} \sum_{\nu=1}^{\mu} P_{\mu\nu} (H_{\mu\nu}^{core} + F_{\mu\nu}) \quad (2.15)$$

e, a partir da energia eletrônica, a energia total de um sistema também pode ser facilmente inferida:

$$E_{total} = E_{elec} + \sum_{A=2}^{n_{atoms}} \sum_{B=1}^{A-1} E_{A,B}^{nuc} \quad (2.16)$$

O termo $E_{A,B}^{nuc}$ na equação 2.16 é o termo que representa a energia de repulsão entre os núcleos atômicos. O seu valor será discutido mais adiante.

A maioria dos modelos semiempíricos utilizam as aproximações do formalismo NDDO [Thiel 2014]. Em meados de 1976, Thiel e Dewar apresentaram uma modificação na forma como o NDDO calcula as integrais de repulsão de 2-elétrons, bem como a introdução de parâmetros numéricos otimizados durante o cálculo dos operadores e das propriedades, dando origem ao método *Modified Neglect of Diatomic Overlap* (MNDO). Nos métodos baseados em MNDO, a energia de repulsão entre os núcleos é calculada a partir dos parâmetros numéricos inseridos, bem como de fórmulas analíticas [Dewar e Thiel 1977], e o seu valor pode ser especificado a partir da equação a seguir:

$$E_{A,B}^{nuc} = (Z_A Z_B (s_A s_A | s_B s_B) + (Z_A Z_B (s_A s_A | s_B s_B) [e^{-\alpha_A R_{AB}} + e^{-\alpha_B R_{AB}}])) \quad (2.17)$$

onde α é um dos parâmetros numéricos ajustáveis no método, e R_{AB} é a distância entre os núcleos atômicos. A integral de repulsão $(s_A s_A | s_B s_B)$ concede uma esfericidade ao potencial de repulsão nuclear, que é centro-esférico.

A partir da energia eletrônica, obtida através do operador de Fock, do operador de 1-elétron e da matriz densidade, e da energia de repulsão nuclear, obtida a partir dos parâmetros numéricos inseridos no cálculo, é possível calcular a energia total do sistema utilizando a equação 2.16.

Os parâmetros numéricos introduzidos no formalismo NDDO podem variar com o intuito de otimizar alguma propriedade calculada. Dewar e colaboradores propuseram a adoção de

um método de otimização não-linear de mínimos quadrados para definição desses parâmetros. Existem vários métodos semiempíricos que adotam as aproximações introduzidas no MNDO e introduzem um novo conjunto de parâmetros, empregando diferentes métodos de otimização, com o objetivo de minimizar os erros introduzidos nas simplificações dos cálculos. Em 2006, Rocha e colaboradores [Rocha et al. 2006] propuseram o *Recife Model 1*, uma reparametrização do método AM1 (*Austin Model 1*), que é baseado no MNDO, para o conjunto de átomos **H, C, N, O, P, S, F, Cl, Br, e I**. O RM1 foi proposto com o intuito de reduzir o erro gerado a partir das aproximações semiempíricas para sistemas que são compostos pelos conjunto de átomos parametrizados originalmente pelo RM1.

2.4 Interface de Passagem de mensagens

A taxonomia de Flynn classifica sistemas computacionais pela quantidade de fluxos de instrução e de dados. As classificações possíveis são:

- **SISD** (*Single Instruction Single Data*): Representa a máquina clássica de von Neumann, sem nenhum tipo de paralelismo. Possui apenas um fluxo de instrução, e trabalha apenas com um fluxo de dados;
- **MISD** (*Multiple Instruction Single Data*): Múltiplos fluxos de instrução com apenas um fluxo de dados, compartilhado. São geralmente utilizados em sistemas com tolerância a falha;
- **SIMD** (*Single Instruction Multiple Data*): Sistemas em que há apenas um fluxo de instruções, e múltiplos fluxos de dados. Um grande exemplo de arquitetura SIMD são as unidades de processamento gráfico (GPUs), em que um conjunto de núcleos computacionais que compartilham a mesma unidade de controle, recebe a mesma instrução. A instrução é a mesma, porém, os dados nos registradores de cada núcleo são particulares daquele núcleo;
- **MIMD** (*Multiple Instruction Multiple Data*): Sistemas com múltiplos fluxos de instrução e de dados. Sistemas MIMD possuem núcleos de processamento independentes uns dos outros: Cada um possui a sua unidade de controle, a sua unidade lógica e aritmética, e trabalham com instruções e dados autônomos; Os supercomputadores são o

maior exemplo dessa arquitetura, onde cada nó possui seu próprio núcleo de processamento e região de memória, são interligados por uma rede, e a colaboração entre eles é dada a partir de um processo de troca de mensagens;

Nos dias de hoje, devido à demanda computacional por sistemas cada vez mais robustos, a tendência é que os sistemas MIMD sejam mais utilizados na resolução de tarefas de alto desempenho. É muito mais fácil aumentar a capacidade computacional de sistemas MIMD, já que sistemas paralelos dessa arquitetura são construídos de forma heterogênea, com nós computacionais com arquiteturas, processadores, e quantidades de recursos disponíveis diferentes.

Sistemas MIMD são assíncronos, seus nós computacionais são independentes entre si, e os processadores desses nós não compartilham um relógio global. Com isso em mente, a dificuldade no desenvolvimento de *softwares* que se aproveitem desses recursos eficientemente para qualquer topologia de rede presente aumenta. Também é importante notar que a forma como os nós computacionais, ou processadores, são organizados na rede local, influencia no desempenho computacional do sistema. À forma como os nós computacionais são organizados e interconectados, e como os mesmos se comunicam numa rede local, damos o nome de **topologia de rede**.

Topologias de rede podem ser regulares ou irregulares. As topologias regulares geralmente assumem formas de grafo conhecida. Porém, as topologias de rede para sistemas de high-performance computing geralmente costumam ser altamente regulares, ao contrário de redes para data centers [Minkenberg 2013], facilitando a escrita de algoritmos de comunicação entre os nós de forma eficiente.

Existem diversas topologias regulares utilizadas na montagem de arquiteturas distribuídas nos dias de hoje. Na fig. 2.4, é possível notar que, devido à variedade de topologias disponíveis, é possível passar mensagens de várias formas diferentes. Nos piores casos, será necessário que algumas mensagens passem por muitos nós intermediários antes de chegar ao destinatário.

Num paradigma de programação distribuída, é difícil adequar um programa a uma dada topologia, em tempo de execução. Existem inúmeras arquiteturas paralelas disponíveis, com diferentes topologias de rede. Precisamos, então de uma abstração robusta o suficiente para lidar com os inúmeros tipos de topologias diferentes.

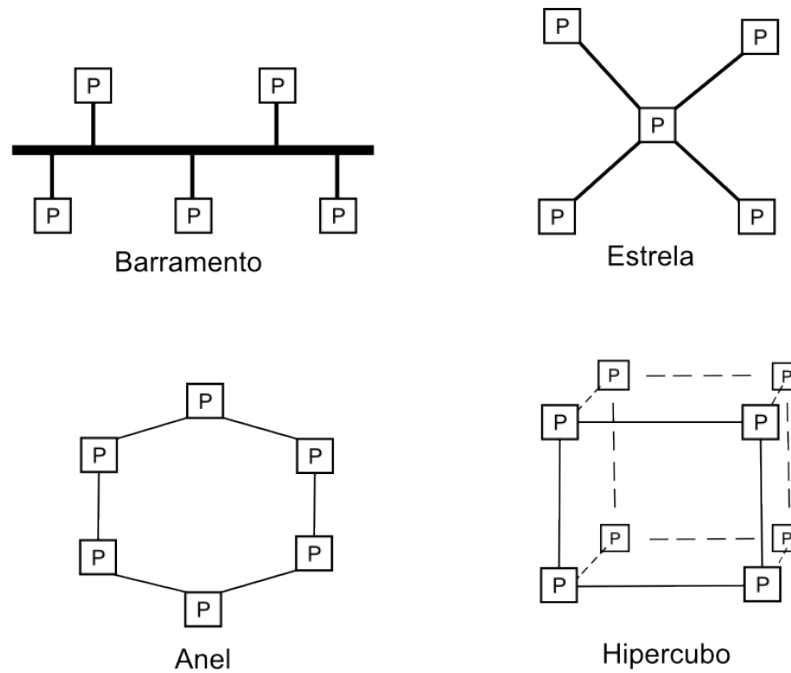


Figura 2.4: Diferentes possibilidades de topologias de rede. A organização física dos nós computacionais é um fator importante quando é necessário passar mensagens de forma eficiente.

A abstração mais utilizada em programação de algoritmos distribuídos é a de passagem de mensagens [Pacheco 1996]. Num procedimento de passagem de mensagens básico, os processos coordenam as suas atividades passando e recebendo mensagens entre si, e essa troca de mensagens definirá a cooperação entre os mesmos.

A ideia de processo é primordial durante a escrita de algoritmos distribuídos. Um processo é um bloco básico em qualquer paradigma de programação paralela. Segundo Pacheco, um processo nada mais é do que uma instância de um programa que executa de forma autônoma em um processador físico [Pacheco 1996]. Cada processo funciona como um programa autônomo, com uma região de memória e ponteiro de instrução, e executa de forma independente dos outros processos envolvidos do programa. A cooperação entre os processos deve ser implementada de forma explícita pelo desenvolvedor. Se o paradigma utilizado for o de passagem de mensagens, as mensagens entre os processos são utilizadas para coordenar a cooperação entre os processos envolvidos numa execução distribuída.

É um desafio de programação otimizar o processo de troca de mensagens. A melhor forma de trocar mensagens entre nós depende de rede presente. Para facilitar a programação,

foi desenvolvida uma Interface de Passagem de Mensagens (MPI). MPI é basicamente uma biblioteca de definições e funções que podem ser invocadas a partir do código-fonte [Pacheco 1996]. Como ela abstrai toda a camada de rede da implementação distribuída, não é necessário Assim, os arquitetos da topologia utilizada em um supercomputador podem alterar a sua distribuição MPI para otimizar as funções de comunicação entre os processos, reduzindo a latência dos programas nessa arquitetura.

A implementação de MPI leva em consideração que os processos são estaticamente alocados. O conjunto de processos de um programa é definido no início da execução do mesmo, e se mantém estático até o final. A cada processo dentro do conjunto é atribuído um número inteiro como índice (Ou *rank*) no início da alocação, que o irá identificar dentre os outros. Nesse modelo, cada processo executa uma cópia do mesmo programa, e a cooperação entre os mesmos se dá a partir do *rank* de cada processo. Nos sistemas MIMD, essa abordagem de programação é comumente conhecida como SPMD (*Single Program Multiple Data*). O controle de fluxo no código, baseado nos *ranks* de cada processo, definirá a cooperação e a divisão do trabalho entre os processos envolvidos.

2.5 Biblioteca de Álgebra Linear ScaLAPACK

O procedimento de Hartree-Fock pode ser visto como sendo um método variacional, envolvendo operações de álgebra linear na sua resolução. Particularmente, operações de álgebra linear fazem parte da operação de diagonalização do operador de Fock e formação da matriz densidade a partir dos coeficientes dos orbitais moleculares. Na equação 2.6, é perceptível que os coeficientes dos orbitais moleculares são os autovetores do operador de Fock.

Existem várias bibliotecas de álgebra linear que implementam as funções necessárias para o cálculo SCF. A montagem da matriz densidade é meramente uma multiplicação de matrizes, uma operação presente na biblioteca BLAS (*Basic Linear Algebra Subprograms*), e que possui várias implementações paralelas [Barrachina et al. 2008] [Goto e Van De Geijn 2008]. O procedimento de diagonalização de matrizes está presente na LAPACK (*Linear Algebra PACKage*) [Angerson et al. 1990], que também possui várias distribuições paralelas [Tomov et al. 2010] [Whaley e Dongarra 1998].

No entanto, a maioria das implementações paralelas das bibliotecas a seguir são apropri-

adas para ambientes de memória compartilhada, onde os dados estão acessíveis para todos os processos envolvidos no cálculo. Como o objetivo desse trabalho é a implementação de um cálculo de estrutura eletrônica semiempírico em memória distribuída, a utilização dessas bibliotecas fica inviável.

Implementações de funções de álgebra linear em memória distribuída já estão disponíveis. A biblioteca mais expressiva nesse caso é a ScaLAPACK (*Scalable Linear Algebra PACKage*) [Choi et al. 1992], de código aberto, escrita em Fortran 90, e que utiliza a biblioteca BLACS (*Basic Linear Algebra Communication Subprograms*) [Anderson et al. 1991] na camada de comunicação. A BLACS encapsula as rotinas de comunicação necessárias em procedimentos de álgebra linear distribuídos, e existem versões disponíveis para várias interfaces de passagem de mensagens existentes. Consequentemente, softwares que utilizam a BLACS são utilizáveis em praticamente todas as plataformas paralelas. Até o presente momento, a ScaLAPACK está disponível apenas em Fortran 90. Porém, C/C++ e Fortran 90 são linguagens interoperáveis, o que facilita a integração da biblioteca em códigos escritos em outras linguagens com abstrações diferentes.

Ao trabalhar com a ScaLAPACK, é necessário definir um *grid* lógico de processos, utilizado para distribuir as matrizes envolvidas nos procedimentos de álgebra linear. O *grid* definido da ScaLAPACK organizará os processos pertencentes a um dado comunicador MPI em uma estrutura bidimensional de matriz de processos, de forma *column-major* (Ou *row-major*, se especificado). As dimensões do *grid* são especificadas pelo usuário. O comunicador MPI, então, é convertido a *grid* da seguinte forma:

$$Comm = \{P_1, P_2, P_3, P_4, P_5, P_6\} \quad (2.18)$$

$$Grid = \begin{bmatrix} P_1 & P_3 & P_5 \\ P_2 & P_4 & P_6 \end{bmatrix} \quad (2.19)$$

Onde o *grid* possui dimensões (2,3) e é organizado na forma *column-major*. O mesmo *grid* poderia também ser organizados de forma *row-major*. Nesse caso, a organização ficaria:

$$Grid = \begin{bmatrix} P_1 & P_2 & P_3 \\ P_4 & P_5 & P_6 \end{bmatrix} \quad (2.20)$$

Para operações de álgebra linear em matrizes densas, a ScaLAPACK utiliza a distribuição bidimensional *block-cyclic*. Essas operações são em sua maioria altamente paralelas, exigindo uma estrutura de dados que permita um bom balanceamento de carga, comunicação, e escalabilidade. Essa distribuição funciona delegando responsabilidade de blocos à processos de *grids* bidimensionais. Dado um conjunto de processos:

$$Comm_{ScaLAPACK} = \{P_1, P_2, P_3, P_4\} \quad (2.21)$$

É possível criar um *grid* bidimensional, 2x2, a partir desse comunicador, com um ordenamento padrão *row-major*:

$$Grid_{ScaLAPACK} = \begin{Bmatrix} P_1 & P_2 \\ P_3 & P_4 \end{Bmatrix} \quad (2.22)$$

A distribuição das matrizes que irão ser operadas na ScaLAPACK é responsabilidade do programador. Cada matriz a ser distribuída possui um *descriptor* anexado, que informa à biblioteca detalhes importantes durante as operações de álgebra linear, como: Dimensões globais da matriz, dimensão dos blocos necessários durante as operações, o *grid* ao qual essa matriz pertence, entre outros. A partir desse descriptor, podemos distribuir a matriz no formato *block-cyclic*.

Dado um descriptor que define blocos 2x2, a distribuição de uma matriz 16x16 utilizando o grid definido em 2.22 é definida da seguinte forma:

$$Dist_{matrix} = \begin{Bmatrix} P_1 & P_2 & P_1 & P_2 & P_1 & P_2 & P_1 & P_2 \\ P_3 & P_4 & P_3 & P_4 & P_3 & P_4 & P_3 & P_4 \\ P_1 & P_2 & P_1 & P_2 & P_1 & P_2 & P_1 & P_2 \\ P_3 & P_4 & P_3 & P_4 & P_3 & P_4 & P_3 & P_4 \\ P_1 & P_2 & P_1 & P_2 & P_1 & P_2 & P_1 & P_2 \\ P_3 & P_4 & P_3 & P_4 & P_3 & P_4 & P_3 & P_4 \\ P_1 & P_2 & P_1 & P_2 & P_1 & P_2 & P_1 & P_2 \\ P_3 & P_4 & P_3 & P_4 & P_3 & P_4 & P_3 & P_4 \end{Bmatrix} \quad (2.23)$$

Na matriz acima, cada elemento P_x representa uma submatriz de dimensão 4x4. A forma como os elementos são atribuídos aos blocos é bastante simples. A cada processo, um conjunto de coordenadas {x,y} é atribuída. A partir das coordenadas de linha e coluna dos

elementos de matriz, e das dimensões dos blocos, podemos definir quais conjuntos de linha e coluna pertencem a quais blocos a partir das seguintes equações a seguir:

$$P_{Row} = \left(\frac{Elem_{Row} - 1}{Blocks_{nrows}} \right) \text{mod } Grid_{nrows} \quad (2.24)$$

$$P_{Col} = \left(\frac{Elem_{Col} - 1}{Blocks_{ncols}} \right) \text{mod } Grid_{ncols} \quad (2.25)$$

A partir das coordenadas $\{P_{Row}, P_{Col}\}$, é possível indexar cada bloco da matriz à um processo do *grid* com as dadas coordenadas.

2.6 Considerações Finais

O procedimento de Hartree-Fock-Roothaan pode ser resolvido para que seja encontrada a função de onda de um sistema com muitos átomos. Devido à alta complexidade dos métodos *ab initio*, os **métodos semiempíricos** foram desenvolvidos, introduzindo aproximações no cálculo dos operadores e das propriedades com o intuito de eliminar alguns gargalos, e de tornar a computação mais eficiente como um todo. As aproximações dos métodos semiempíricos são apropriadas quando precisamos tratar sistemas quânticos com muitos átomos, como proteínas, nanotubos, zeólitas, etc.

Além disso, com o advento da computação de alto desempenho, é possível tratar sistemas ainda maiores ao utilizar sistemas distribuídos durante o cálculo, que são heterogêneos e facilmente incrementados. O paradigma de programação mais utilizado para a criação de sistemas computacionais que se beneficiam de arquiteturas distribuídas, é o de **passagem de mensagens**. A biblioteca MPI funciona como um *middleware* que facilita a passagem de mensagem entre os diferentes processos envolvidos durante o cálculo, e também é requisito para a utilização da biblioteca de álgebra linear distribuída *ScaLAPACK*.

A ScaLAPACK pode ser utilizada na resolução das operações de álgebra linear envolvidas no procedimento de *Hartree-Fock-Roothaan*, como a diagonalização do operador de Fock e a montagem da matriz densidade a partir dos coeficientes dos orbitais moleculares.

Baseando-se nesse conhecimento, é possível propor um novo *software* de química quântica semiempírico, apropriado para sistemas distribuídos, utilizando a biblioteca ScaLA-

PACK para a resolução das equações de Hartree-Fock-Roothaan. Porém, as operações de montagem dos operadores especificadas na seção 2.3.1 devem ser portadas para o formato distribuído específico da ScaLAPACK para a utilização das operações de álgebra linear da mesma. Os detalhes dessas implementações serão discutidos a seguir.

Capítulo 3

Trabalhos Relacionados

A literatura existente na área de modelagem computacional é extensa, e nesse capítulo serão abordados alguns trabalhos que implementam métodos Hartree-Fock semiempíricos, e que possuem algum nível de paralelização nos gargalos computacionais durante o SCF.

3.1 *Softwares* de Estrutura Eletrônica Paralelizados

A química computacional deu seus primeiros passos a partir da concepção do *software* Gaussian [Gaussian 2009] por John Pople, em 1970. O *software* proposto era capaz de realizar cálculos de estrutura eletrônica geral baseado na utilização de funções gaussianas como funções de base, e se mostrava bem mais eficiente que os seus competidores para cálculo de moléculas poliatômicas, já que além de prover novas funcionalidades, ainda possuía uma eficiência maior [Keinan e Schechter 2001]. Pople foi agraciado com o prêmio Nobel de química em 1998, e, no passado, já foi estimado que 90% de todos os cálculos de química quântica sejam realizados a partir de algum descendente computacional do Gaussian [Keinan e Schechter 2001]. Hoje em dia, esse *software* ainda é muito utilizado, e possui inúmeras funcionalidades implementadas: Vários métodos semiempíricos, até métodos altamente correlacionados com dados experimentais.

Além do Gaussian, existem outras opções de *software* disponíveis. O pacote ORCA é um programa de química quântica com praticamente todos os métodos de estrutura eletrônica modernos disponíveis, dos semiempíricos aos *ab initio*, e paralelização completa [Neese 2012]. Além disso, possui interface nativa de QM/MM com programas de dinâmica mole-

cular, como o GROMACS e o NAMD[Melo et al. 2018].

O *software* GAMESS (*General Atomic and Molecular Electronic Structure System*) [Schmidt et al. 1993] é um *software ab initio* capaz de calcular a função de onda SCF com diferentes níveis de exatidão. A maioria dos módulos quânticos do GAMESS podem se beneficiar de hardwares paralelos, como GPUs, e sistemas distribuídos.

Na mesma linha, o *software* NWChem provê algoritmos aos usuários que são capazes de tratar sistemas de grande complexidade, como biomoléculas, sistemas em estado sólido, e nanoestruturas [Valiev et al. 2010]. O NWChem também é capaz de tratar esses sistemas através de métodos *ab initio* e DFT de forma paralela. Porém, o mesmo não possui métodos semiempíricos implementados. O *software* NAMD, utiliza campos de força conhecidos, como o AMBER e CHARMM para processos de dinâmica molecular. Originalmente, o NAMD não possui nativamente um método de estrutura eletrônica na sua composição, mas em 2018, Melo e colaboradores propuseram uma interface do NAMD com programas de estrutura eletrônica existentes para a realização de cálculos híbridos QM/MM [Melo et al. 2018].

Para cálculos puramente DFT, o ErgoSCF é um *software* que possui como alvo sistemas moleculares de larga escala [Rudberg, Rubensson e Salek 2008]. Esse *software* é capaz de lidar com sistemas complexos devido à implementação de métodos de escalonamento linear durante o SCF, como a purificação da matriz densidade, o método dos multipolos rápidos, e a adoção de operações de álgebra linear utilizando matrizes esparsas de forma paralela.

Além do ErgoSCF, o *software* CP2K é um pacote que também implementa a maioria dos métodos DFT disponíveis. Porém, o CP2K é um *software* extremamente robusto, com vários métodos de modelagem molecular implementados. O módulo *Quickstep* implementa todos os métodos DFT, métodos Hartree-Fock de primeiros princípios, funcionalidades de escalonamento linear explorando a esparsidade inerente dos sistemas quânticos, cálculo de estados excitados, entre outros. A partir desse módulo, o CP2K também possibilita o cálculo de dinâmicas moleculares *ab initio*, e cálculos de dinâmica molecular híbridos QM/MM. Um dos pontos fortes do CP2K é a biblioteca distribuída para multiplicação de matrizes no formato esparsa DBCSR [Shan et al. 2014], resultantes após a utilização da técnica de escalonamento linear dos orbitais moleculares localizados.

3.1.1 Programas semiempíricos

Os métodos semiempíricos estão disponíveis em vários *softwares* diferentes. Grupos que parametrizam métodos semiempíricos geralmente tem um *software* próprio que facilita a calibração dos parâmetros numéricos envolvidos durante o cálculo e a formulação de novas equações que modelam as propriedades calculadas no SCF. É impossível falar de programas semiempíricos sem citar o MOPAC [Stewart 1990], que é considerado atualmente o programa padrão para métodos semiempíricos. O MOPAC foi desenvolvido em meados da década de 80, e foi de código aberto até a versão 7.1. As versões superiores do MOPAC são proprietárias, mas a sua licença é gratuita para acadêmicos.

Além do MOPAC, existem alguns outros *softwares* capazes de realizar cálculos semiempíricos. O *Software* VAMP [Biovia] implementa a maioria dos Hamiltonianos semiempíricos existentes. O mesmo também é capaz de calcular efeitos de solvente, otimizar a geometria, entre outros. Em 2014, Hennemann e Clark propuseram o EMPIRE [Hennemann e Clark 2014], um novo algoritmo semiempírico distribuído, que possui uma performance similar à do VAMP em um nó computacional, e escala eficientemente em até 1280 núcleos computacionais.

O MSINDO também é um programa e um método nativamente semiempírico. Ele iniciou-se primeiramente como uma modificação do método SINDO [Nanda e Jug 1980]. Além de possuir implementações que são capazes de calcular estados excitados de grandes sistemas moleculares [Gadaczek, Hintze e Bredow 2012], o programa também possui módulos para o cálculo de frequências vibracionais, e métodos procedimentos de localização dos orbitais moleculares.

O AMPAC [Semichem 2018] possui funcionalidades parecidas com as do MOPAC. Proposto em 1992, o AMPAC foi utilizado principalmente, segundo os idealizadores, para difundir os métodos semiempíricos propostos por Michael J.S. Dewar [Dewar et al. 1985] [Dewar e Thiel 1977]. Apesar de ser um código semiempírico que possui uma versão de demonstração, a sua versão completa exige uma licença paga até para propósitos acadêmicos.

3.2 Considerações Finais

A área de química quântica computacional é vasta, e a maioria dos métodos teóricos existentes na literatura já possuem algum tipo de implementação em *softwares* existentes, como o Gaussian, ORCA, e CP2K. Porém, métodos puramente semiempíricos distribuídos ainda são escassos, já que a complexidade computacional dos métodos *ab initio* e DFT é maior. Muitas vezes, os semiempíricos são encarados como já sendo otimizados "o suficiente", o que volta os esforços de implementação em outras direções.

Portanto, existe a necessidade de que seja proposto um método nativamente semiempírico, paralelo, distribuído, e que seja capaz de lidar com sistemas de grande complexidade, como proteínas e outras biomoléculas. A partir desse pensamento, na seção a seguir serão discutidos os detalhes de implementação do novo programa.

Capítulo 4

O Software Quantum Scenery

A concepção do programa *Quantum Scenery* (referenciado por QS daqui em diante), *a priori*, é de um código semiempírico, que seja de fácil utilização, e pensado do início para arquiteturas distribuídas em memória distribuída. Sistemas químicos são normalmente de alta complexidade, e o cálculo das suas propriedades pode escalonar com até $O(N^7)$ para métodos com maior capacidade de prever propriedades observadas experimentalmente. Porém, o presente trabalho tem como objetivo implementar as equações do método semiempírico NDDO, que apesar de ser computacionalmente custoso, possui complexidade computacional mais baixa, possibilitando o tratamento de sistemas com um maior número de átomos e de grande interesse químico, como biomoléculas e nanotubos.

O objetivo principal do QS é encontrar a função de onda semiempírica que representa o sistema molecular em questão. O output no final do procedimento detalhado na fig. 2.1 é justamente a função de onda desejada do sistema. A partir da função de onda, é possível extrair outras propriedades do sistema utilizando diferentes operadores. Nesse trabalho, o foco é o cálculo da energia total do sistema quântico em questão, visto que essa é uma propriedade que geralmente interessa na maioria das vezes. Portanto, o ponto de partida para a concepção do QS é a implementação do fluxograma da fig. 2.1.

Existem duas possíveis abordagens para a codificação desse procedimento utilizando uma abordagem distribuída em MPI. A primeira é a replicação da informação decorrente do SCF em cada processo, e cada processo trabalharia em um trecho apenas da informação. Essa abordagem facilita muito a escrita dos algoritmos paralelos, já que cada processo possui toda a informação necessária para o cálculo na memória. A falha nessa abordagem é o número de

mensagens trocadas para manter a informação atualizada em cada passo do procedimento. Como o procedimento SCF é iterativo, vários *broadcasts* em matrizes seriam necessários para manter as mesmas atualizadas e coerentes entre os processos até que o SCF seja finalizado. O tamanho máximo do sistema atômico também seria limitado superiormente pelo processo com menos memória disponível no ambiente distribuído, independentemente da quantidade de nós computacionais alocados.

A segunda abordagem possível é a distribuição da informação entre cada processo, e cada processo trabalha apenas na informação que ele tem disponível. Essa segunda abordagem é mais indicada, já que não é necessário o envio de informação redundante entre os nós, para que a coerência entre as matrizes alocadas seja mantida durante o SCF. O problema do limite superior do tamanho do sistema molecular trabalhado também é solucionado por essa abordagem. Se o usuário alocar mais nós computacionais durante o cálculo, a memória alocada necessária em cada nó é reduzida, já que não existe redundância entre a informação guardada entre os nós. A segunda abordagem foi escolhida para a concepção do QS.

É necessária a adoção de uma biblioteca de álgebra linear para a resolução dos procedimentos matriciais do SCF. A biblioteca escolhida foi a ScaLAPACK, que será responsável por realizar as operações de álgebra linear, e também por definir a distribuição das matrizes densas emergentes do cálculo SCF entre os processos através de um *grid* lógico gerado pelo usuário. Essa distribuição é explicada no capítulo 2.5.

Todas as matrizes envolvidas nos procedimentos de álgebra linear foram distribuídas de acordo com o formato *two-dimensional block-cyclic* da ScaLAPACK. Como a montagem das matrizes é de responsabilidade do programador, todos os operadores envolvidos no cálculo quântico foram montados de acordo com a especificação desse formato.

As sessões a seguir irão detalhar as atividades envolvidas no cálculo mais profundamente.

4.1 Input das Coordenadas Atômicas

O primeiro passo em um cálculo de química quântica semiempírico é realizar o parse das coordenadas atômicas, visto que a função de onda é função das coordenadas atômicas. O sistema molecular é representado a partir de um arquivo de input onde cada átomo é representado por 4 colunas.

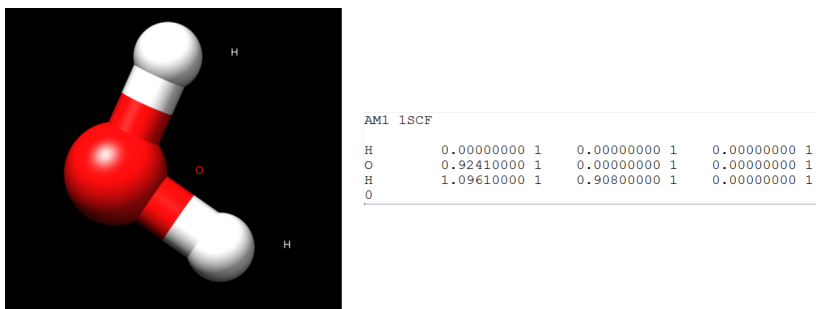


Figura 4.1: À esquerda, a geometria tridimensional da molécula de água e o seu arquivo de *input* representante à direita. As coordenadas tridimensionais dos átomos são especificadas utilizando o formato *xyz*.

Observando fig. 4.1, a primeira coluna do arquivo de *input* representa o símbolo atômico, contido na tabela periódica. As 3 colunas seguintes dizem respeito às coordenadas cartesianas de cada átomo no espaço. Além disso, nas primeiras linhas do arquivo, é possível notar o conjunto de palavras-chave que controlam opções durante o cálculo, como o conjunto de parâmetros do método semiempírico, se a geometria do sistema deve ser otimizada, se os tempos devem ser mostrados durante os cálculos, entre outros. Na fig. 4.1, foram informadas as palavras-chave **AM1** e **1SCF**, que informam ao programa, respectivamente, que conjunto de parâmetros a ser utilizado é o conjunto **AM1**, e que seja realizada um cálculo de energia nessa geometria fixa.

Para a versão inicial do QS, foram adotadas o conjunto de palavras-chave utilizadas no programa MOPAC [Stewart 2014], que possui versões de código aberto, e é o programa de referência em cálculos quânticos semiempíricos atualmente.

4.2 Palpite da Primeira Matriz Densidade

A partir das coordenadas dos átomos, agora é possível iniciar o cálculo SCF propriamente dito. Como vimos anteriormente, existe uma dependência circular durante o procedimento de SCF, obrigando a adoção de um método iterativo para a resolução do problema. Precisamos dos coeficientes dos orbitais moleculares para montar o operador de Fock (**F**), representado a partir de uma matriz. Os coeficientes dos orbitais são justamente a solução do problema, e são autovetores da matriz de Fock. Isso leva a um raciocínio simples: *é necessária uma*

matriz de Fock para que se obtenha uma nova. O procedimento exemplificado na seção 2.2 demonstra como podemos achar, a partir de um chute inicial dos coeficientes, a matriz de Fock final e a função de onda semiempírica.

O mais comum a partir disso, é chutar uma matriz densidade (\mathbf{P}) inicial. A partir de \mathbf{P} , é possível, segundo o procedimento de Hartree-Fock-Roothaan, encontrar \mathbf{F} , e sucessivamente \mathbf{P} a partir de \mathbf{F} , até que a convergência seja alcançada.

O palpite adotado da matriz \mathbf{P} é o de uma matriz diagonal, com o coeficiente da diagonal variando a partir da carga total do sistema, e da carga de cada átomo. A partir desse primeiro palpite, é possível montar a matriz de Fock. Esse procedimento será explicado adiante.

4.3 Montagem da matriz de Fock

O algoritmo de montagem da matriz de Fock é o procedimento mais complicado, do ponto de vista químico, durante o SCF. Observando as eqs. 2.12, 2.13 e 2.14, é possível notar que a montagem da matriz de Fock depende de 3 termos principais: A matriz densidade \mathbf{P} , uma matriz H_{core} , e termos $(\mu\nu|\lambda\sigma)$ que representam repulsões eletrônicas entre dois elétrons. A matriz \mathbf{P} pode ser encontrada primeiramente a partir do chute exposto na seção 4.2. Os outros termos serão esclarecidos a seguir.

4.3.1 Integrais de repulsão de 2-elétrons

As integrais de repulsão de 2-elétrons representam a energia de interação entre duas distribuições de cargas de dois orbitais atômicos de átomos diferentes [Dewar e Thiel 1977].

No trabalho de Dewar e colaboradores [Dewar e Thiel 1977], os autores estabelecem soluções analíticas para o valor dessas repulsões. Como a geometria definida é sempre a mesma, os valores das repulsões entre os elétrons não mudam durante o ciclo SCF. Isso permite duas abordagens diferentes para esse procedimento.

A primeira possibilidade é a pré-computação dessas integrais na memória principal no início do cálculo. Os valores das repulsões são armazenados em estruturas de dados e acessados quando necessário durante a montagem da matriz de Fock. Essa abordagem é bastante eficiente computacionalmente, pois o mesmo valor de integral é acessado várias vezes durante o ciclo SCF. Dewar e Thiel [Dewar e Thiel 1977] explicitaram que, para um par de

átomos pesados (Com até 4 orbitais atômicos), existem 22 integrais que devem ser armazenadas. Pares pesado-leve (4 orbitais atômicos x 1 orbital atômico) possuem 10 integrais de repulsão, e os pares leve-leve possuem apenas 1 integral a ser armazenada.

Não parece muito, mas se observarmos que as matrizes envolvidas no cálculo possuem dimensão N^2 , sendo N o número de orbitais atômicos do sistema, é perceptível que a estrutura de dados que contém as integrais escala de forma mais rápida e acentuada que as próprias matrizes do cálculo. Para sistemas muito grandes, pode não ser ideal pré-computar e armazenar as integrais de repulsão de 2-elétrons.

Para o fulereno C_{540} , por exemplo, que só contém carbonos na sua constituição, e possui 4 orbitais para cada carbono, a matriz de Fock possui um número de elementos igual a:

$$Dim(C_{540}) = \frac{(540 \times 4) \times (540 \times 4 + 1)}{2} = 2333880 \quad (4.1)$$

Já a estrutura de dados que armazena as integrais de repulsão possui uma dimensão de:

$$Intgs(C_{540}) = \left(100 \times 540\right) + \left(100 \times \frac{540 \times 541}{2}\right) = 14661000 \quad (4.2)$$

Para esse caso, a estrutura de dados que armazena as integrais de repulsão é cerca de 6,281 vezes maior que a matriz alocada durante o cálculo.

Outra dificuldade nessa abordagem é que o formato (e o acesso) à essas estruturas de dados é muito irregular durante a montagem da matriz de Fock. Isso complica o particionamento dessa estrutura entre os processos para particionar o trabalho durante a montagem da matriz de Fock: Um processo precisa de muitas posições diferentes e irregulares da estrutura de dados que armazena as integrais.

A outra possibilidade é o não armazenamentos das integrais de repulsão. O valor dos quartetos $(\mu\nu|\lambda\sigma)$ que representam as integrais retornados por uma função, que é chamada sempre que uma repulsão precisa ser avaliada. Essa abordagem é conhecida como abordagem *direta*.

O benefício da abordagem direta é que, obviamente, a estrutura de dados que armazena as integrais não precisa ser alocada. Porém, como cada integral é acessada várias vezes durante o processo de montagem da matriz de Fock, alguns valores serão avaliados várias vezes.

No QS, foram implementadas as duas opções de cálculo. Se o usuário optar por uma abordagem não-direta, cada processo executando o cálculo deverá alocar uma estrutura de

dados completa para armazenar todos os valores correspondentes às avaliações das integrais de repulsão. Se a abordagem for direta, cada processo invocará as funções de computação das integrais *on-the-fly*, recalculando valores de integrais quando necessário.

Na opção de cálculo não-direto, é necessário propor uma estrutura de dados que armazene as integrais de repulsão de 2-elétrons, e que permita o acesso a um valor de forma eficiente. É possível enxergar as integrais de repulsão de 2-elétrons como sendo um vetor tridimensional, e cada dimensão representa uma interação diferente.

É possível enxergar os termos de integrais como sendo:

$$Intg_{repulsao} = (\mu\nu|\lambda\sigma) \quad (4.3)$$

Onde μ e ν são orbitais de um dado átomo A , e λ e σ são orbitais de outro átomo B . Existe uma simetria entre os pares de átomos, onde o valor da integral é o mesmo se permutarmos os orbitais dos átomos:

$$(\mu_A\nu_A|\lambda_B\sigma_B) = (\lambda_B\sigma_B|\mu_A\nu_A) \quad (4.4)$$

De forma análoga, existe simetria permutacional entre os orbitais de cada átomo, ou seja:

$$(\mu_A\nu_A|\lambda_B\sigma_B) = (\nu_A\mu_A|\lambda_B\sigma_B) = (\mu_A\nu_A|\sigma_B\lambda_B) = (\nu_A\mu_A|\sigma_B\lambda_B) \quad (4.5)$$

O número de orbitais é um parâmetro de cada átomo, e pode variar dentro de um conjunto possível de valores. O hidrogênio é o único átomo que só contém 1 orbital. Todos os outros átomos possuem 4 orbitais. Então:

$$Set_{orbs}(Atom) = \begin{cases} \{s\}, & \text{Se } Atom = \text{'H'} \\ \{s, p_x, p_y, p_z\}, & \text{Se } Atom \neq \text{'H'} \end{cases} \quad (4.6)$$

Se tomarmos cada orbital como um valor inteiro ($s = 1, p_x = 2, \dots$), e levarmos em consideração a simetria permutacional dos orbitais, podemos enxergar um par de orbitais

como sendo um índice em uma matriz simétrica.

$$\begin{bmatrix} s s \\ p_x s & p_x p_x \\ p_y s & p_y p_x & p_y p_y \\ p_z s & p_z p_x & p_z p_y & p_z p_z \end{bmatrix} \implies \begin{bmatrix} 1 \\ 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 & 10 \end{bmatrix} \quad (4.7)$$

De forma análoga, é possível encarar a interação entre um par de átomos como sendo um índice em uma matriz simétrica. Portanto, podemos definir a nossa estrutura a partir desses 3 índices inteiros:

$$Indice_{atomos} \implies Indice_{orbitaisdeA} \implies Indice_{orbitaisdeB} \quad (4.8)$$

Assim, o algoritmo que armazena os valores na estrutura de dados segue:

A partir dos índices i , j , e k presentes no algoritmo 1, é possível acessar um valor de integral de repulsão de 2-elétrons da estrutura *intgs*. Em execuções que envolvem mais de um processo, essa estrutura deve ser alocada igualmente para cada um dos processos. Ainda não foi proposto nenhum esquema de particionamento dessa estrutura entre os processos, já que a irregularidade nos acessos aos valores durante o processo de montagem da matriz de Fock. Na literatura, já existem abordagens paralelas do cálculo das integrais de repulsão de 2-elétrons em sistemas de memória distribuída para métodos ab-initio [Furlani e King 1995] [Alexeev, Kendall e Gordon 2002]. Nesses métodos, esse tipo de cálculo representa o gargalo computacional do procedimento SCF, com complexidade de $O(N^4)$. Como o objetivo do presente trabalho é a função de onda *semiempírica*, o trecho do cálculo das integrais deixa de representar um gargalo computacional, diminuindo assim a sua importância da sua paralelização de forma eficiente. Portanto, nas execuções do QS com muitos processos, é preferível que seja empregada a abordagem direta.

4.3.2 Operador de 1-elétron

Como citado na seção 2.3.1, o operador de 1-elétron representa a energia cinética dos elétrons, e a atração entre o elétron e os núcleos do sistema trabalhado. Empregando as equações 2.9, 2.10 e 2.11, é possível construir esse operador sem mais dificuldades, conhecendo apenas a geometria do sistema e os parâmetros numéricos envolvidos no método. Porém, essa

matriz precisa ser construída de forma distribuída, já que um processo não alocará toda a informação ao mesmo tempo. Nesse paradigma, os processos precisam atualizar apenas as informações locais contidas nas suas regiões de memória, garantindo a divisão do trabalho. O algoritmo 2 monta a matriz que representa o operador de 1-elétron.

É possível notar que cada processo percorrerá apenas as suas matrizes locais, que já foram previamente distribuídas. As funções que, a partir de um índice local, inferem o índice global da matriz, também são fornecidas pela biblioteca. Os elementos da matriz de *overlap*, representada no algoritmo 2 como $S_{\mu\nu}$, são calculados *on-the-fly* durante a montagem do operador de 1-elétron. Como o cálculo do operador de 1-elétron é realizado apenas uma vez durante o SCF, o armazenamento dos *overlaps* não é necessário.

Dessa forma, a partir das posições μ e ν inferidas a partir dos índices μ_{loc} e ν_{loc} , é possível, a partir das equações 2.9, 2.10 e 2.11, calcular o valor associado à cada elemento da matriz de 1-elétron. A matriz de *output* do algoritmo 2 mantém a sua distribuição inalterada, e é necessária durante o procedimento de montagem da matriz de Fock, que será discutido a seguir.

4.3.3 Montagem do Operador de Fock

A partir das integrais de repulsão de 2-elétrons, e do operador de 1-elétron, é possível montar o operador de Fock. Esse operador é necessário para a avaliação da energia do sistema molecular, segundo a equação 2.16. Os autovalores e autovetores dessa matriz, representam, respectivamente, as energias e os coeficientes dos orbitais moleculares.

As equações 2.12, 2.13 e 2.14 regem a montagem do operador de Fock. Nas equações 2.12 e 2.13 é possível notar que, para o cálculo dos elementos de 1-Centro, a informação de todos os blocos da diagonal da matriz densidade \mathbf{P} é necessária. De forma análoga, na equação 2.14, blocos de 2-centros são necessários para a computação de cada elemento da matriz. A ScaLAPACK distribui as matrizes envolvidas no cálculo por blocos. Porém, como os blocos da ScaLAPACK possuem dimensão regular, e não variável, como é o caso das matrizes emergentes da química, é impossível definir uma dimensão de blocos que divida todo o trabalho de forma ótima. Além disso, é possível que um processo não tenha toda a informação necessária para calcular elementos pertencentes à blocos de 2-centros.

Para remediar essa situação, foram propostas duas estruturas de dados auxiliares que são

atualizadas entre os ciclos do SCF. A primeira é um vetor que contém todos os valores dos elementos de 1-centro da matriz \mathbf{P} . Essa estrutura deverá ser replicada entre todos os processos envolvidos no cálculo. Como apenas a informação dos blocos da diagonal é armazenada, essa estrutura de dados escala linearmente na memória, o que não acarreta um custo muito acentuado de memória entre cada processo. Como a informação da diagonal está disponível entre todos os processos, toda a informação que os elementos de 1-centro precisam para serem calculados está disponível. O algoritmo 3 explicita como calcular os termos de 1-centro da matriz de Fock.

É importante notar que a fórmula que calcula o $termo_2$ no algoritmo 3 é diferente entre os elementos da diagonal e fora dela. Se o elemento pertencer à diagonal, o $termo_2$ deverá ser calculado na linha 13. Se o elemento for de 1-centro e não pertencer à diagonal, a fórmula utilizada deverá ser representada pelo termo na linha 17. Em ambos os casos, os valores acessados da matriz são de 1-centro. Isso permite a utilização da primeira estrutura de dados proposta. O valor de P nas linhas 13 e 17 do algoritmo 3 foi retirado dessa estrutura auxiliar.

Toda a informação necessária para os elementos de 1-centro da matriz de Fock já está disponível. Os blocos de 2-centros da matriz de Fock são calculados a partir da equação 2.14, onde o termo $-\frac{1}{2} \sum_{\lambda \in A} \sum_{\sigma \in B} P_{\lambda\sigma}(\mu\lambda|\nu\sigma)$ não é trivialmente calculável. Os elementos $P_{\lambda\sigma}$ envolvidos no somatório representam valores do bloco de 2-centros da interação entre A e B. Esse bloco está distribuído entre os N processos envolvidos no cálculo semiempírico. Não é possível retirar informação da estrutura de dados envolvida no cálculo do algoritmo 3, já que a mesma só contém elementos de 1-centro. Portanto, é necessário que uma abordagem diferente seja utilizada nesse trecho do cálculo.

Na equação 2.14, os elementos acessados de \mathbf{P} durante a atualização de um bloco de 2-centros de \mathbf{F} são exatamente os mesmos. Isso quer dizer que se um processo contiver toda a informação acerca de um bloco de 2-centros da matriz \mathbf{P} , ele é capaz de calcular o termo $-\frac{1}{2} \sum_{\lambda \in A} \sum_{\sigma \in B} P_{\lambda\sigma}(\mu\lambda|\nu\sigma)$ para todos os elementos do bloco correspondente da matriz de Fock. Com isso, é possível delegar a responsabilidade da atualização de blocos inteiros da matriz de Fock para processos do cálculo. Para isso, é necessário que cada processo contenha a informação dos blocos de 2-centros de \mathbf{P} correspondentes aos blocos de 2-centros de \mathbf{F} que o mesmo precisa atualizar. A segunda estrutura de dados contém justamente essa informação: Os processos alocam um vetor auxiliar que conterà os blocos correspondentes

de 2-centros de \mathbf{P} para cada bloco de \mathbf{F} que o mesmo atualizará. Diferentemente da primeira estrutura de dados, a segunda estrutura é particular para cada processo, com dimensões e valores diferentes entre eles.

A partir dessa segunda estrutura de dados, é possível propor um procedimento que atualiza os blocos de 2-centros da matriz \mathbf{F} . O procedimento é descrito no algoritmo 4.

O algoritmo 4 explicita a montagem da parte de 2-centros da matriz de Fock. É importante notar que, apesar de um processo ser capaz de calcular um bloco de 2-centros inteiro a partir de valores correspondentes da matriz \mathbf{P} , as atualizações nos valores $F_{\mu\nu}$ é remota, já que o elemento a ser atualizado não necessariamente pertence ao processo que o está calculando. A partir dos procedimentos 3 e 4, é possível montar o operador de Fock adequado.

Com o operador de Fock e matriz densidade, é possível avaliar a energia total do sistema a partir da fórmula 2.16. Um dos critérios de convergência do QS é a diferença da energia total calculada entre as iterações do SCF. Os critérios de convergência serão discutidos na seção 4.3.4.

4.3.4 Operações de Álgebra linear e Critérios de Convergência

A partir do operador de Fock montado, é possível encontrar os coeficientes dos orbitais moleculares e as energias dos orbitais a partir de uma diagonalização do mesmo. A matriz \mathbf{F} pode ser diagonalizada utilizando a função **PDSYEVD** da ScaLAPACK, que encontra os autovalores, e opcionalmente os autovetores de um operador Hermitiano. Como estamos trabalhando com operadores reais e simétricos aqui, é possível utilizar essa função, já que a parte complexa do operador é zero.

A operação de montagem da matriz densidade é definida a partir da equação 2.8. A partir dos coeficientes dos orbitais moleculares, autovetores do operador de Fock, é possível calcular uma matriz densidade, e a partir dela, é possível montar um novo operador de Fock, e assim sucessivamente.

A montagem da matriz densidade pode ser definida como uma multiplicação de matrizes: $2CC^T$. Essa operação é largamente utilizada em vários problemas, e a ScaLAPACK possui funções que realizam esse tipo de cálculo. Em específico, a multiplicação de uma matriz simétrica pela sua transposta é definida na função **PDSYRK** da biblioteca. Portanto, a partir dos autovetores de saída, calcular a matriz densidade é trivial.

A partir de \mathbf{P} e \mathbf{F} , a equação 2.16 pode ser aplicada para a avaliação da energia total da iteração. A energia total pode ser calculada a partir da: (1) energia eletrônica do sistema e da (2) energia de repulsão nuclear. A energia eletrônica do sistema pode ser calculada a partir dos seguintes termos:

- O operador de 1-elétron H^{core}
- A matriz densidade
- O operador de Fock

A partir desses operadores, é possível aplicar a equação 2.15 e calcular o termo da energia eletrônica correspondente. Nessa equação, o valor dessa energia nada mais é do que uma operação de soma e multiplicação dos operadores, termo a termo. No nosso caso, como os termos estão distribuídos entre os processos, cada processo pode computar o seu resultado parcial, e com uma chamada da função **MPI_REDUCE**, esses resultados podem ser somados entre si e o valor da energia eletrônica pode ser recuperado de forma distribuída.

O termo da energia de repulsão nuclear pode ser calculado de forma análoga. Cada par de átomos A e B na equação 2.17 pode ser calculado por um processo utilizando um esquema análogo ao abordado no algoritmo 4. Após a execução de todos os processos, o valor da energia de repulsão nuclear pode ser calculado utilizando a função **MPI_REDUCE** entre os valores parciais.

Um dos critérios de parada do procedimento iterativo é a *diferença de energia* entre uma iteração e outra, ou seja:

$$Crit_{ener} = \begin{cases} V, & \text{Se } |E_n^{total} - E_{n-1}^{total}| \leq 10^{-6} \\ F, & \text{Se } |E_n^{total} - E_{n-1}^{total}| > 10^{-6} \end{cases} \quad (4.9)$$

O valor da energia total do sistema é dada a partir da equação 2.16. Para que o sistema atômico alcance a convergência, é necessário também que a matriz \mathbf{P} da iteração atual seja muito parecida com a matriz \mathbf{P} encontrada na iteração passada, ou seja:

$$Crit_{\mathbf{P}} = \begin{cases} V, & \text{Se } |P_n - P_{n-1}| \leq 10^{-6} \\ F, & \text{Se } |P_n - P_{n-1}| > 10^{-6} \end{cases} \quad (4.10)$$

No QS, calculamos a norma de Frobenius na matriz resultante da operação $P_n - P_{n-1}$. Essa norma é um procedimento computacionalmente eficiente e está implementada na rotina PDLANGE da ScaLAPACK. A partir do momento que os critérios 4.9 e 4.10 se tornam verdadeiros, o procedimento SCF pode parar e a energia final do sistema pode ser calculada.

4.4 Gradiente Numérico da Energia e Otimização de Geometria

A geometria mais estável de um sistema molecular é aquela que possui a menor energia total possível. Então, é possível enxergar a busca da geometria mais estável de uma molécula como sendo um procedimento de minimização da função-custo energia a partir dos parâmetros representados pelas coordenadas atômicas. Mínimos locais também são interessantes, pois representam pontos de equilíbrio e estruturas de transição entre duas conformações de um dado sistema molecular.

A maioria dos métodos de otimização de geometria atômica precisam da informação do gradiente da energia total calculada [Schlegel 2011]. Então, para a otimização de geometria, é necessário que seja possível a computação de um vetor gradiente associado à energia total calculada pelo SCF. Para a maioria dos níveis de teoria, o gradiente analítico pode ser computado com um peso computacional comparável ao de uma avaliação da energia total, e já existem formas de se calcular o gradiente analítico a partir da energia no método semiempírico MNDO [Dewar e Yamaguchi 1978]. O ponto negativo desse tipo de cálculo é que as integrais de repulsão de 2-elétrons devem ser recalculadas a partir de novas fórmulas.

O gradiente numérico pode ser utilizado como uma alternativa ao analítico. Além de ser mais eficiente, o gradiente numérico pode ser calculado facilmente a partir da energia total do sistema. Utilizamos a estratégia numérica implementada no MOPAC 7.1, que segue:

1. Armazenar a matriz P computada a partir do SCF;
2. Para todos os pares (i,j) de átomos do sistema:
 - (a) Montar um sistema diatômico a partir do par de átomos, onde a primeira matriz densidade é calculada a partir de P : São retirados os blocos de 1-centro dos

átomos i e j , e o bloco de 2-centros do par, e uma nova matriz densidade P_{der} é montada desses blocos;

- (b) Deslocar todas as coordenadas do átomo i em $5 \cdot 10^{-5}$ ângstroms
- (c) Calcular H_{der}^{core} e F_{der} a partir de P_{der} e das novas coordenadas;
- (d) Computar a energia $ener_1$ a partir de H_{der}^{core} , F_{der} , e P_{der} e das novas coordenadas;
- (e) Para cada coordenada x do átomo j :
 - i. Deslocar a coordenada x em $1,0 \cdot 10^{-4}$ ângstroms
 - ii. Calcular H_{der}^{core} e F_{der} a partir de P_{der} e das novas coordenadas;
 - iii. Computar a energia $ener_2$ a partir de H_{der}^{core} , F_{der} , e P_{der} e das novas coordenadas;
 - iv. Computar $der = \frac{ener_1 - ener_2}{1,0 \cdot 10^{-4}}$
 - v. Adicionar o valor calculado der na posição x dos átomos i e j no vetor gradiente

No final desse procedimento, é possível computar um gradiente correspondente para cada átomo. Esse gradiente pode ser utilizado por métodos de otimização *quasi-Newton* para estimar uma matriz Hessiana e realizar um procedimento de otimização local, com o objetivo de encontrar estados estacionários do sistema molecular.

Existem várias bibliotecas disponíveis que contêm os algoritmos necessários para a otimização da função-custo. Primeiramente, empregamos a biblioteca Ceres Solver durante o processo de otimização de geometria do sistema molecular, já que ela contém a maioria dos métodos baseados em gradientes implementados, como: BFGS, gradiente conjugado, *steepest descent*, entre outros.

Um dos critérios utilizados na otimização de geometria de sistemas quânticos é a norma do vetor gradiente calculado:

$$grad_{norm} = \sqrt{\sum_{a=1}^{N_{atomos}} x_a^2 + y_a^2 + z_a^2} \quad (4.11)$$

Se a norma do gradiente calculado for menor do que um critério pré-estabelecido pelo usuário, a otimização finaliza. Como a Ceres Solver utiliza um critério de convergência

diferente da norma euclidiana do gradiente, implementamos os seguintes critérios de convergência de forma externa:

1. O número de avaliações da energia total for maior que 500;
2. A norma do gradiente for ≤ 0.01 ;

Enquanto nenhum desses critérios é verdadeiro, o otimizador da Ceres é retroalimentado com os seus parâmetros de saída. Porém, foi observado preliminarmente que o processo de otimização da Ceres Solver não é eficiente para sistemas moleculares: Nem sempre o otimizador era capaz de chegar em uma geometria boa o suficiente com apenas uma execução, sendo necessário várias em sequência. Para alguns sistemas mais difíceis, nem várias execuções foram capazes de reduzir a norma do gradiente até 0.01.

Decidimos, por fim, utilizar o *software* DL-Find [Kästner et al. 2009] para otimizar a geometria dos nossos sistemas moleculares. Esse *software* é escrito como uma biblioteca, com o objetivo de ser *linkado* por outros códigos. O usuário fornece os métodos para computação das energias e das derivadas, e o algoritmo implementado se encarrega de otimizar as coordenadas dos átomos. A DL-FIND é uma alternativa viável à Ceres, já que, além de código aberto, ela é pensada para sistemas moleculares. Além de conter a maioria dos algoritmos de otimização, como o *steepest descent*, gradiente conjugado, *quasi-Newton* e L-BFGS, ela implementa um sistema de conversão entre coordenadas cartesianas e internas. Coordenadas internas são muito mais quimicamente relevantes do que cartesianas, e facilitam a obtenção de mínimos globais durante o processo de otimização da geometria [Baker 1993].

A DL-Find não utiliza o critério da norma do gradiente. Ao invés disso, a biblioteca emprega os seguintes critérios de convergência:

1. O valor absoluto do maior componente do vetor gradiente deve ser $\leq c = 0.00045$
2. O RMS (*root-mean-square*) do gradiente deve ser menor que $\frac{2c}{3}$
3. O valor absoluto do maior componente do vetor *step* deve ser menor que $4c$
4. O RMS do vetor *step* deve ser menor que $\frac{8c}{3}$
5. A variação na energia entre as duas últimas avaliações deve ser $\leq 10^{-6}$

A partir desses critérios, podemos fornecer um conjunto de coordenadas cartesianas, a função que avalia a energia e que calcula as derivadas, e a biblioteca se encarrega de encontrar uma conformação geométrica de mais baixa energia.

4.5 Considerações Finais

A energia semiempírica pode ser calculada a partir do procedimento SCF, que envolve algumas operações-chave durante a sua realização. A maioria dos *bottlenecks* computacionais envolvem operações de álgebra linear, que geralmente são paralelizáveis. Num ambiente de memória distribuída, é possível utilizar a biblioteca ScaLAPACK para o tratamento das matrizes durante o cálculo.

A partir das estruturas de dados definidas de acordo com o formato especificado na ScaLAPACK, é possível escrever algoritmos paralelos que montam os operadores de 1-elétron, Fock, e a matriz densidade. Existe uma dificuldade no processo de montagem do operador de Fock que é sanado a partir da introdução de duas estruturas de dados auxiliares.

A partir dos operadores e das operações de álgebra linear, o procedimento SCF pode ser inicializado, e sua execução dura até que os critérios de convergência sejam alcançados.

A partir dos últimos operadores, é possível calcular propriedades do sistema, como: o calor de formação, as cargas atômicas, a densidade eletrônica, e outros procedimentos. Além disso, é possível otimizar a geometria do sistema molecular de forma que o mesmo represente um sistema mais estável, de menor energia, através de um processo de otimização com a biblioteca DL-Find.

Na seção a seguir, iremos abordar a corretude do algoritmo implementado, bem como o seu desempenho em relação ao número de processos envolvidos no cálculo.

Algoritmo 1: Armazenamento das integrais de repulsão**Input:** Informações sobre a geometria do sistema**Output:** *intgs* como a estrutura de dados que contém as integrais de repulsão de 2-elétrons

```

1 begin
2    $i = 0$ 
3    $num_{it} = \frac{num_{atoms} \times (num_{atoms} + 1)}{2}$ 
4   while  $i \leq num_{it}$  do
5      $A = primeiroAtomo(i)$ 
6      $B = segundoAtomo(i)$ 
7      $j = 0$ 
8     while  $j \leq A.NumPairs()$  do
9        $\mu = A.primeiroOrbital(j)$ 
10       $\nu = A.segundoOrbital(j)$ 
11       $k = 0$ 
12      while  $k \leq B.NumPairs()$  do
13         $\lambda = B.primeiroOrbital(k)$ 
14         $\sigma = B.segundoOrbital(k)$ 
15         $intgs[i][j][k] = (\mu\nu|\lambda\sigma)_{A,B}$ 
16      end
17    end
18  end
19 end

```

Algoritmo 2: Algoritmo de montagem da matriz H_{core} distribuída

Input: Dados sobre a geometria do sistema, DistMatrix H_{core} **Output:** Matriz H_{core} montada

```

1  $\mu_{loc} = 0; \nu_{loc} = 0$ 
2 while ( $\mu_{loc} \leq Dim_x(H_{core})$ ) E ( $\nu_{loc} \leq Dim_y(H_{core})$ ) do
3    $\mu = H_{core}.IndiceGlobal(\mu_{loc}); \nu = H_{core}.IndiceGlobal(\nu_{loc})$ 
4    $A = AtomoDoOrbital(\mu); B = AtomoDoOrbital(\nu)$ 
5   if  $A == B$  then
6      $AtracaoAtomica = 0$ 
7     foreach  $\acute{A}tomo \neq A$  do
8        $AtracaoAtomica += (\mu\nu|ss)$ 
9     end
10     $H_{core}^{\mu_{loc}\nu_{loc}} += -1.0 \times AtracaoAtomica$ 
11    if  $\mu == \nu$  then
12       $H_{core}^{\mu_{loc}\nu_{loc}} += U_{\mu}$ 
13    end
14  end
15  else
16     $H_{core}^{\mu_{loc}\nu_{loc}} = \frac{1}{2}(\beta_{\mu} + \beta_{\nu})S_{\mu\nu}S_{\mu\nu}$ 
17  end
18 end

```

Algoritmo 3: Algoritmo de montagem dos elementos de 1-centro da matriz distribuída

F

Input: Matriz \mathbf{P} , informações sobre a geometria do sistema

Output: Elementos de 1-centro da matriz F atualizados

```

1   $\mu_{loc} = 0$ 
2  while  $m\mu_{loc} < Dim_x(F)$  do
3       $\nu_{loc} = 0$  while  $\nu_{loc} < Dim_y(F)$  do
4           $\mu = F.IndiceGlobal(\mu_{loc})$ 
5           $\nu = F.IndiceGlobal(\nu_{loc})$ 
6           $termo_2 = 0$ 
7           $termo_3 = 0$ 
8           $A = AtomoDoOrbital(\mu)$ 
9           $B = AtomoDoOrbital(\nu)$ 
10         if  $A == B$  then
11             if  $\mu = \nu$  then
12                 for Todos os orbitais  $\lambda \in A$  do
13                      $termo_2+ = P_{\lambda\lambda} \left[ (\mu\mu|\lambda\lambda) - \frac{1}{2}(\mu\lambda|\mu\lambda) \right]$ 
14                 end
15             end
16             else
17                  $termo_2 = P_{\mu\nu} \left[ \frac{3}{2}(\mu\nu|\mu\nu\mu\nu) - \frac{1}{2}(\mu\mu|\nu\nu) \right]$ 
18             end
19             for Todos os orbitais  $\lambda e \sigma \in B, \forall B \neq A$  do
20                  $termo_3+ = P_{\lambda\sigma} \left[ (\mu\mu|\lambda\sigma) \right]$ 
21             end
22              $F_{\mu\nu} = H_{\mu\nu}^{core} + termo_2 + termo_3$ 
23         end
24     end
25 end

```

Algoritmo 4: Algoritmo de montagem de 2-centros da matriz distribuída F

Input: Matriz \mathbf{P} , informações sobre a geometria do sistema

Output: Elementos de 2-centros da matriz F atualizados

```

1 for Todos os blocos de 2-centros do processo do
2    $A \leftarrow$  Primeiro atomo do bloco de 2-centros
3    $B \leftarrow$  Segundo atomo do bloco de 2-centros
4   foreach  $\mu \in A$  do
5     foreach  $\nu \in B$  do
6       foreach  $\lambda \in A$  do
7         foreach  $\sigma \in B$  do
8            $F_{\mu\nu+} = P_{\lambda\sigma}(\mu\lambda|\nu\sigma)$ 
9         end
10      end
11       $F_{\mu\nu+} = H_{\mu\nu}^{core}$ 
12    end
13  end
14 end

```

Capítulo 5

Resultados

No capítulo anterior, foram apresentados os aspectos computacionais da implementação do *software* QS. Nesse capítulo, iremos abordar os resultados obtidos a partir desse *software*.

Primeiramente, é necessário garantir que a energia total calculada é correta. O ideal seria calcular as características de forma experimental e comparar com os resultados obtidos pelo modelo semiempírico proposto. James Stewart, o idealizador do MOPAC, já comparou a exatidão dos métodos propostos com resultados experimentais no MOPAC (Comparação disponível em: openmopac.net/PM7_accuracy/molecules.html), portanto, uma possibilidade é comparar a energia total encontrada a partir do QS com a do MOPAC.

O conjunto de testes utilizado utilizado como validação da energia encontrada foi o mesmo empregado na parametrização do método RM1 [Rocha et al. 2006], que, por si só, representa um subconjunto do utilizado na parametrização dos métodos PM3, PM6, e PM7. Para cada sistema molecular do conjunto, fizemos um cálculo *single-point* do MOPAC e do QS, e comparamos a energia total encontrada entre os dois. No momento, o único conjunto de parâmetros implementado no programa é o RM1, então utilizamos o mesmo conjunto nas duas versões do cálculo.

A figura 5.1 explicita que, no quesito energia total, o *software* QS é praticamente a mesma a do programa semiempírico mais usado no geral. Como a energia é uma propriedade muito interessante do ponto de vista químico, é primordial que a energia do código proposto seja acurada.

É possível, a partir da energia e dos gradientes, otimizar as coordenadas dos átomos para encontrar a geometria mais estável do sistema molecular em questão. Resultados obtidos

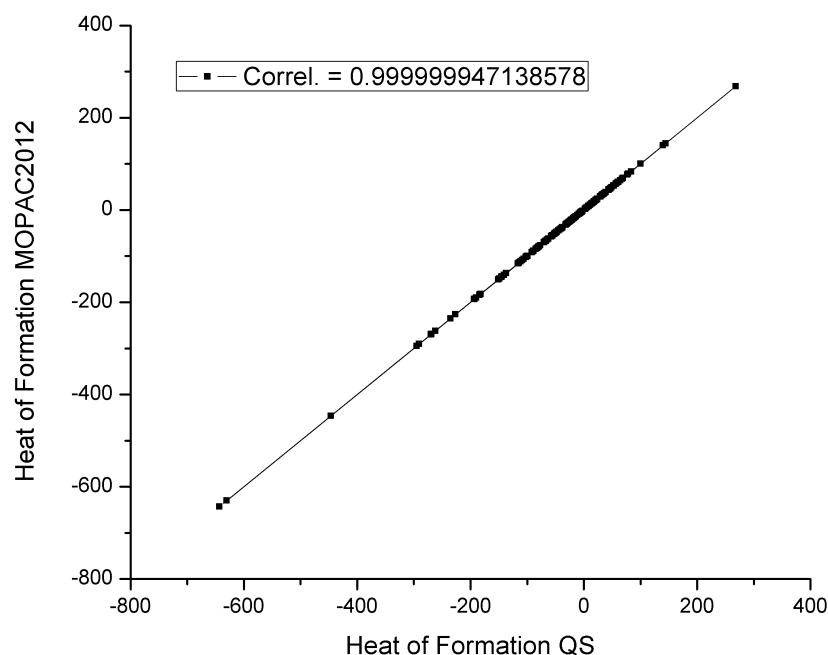


Figura 5.1: Correlação entre a energia encontrada a partir do MOPAC2012 e o QS para os sistemas contidos no conjunto de parametrização do RM1

anteriormente reforçaram a ideia de que é necessário realizar um procedimento de otimização próprio para sistemas moleculares. Consequentemente, a biblioteca DL-Find [Kästner et al. 2009] foi escolhida para otimizar a geometria dos sistemas atômicos a partir da energia semiempírica calculada pelo nosso método. Utilizamos o algoritmo L-BFGS implementado na biblioteca, juntamente com os gradientes numéricos calculados a partir do procedimento detalhado na seção 4.4 para otimizar a geometria dos sistemas do mesmo conjunto de testes utilizado na validação da energia.

Para cada sistema molecular do conjunto, realizamos um cálculo de otimização de geometria do MOPAC e do QS, e comparamos as posições finais dos átomos encontradas a partir desse procedimento. O critério de convergência aplicado ao MOPAC foi definido como sendo a norma euclidiana do gradiente calculado pelo mesmo: Se a norma calculada for menor do que 0.01, a otimização deverá parar. De forma análoga, estabelecemos empiricamente um critério $c = 0.05$ para a DL-Find, que foi julgado "bom o suficiente" para otimizar a geometria dos sistemas pertencentes ao conjunto. Com as duas estruturas finais

computadas por ambos os programas, utilizamos, como medida de comparação, o RMSD entre as duas geometrias.

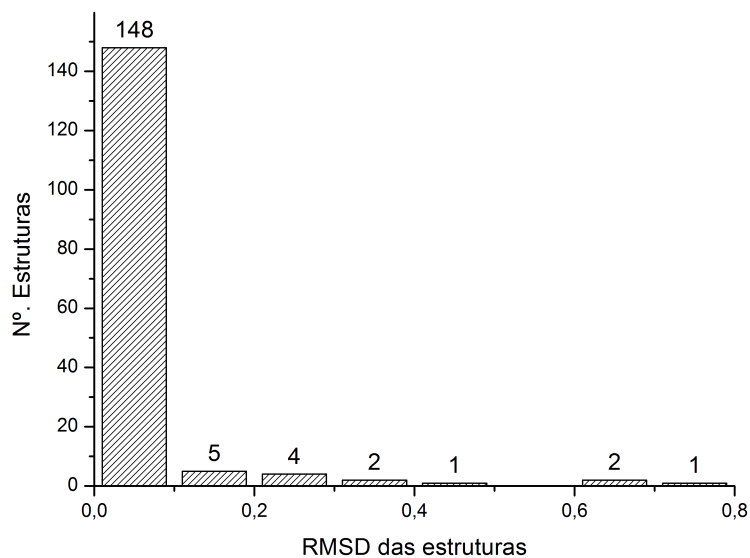


Figura 5.2: Histograma com o número de estruturas em cada intervalo de RMSD da DL-Find. 148 sistemas obtiveram um valor de $\text{RMSD} \leq 0.1$. Apenas 3 sistemas obtiveram um $\text{RMSD} \geq 0.5$, considerado alto para sistemas pequenos.

Na figura 5.2, podemos perceber a maioria das geometrias finais geradas a partir da otimização do MOPAC e do QS possuem um RMSD de valor < 0.1 . Geometrias com esse valor de RMSD geralmente são muito próximas, apesar de possuírem energias com pequenas diferenças a partir da terceira casa decimal. A maioria dos sistemas ($\approx 90\%$) possui um RMSD inferior a 0.1. Apenas 3 estruturas finais apresentaram um RMSD maior que 0.5.

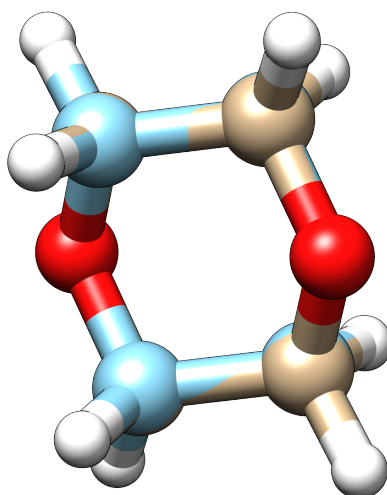


Figura 5.3: Superposição entre duas geometrias do 1,4-Dioxano, com RMSD < 0.1. Diferentes geometrias são representadas a partir das cores distintas nas ligações entre os carbonos.

A figura 5.3 representa uma superposição das duas geometrias finais encontradas pelo QS e MOPAC. Os carbonos da geometria encontrada pelo QS são ilustrados em marrom. Os carbonos em azul pertencem à geometria encontrada pelo processo de otimização do MOPAC. A partir dessa figura, é possível concluir que as duas geometrias são praticamente as mesmas.

A estrutura que apresentou o maior RMSD entre as geometria finais (≈ 0.73) foi o fenil-nitrometano. A sua superposição segue:

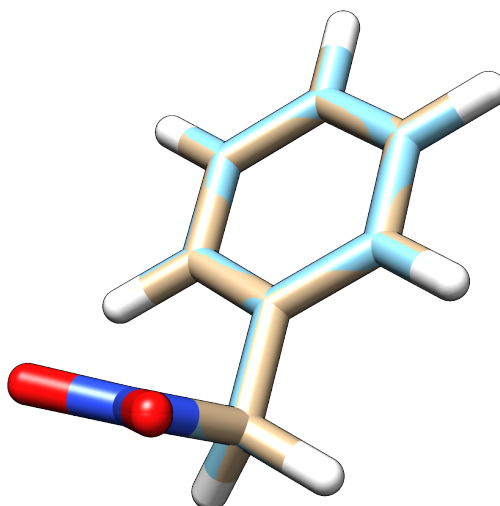


Figura 5.4: Superposição das duas estruturas finais encontradas do fenil-nitrometano, com $\text{RMSD} \approx 0.73$. De forma análoga à figura 5.3, geometrias diferentes são representadas a partir de cores distintas nas suas ligações entre os carbonos

A partir da figura 5.4, não é perceptível, a priori, que as duas estruturas não são idênticas. Porém, se observarmos com cuidado as estruturas representadas por fios, é possível notar que o *overlap* entre as duas não é exato.

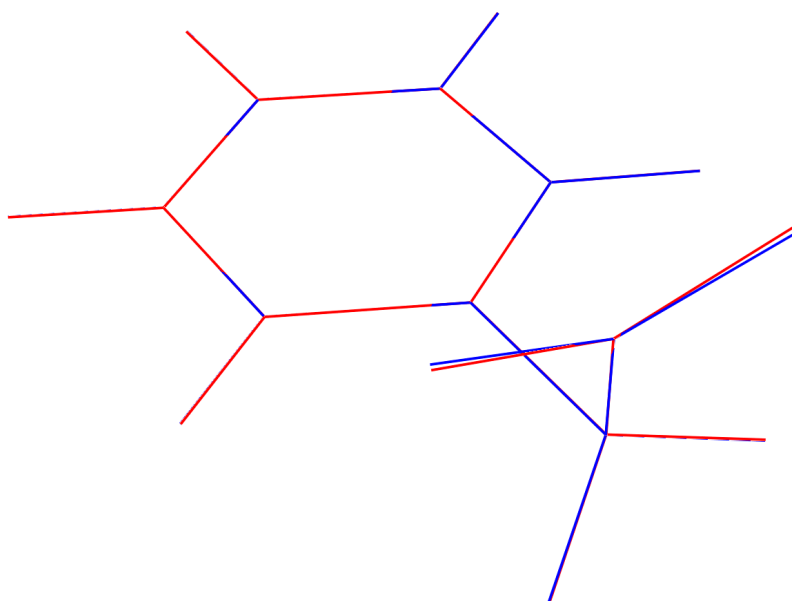


Figura 5.5: Superposição das duas geometrias em formato de fio do fenil-nitrometano, com $\text{RMSD} \approx 0.73$. Acima, a estrutura encontrada pelo MOPAC é representada em vermelho, e a estrutura final do QS, em azul.

Na figura 5.5, a estrutura final calculada pelo MOPAC é mostrada em azul, e a estrutura calculada pelo QS, em vermelho. Percebemos que, na região do anel aromático, a sobreposição entre as duas estruturas é quase perfeita. Já nos oxigênios do nitrometano as duas estruturas não se sobrepõem exatamente. Apesar disso, a diferença de energia entre as duas estruturas é de ≈ 0.002 eV. Então, é plausível concluir que o método de otimização de geometria implementado no QS é robusto o suficiente. Um possível melhoramento seria a introdução do cálculo dos gradientes analíticos da energia semiempírica [Dewar e Yamaguchi 1978], que apesar de mais lentos, correspondem ao valor exato da primeira derivada da energia.

Além disso, também foi proposto um algoritmo distribuído para a montagem dos operadores e cálculo dos autovetores utilizando a ScaLAPACK. Testamos o desempenho desse algoritmo com o auxílio do *cluster* computacional do departamento de química da UFPB. Foram reservados dois nós computacionais com: Dois Processadores Intel Xeon E5-2630 v2

2.6GHz e 64GB de memória principal, conectados através de uma rede Infiniband para troca de mensagens MPI.

Variamos o número de processos MPI envolvidos no cálculo da energia de 2 até 48. Como dispomos de apenas 2 nós computacionais, cada um contendo 12 núcleos físicos e 24 *threads* virtuais, não seria vantajoso aumentar o número de processos MPI do cálculo além do valor máximo de *threads* virtuais. Utilizamos como sistema de teste uma caixa esférica de raio 20Å com 1120 moléculas de água, 3360 átomos e 6720 orbitais. Os tempos de execução seguem:

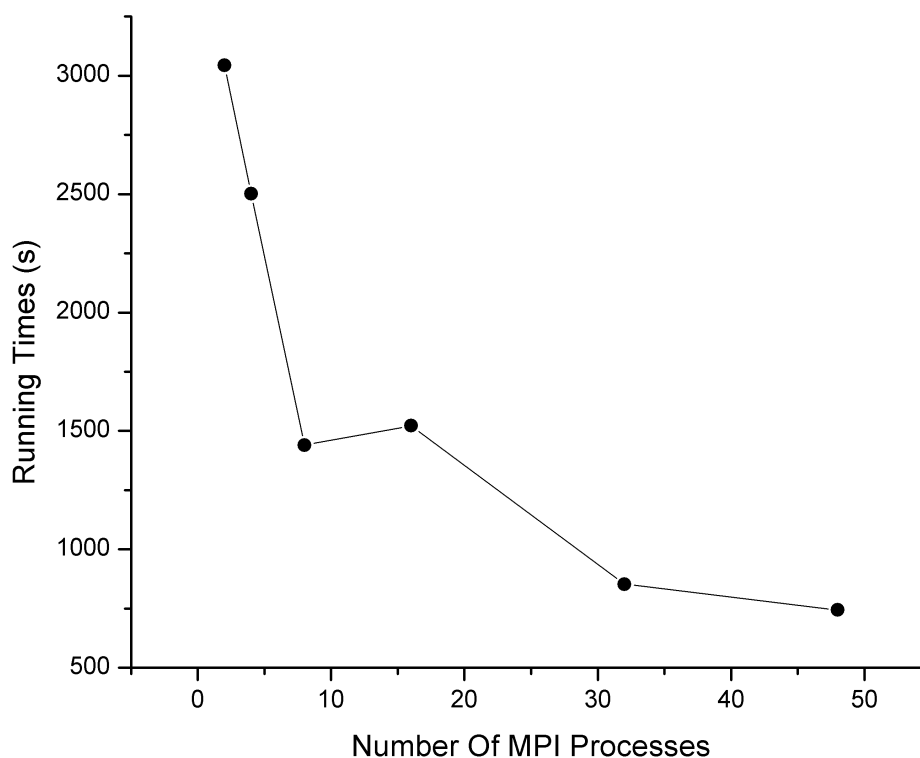


Figura 5.6: Tempos total do cálculo da energia para uma caixa esférica de 20Å para um número de processos MPI $N=\{2,4,8, 16, 32 \text{ e } 48\}$

A partir da figura 5.6, é perceptível que o tempo de execução decresce com o número de processos envolvidos no cálculo. Porém, para algumas instâncias, a diminuição no tempo de execução não é tão acentuada. O ganho de desempenho entre o cálculo com número de

processos $N=2$ e $N=4$ é de apenas 21%. Porém, se compararmos os tempos de execução entre $N=4$ e $N=8$, o ganho de desempenho é de 73%. Essas avaliações podem ser aprofundadas se observarmos o tempo de execução total de cada procedimento do cálculo de energia individualmente, e os seus pesos no tempo total de execução.

N	Fock	Diagonalização	Densidade	Estruturas	Total
2	2345.11	552.85	28.5	60.04	2986.5
4	1732.08	644.95	28.01	41.91	2446.95
8	857.09	481.65	36.72	31.93	1407.39
16	884.07	508.9	38.74	45.94	1477.65
32	450.65	298.81	22.79	42.38	814.63
48	404.93	242.47	16.1	42.24	705.74

Tabela 5.1: Tempos totais em segundos de execução dos procedimentos computacionalmente custosos durante o cálculo do SCF, para cada valor de N

A tabela 5.1 contém os tempos totais de execução com um dado número de processos N , para cada procedimento computacionalmente custoso pertencente ao cálculo do SCF. Como já esperado, a montagem da matriz de Fock aparece como o *hotspot* do cálculo para todos os valores de N , visto que, para cálculos distribuídos, empregamos uma abordagem direta, onde todas as integrais de repulsão de 2-elétrons (Detalhadas na subseção 4.3.1) são computadas *on-the-fly*. Como o procedimento de montagem desse operador é trivialmente paralelizável, i.e. não existe dependência de dados entre os valores a serem calculados da matriz, esse procedimento escala muito bem com o aumento do número de processos no cálculo. O tempo de execução diminui em ≈ 6 vezes para o maior valor de N .

Para o procedimento de diagonalização da matriz de Fock, foi empregada a biblioteca ScaLAPACK. Nas duas primeiras instâncias de cálculo, o tempo de execução da biblioteca aumenta, o que explica o pouco ganho de performance evidenciado na fig. 5.6. As estratégias para otimização da ScaLAPACK em sistemas distribuídos (Apresentadas em <http://netlib.org/scalapack/slug/node106.html>), como a dimensão dos blocos utilizados para distribuição, e a escolha correta das dimensões do grid de processos, já foram implementadas. Por consequência, a performance do diagonalizador melhora a partir de $N=8$.

Um detalhe importante é a análise do caso $N=16$. Ao observar a fig. 5.6, o tempo de execução total dessa instância aumenta. Esse comportamento é comum para sistemas computacionais Intel com *hyperthreading*, que buscam otimizar a utilização dos recursos de processamento permitindo um aumento do número de *threads* executando concorrentemente nos núcleos físicos. Algoritmos com controles de fluxo complicado, onde cada *thread* executa conjuntos de instruções diferentes se beneficiam bem desse tipo de tecnologia. Porém, em problemas de alto desempenho, esse não é o caso. A maioria das *threads* em execução seguem as mesmas instruções, e são executadas em alta eficiência, descartando a necessidade de *hyperthreading* para otimização da utilização dos recursos computacionais. Como foram utilizados dois nós Intel com *hyperthreading*, o sistema de fila utilizado "enxerga" 24 núcleos disponíveis por nó, sendo 12 físicos e 12 virtuais. Quando é requisitado ao sistema de fila um cálculo contendo 2 nós e 16 processos, o mesmo é escalonado para apenas um nó. O processador desse só possui 12 núcleos físicos, e, conseqüentemente, não haverá um ganho de desempenho. Pelo contrário: Nesse caso, os 4 processos excedentes competiram pelos núcleos físicos e diminuíram a performance geral do cálculo.

Essas afirmações são corroboradas pela próxima instância do cálculo: $N=32$. Como nenhum dos nós computacionais possui 32 *threads* disponíveis, o mesmo é escalonado para os 2 nós requisitados simultaneamente. Isso acarreta na execução de 16 processos em um nó, e 16 em outro. Como estamos aumentando o número de núcleos físicos envolvidos no cálculo, o tempo de execução diminui bastante, apresentando um ganho de desempenho de 81% com relação ao cálculo anterior com $N=16$. Essa análise confirma previamente que o sistema escalonou bem com o número de nós computacionais envolvidos no cálculo, e pode ser ampliada no futuro para um número de nós computacionais maior.

5.1 Considerações Finais

Os resultados obtidos indicam que a avaliação da energia total do sistema é acurada. Realizamos o cálculo SCF para sistemas pertencentes ao conjunto de benchmark do RM1, e as energias finais encontradas são correspondentes às do *software* MOPAC. Além disso, também foi implementado um procedimento de cálculo dos gradientes numéricos a partir da energia semiempírica calculada. Com os gradientes, é possível empregar um método de

otimização baseado em gradientes para encontrar a conformação espacial atômica que corresponde à menor energia possível. O processo de otimizar a posição dos átomos é conhecido como otimização de geometria.

Esse procedimento foi implementado a partir da biblioteca DL-Find. De todos os sistemas moleculares do conjunto de testes, apenas 3 estruturas final calculadas pelo QS divergiram em mais de 0.5 no seu valor de RMSD das estruturas finais computadas pelo MOPAC. Para o sistema com maior valor de RMSD entre as estruturas finais, a energia associada à essa estrutura calculada pelo QS é bastante próxima à do MOPAC (Com uma diferença de ≈ 0.002). Com isso, julgamos que o método de otimização implementado na DL-Find é robusto o suficiente para os sistemas moleculares trabalhados. Futuros melhoramentos podem incluir o cálculo dos gradientes numéricos para resultados mais exatos, e as modificações do algoritmo BFGS para sistemas moleculares propostas por Stewart (Disponível em http://openmopac.net/manual/BFGS_optimizer.html).

Além disso, pudemos concluir que o método proposto conseguiu se beneficiar da utilização de vários núcleos computacionais na sua execução. Para um sistema com 3360 átomos, o tempo computacional necessário para a execução de um cálculo SCF diminuiu com o aumento do número de núcleos computacionais envolvidos. No melhor caso, com 2 nós computacionais e 48 processos, o cálculo da energia apresentou um *speedup* de ≈ 6 vezes. No futuro, podemos ampliar essa análise de desempenho para um número maior de nós computacionais disponíveis.

Capítulo 6

Conclusão

O presente trabalho propôs um novo *software* de mecânica quântica semiempírico para sistemas distribuídos. A primeira versão do programa já foi finalizada, e já temos um algoritmo distribuído que avalia a energia a partir de uma geometria fixa no espaço. Além disso, também foi proposto um algoritmo distribuído que computa a derivada da energia, e a partir dele, um procedimento de otimização de geometria a partir da biblioteca DL-Find. Os resultados mostraram que, além de conseguirmos uma energia acurada, o procedimento de otimização de geometria a partir da DL-Find foi capaz de encontrar geometrias finais muito próximas da correspondente ao procedimento do MOPAC. Para alguns sistemas, muitas avaliações da função-custo são necessárias para que uma energia próxima suficiente do mínimo seja encontrada.

Além disso, também comprovamos a eficiência do algoritmo distribuído proposto. Para um ambiente de memória distribuída, com 2 nós computacionais. Vimos que, para o melhor dos casos ($N=48$), nosso algoritmo é aproximadamente 6 vezes mais rápido que o cálculo sequencial, reforçando a ideia de que a sua performance escala bem com o número de núcleos computacionais envolvidos. Futuramente, pretendemos ampliar esses experimentos, aumentando o número de nós computacionais. Também é uma expectativa que o programa leve em consideração nós contendo múltiplas GPUs, e se beneficie dos seus muitos núcleos de processamento.

Finalmente, existem vários outros desenvolvimentos futuros que são interessantes do ponto de vista químico. A modificação mais importante seria a introdução de um método que descreve átomos com orbitais 'd'. Isso permitiria o cálculo de propriedades de sistemas

que contém átomos pesados, como o enxofre, e metais. Já existem métodos semiempíricos que levam em consideração esses orbitais [Thiel e Voityuk 1992]. Porém, o número de integrais de repulsão de 2-elétrons a serem avaliadas aumenta muito durante a montagem do operador de Fock (2025 integrais a serem avaliadas entre 2 orbitais 'd'), aumentando consideravelmente o esforço de desenvolvimento nesse trecho.

Outra adição importante ao código seria um método de avaliação da energia dos sistemas que apresente uma complexidade linear com relação ao número de átomos do sistema molecular. Esses métodos tratam a *esparsidade* natural dos sistemas moleculares, e ignoram os zeros emergentes dos operadores durante as operações de álgebra linear. Sistemas muito complexos, como proteínas, apresentam um alto grau de esparsidade nas matrizes que representam os operadores durante o cálculo, e biblioteca de álgebra linear que tratam com formatos esparsos podem ser empregadas nesse passo [Borštnik et al. 2014]. Essa abordagem permitiria o tratamento de sistemas muito maiores, e representaria o último esforço de implementação para a modelagem de biomoléculas com muitos átomos.

Referências Bibliográficas

- [Alexeev, Kendall e Gordon 2002]ALEXEEV, Y.; KENDALL, R. A.; GORDON, M. S. The distributed data SCF. *Computer Physics Communications*, v. 143, n. 1, p. 69–82, 2002. ISSN 00104655.
- [Anderson et al. 1991]ANDERSON, E. et al. Basic Linear Algebra Communication Subprograms. In: *Sixth Distributed Memory Computing Conference Proceedings*. [S.l.: s.n.], 1991. p. 287–290.
- [Angerson et al. 1990]ANGERSON, E. et al. LAPACK: A portable linear algebra library for high-performance computers. *Proceedings SUPERCOMPUTING '90*, 1990.
- [Baker 1993]BAKER, J. Techniques for geometry optimization: A comparison of cartesian and natural internal coordinates. *Journal of Computational Chemistry*, v. 14, n. 9, p. 1085–1100, 1993. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.540140910>>.
- [Barrachina et al. 2008]BARRACHINA, S. et al. Evaluation and tuning of the level 3 CUBLAS for graphics processors. In: *IPDPS Miami 2008 - Proceedings of the 22nd IEEE International Parallel and Distributed Processing Symposium, Program and CD-ROM*. [S.l.: s.n.], 2008. ISBN 9781424416943. ISSN 1530-2075.
- [Biovia]BIOVIA. *Biovia Materials Studio VAMP*.
- [Borštnik et al. 2014]BORŠTNIK, U. et al. Sparse matrix multiplication: The distributed block-compressed sparse row library. *Parallel Computing*, v. 40, n. 5-6, p. 47–58, 2014. ISSN 01678191.

- [Choi et al. 1992]CHOI, J. et al. ScaLAPACK: a scalable linear algebra library for distributed memory concurrent computers. In: *ScaLAPACK*. [S.l.: s.n.], 1992. p. 120–127. ISBN 0-8186-2772-7.
- [Dewar e Thiel 1977]DEWAR, M. J.; THIEL, W. Ground States of Molecules. 38. The MNDO Method. Approximations and Parameters. *Journal of the American Chemical Society*, v. 99, n. 15, p. 4899–4907, 1977. ISSN 15205126.
- [Dewar e Yamaguchi 1978]DEWAR, M. J.; YAMAGUCHI, Y. Analytical first derivatives of the energy in MNDO. *Computers & Chemistry*, v. 2, n. 1, p. 25–29, jan 1978. ISSN 00978485. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/0097848578800059>>.
- [Dewar e Thiel 1977]DEWAR, M. J. S.; THIEL, W. A Semiempirical Model for the Two-Center-Repulsion Integrals in the NDDO Approximation. *Theor. Chim. Acta*, v. 46, p. 89., 1977.
- [Dewar e Thiel 1977]DEWAR, M. J. S.; THIEL, W. Ground States of Molecules. 38. The MNDO Method. Approximations and Parameters. *Journal of the American Chemical Society*, v. 99, n. 15, p. 4899–4907, 1977. ISSN 15205126.
- [Dewar et al. 1985]DEWAR, M. J. S. et al. Development and use of quantum mechanical molecular models. 76. AM1: a new general purpose quantum mechanical molecular model. *Journal of the American Chemical Society*, v. 107, n. 13, p. 3902–3909, 1985. ISSN 0002-7863. Disponível em: <<http://dx.doi.org/10.1021/ja00299a024%5Cnhttp://pubs.acs.org/doi/abs/10.1021/ja00299a024>>.
- [Field, Bash e Karplus 1990]FIELD, M. J.; BASH, P. A.; KARPLUS, M. A Combined Quantum Mechanical and Molecular Mechanical Potential for Molecular Dynamics Simulations. *J. Comput. Chem.*, v. 11, n. 6, p. 700–733, 1990. ISSN 0192-8651. Disponível em: <<http://doi.wiley.com/10.1002/jcc.540110605>>.
- [Furlani e King 1995]FURLANI, T. R.; KING, H. F. Implementation of a parallel direct SCF algorithm on distributed memory computers. *Journal of Computational*

- Chemistry*, v. 16, n. 1, p. 91–104, jan 1995. ISSN 0192-8651. Disponível em: <<http://doi.wiley.com/10.1002/jcc.540160108>>.
- [Gadaczek, Hintze e Bredow 2012]GADACZEK, I.; HINTZE, K. J.; BREDOW, T. Periodic calculations of excited state properties for solids using a semiempirical approach. *Phys. Chem. Chem. Phys.*, The Royal Society of Chemistry, v. 14, p. 741–750, 2012. Disponível em: <<http://dx.doi.org/10.1039/C1CP22871D>>.
- [Gaussian 2009]GAUSSIAN, I. *Gaussian*. 2009. Disponível em: <<http://gaussian.com/>>.
- [Goto e Van De Geijn 2008]GOTO, K.; Van De Geijn, R. High-performance implementation of the level-3 BLAS. *ACM Transactions on Mathematical Software*, v. 35, n. 1, p. 1–14, 2008. ISSN 00983500. Disponível em: <<http://portal.acm.org/citation.cfm?doid=1377603.1377607>>.
- [Hennemann e Clark 2014]HENNEMANN, M.; CLARK, T. EMPIRE: a highly parallel semiempirical molecular orbital program: 1: self-consistent field calculations. *Journal of Molecular Modeling*, v. 20, n. 7, p. 2331, jul 2014. ISSN 1610-2940. Disponível em: <<http://link.springer.com/10.1007/s00894-014-2331-4>>.
- [Hestenes e Stiefel 1952]HESTENES, M.; STIEFEL, E. *Methods of conjugate gradients for solving linear systems*. 1952. 409 p. Disponível em: <http://nvlpubs.nist.gov/nistpubs/jres/049/jresv49n6p409_A1b.pdf>.
- [Janssen e Nielsen 2008]JANSSEN, C. L.; NIELSEN, I. M. B. *Parallel computing in quantum chemistry*. 2008. 445–450 p.
- [Karplus e McCammon 2002]KARPLUS, M.; MCCAMMON, J. A. Molecular dynamics simulations of biomolecules. *Nature Structural Biology*, v. 9, n. 9, p. 646–652, 2002. ISSN 10728368. Disponível em: <<http://dx.doi.org/10.1038/nsb0902-646>>.
- [Kästner et al. 2009]KÄSTNER, J. et al. DL-FIND: An open-source geometry optimizer for atomistic simulations*. *Journal of Physical Chemistry A*, v. 113, n. 43, p. 11856–11865, 2009. ISSN 10895639.
- [Keinan e Schechter 2001]KEINAN, E.; SCHECHTER, I. *Chemistry for the 21st Century*. [S.l.: s.n.], 2001. 308 p. ISBN 9783527612956.

- [Li e Fukushima 2001]LI, D.-H.; FUKUSHIMA, M. On the Global Convergence of the BFGS Method for Nonconvex Unconstrained Optimization Problems. *SIAM Journal on Optimization*, v. 11, n. 4, p. 1054–1064, 2001. ISSN 1052-6234.
- [Maia et al. 2012]MAIA, J. D. C. et al. GPU linear algebra libraries and GPGPU programming for accelerating MOPAC semiempirical quantum chemistry calculations. *Journal of Chemical Theory and Computation*, v. 8, n. 9, p. 3072–3081, 2012. ISSN 15499618.
- [Melo et al. 2018]MELO, M. C. R. et al. NAMD goes quantum: an integrative suite for hybrid simulations. *Nature Methods*, 2018. ISSN 1548-7091. Disponível em: <<http://www.nature.com/doi/10.1038/nmeth.4638>>.
- [Minkenber 2013]MINKENBERG, C. *Interconnection Network Architectures for High-Performance Computing HPC interconnection networks*. 2013.
- [Nanda e Jug 1980]NANDA, D. N.; JUG, K. SINDO1. A semiempirical SCF MO method for molecular binding energy and geometry I. Approximations and parametrization. *Theoretica Chimica Acta*, v. 57, n. 2, p. 95–106, 1980. ISSN 00405744.
- [Neese 2012]NEESE, F. The ORCA program system. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, v. 2, n. 1, p. 73–78, 2012. ISSN 17590876.
- [Nobel 1998]NOBEL. *The Nobel Prize in Chemistry 1998*. 1998.
- [Nobel 2013]NOBEL. *The Nobel Prize in Chemistry 2013*. 2013.
- [Pacheco 1996]PACHECO, P. *Parallel Programming with MPI*. 1. ed. San Francisco - CA: Morgan Kaufmann Publishers Inc., 1996. ISBN 1558603395.
- [Phillips et al.]PHILLIPS, J. C. et al. Scalable molecular dynamics with namd. *Journal of Computational Chemistry*, v. 26, n. 16, p. 1781–1802. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.20289>>.
- [Rocha et al. 2006]ROCHA, G. B. et al. RM1: A reparameterization of AM1 for H, C, N, O, P, S, F, Cl, Br, and I. *Journal of Computational Chemistry*, v. 27, n. 10, p. 1101–1111, 2006. ISSN 01928651.

- [Rocha G.B.; Simas 2007]ROCHA G.B.; SIMAS, A. Métodos semiempíricos de estrutura eletrônica em química quântica. In: MORGON N.H.; COUTINHO, K. (Ed.). *Métodos de Química Teórica E Modelagem Molecular*. Rua Enéas Luis Carlos Barbanti, 193 - Freguesia do Ó - São Paulo: Editora Livraria da Física, 2007. cap. 2.
- [Rudberg, Rubensson e Salek 2008]RUDBERG, E.; RUBENSSON, E. H.; SALEK, P. Hartree-Fock calculations with linearly scaling memory usage. *Journal of Chemical Physics*, v. 128, n. 18, 2008. ISSN 00219606.
- [Schlegel 2011]SCHLEGEL, H. B. Geometry optimization. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, v. 1, n. 5, p. 790–809, 2011. ISSN 17590876.
- [Schmidt et al. 1993]SCHMIDT, M. W. et al. General atomic and molecular electronic structure system. *Journal of Computational Chemistry*, v. 14, n. 11, p. 1347–1363, 1993. ISSN 1096987X.
- [Seaton 1977]SEATON, M. J. *Hartree-Fock method*. 1977. 631–631 p.
- [Semichem 2018]SEMICHEM, I. *AMPAC Semiempirical Quantum Mechanics*. 2018. <http://www.semichem.com/ampac/>. Acessado: 2018-25-04.
- [Shan et al. 2014]SHAN, H. et al. Sparse matrix multiplication: The distributed block-compressed sparse row library. *The Journal of Chemical Physics*, v. 4, n. 2, p. 47–58, jul 2014. ISSN 15499618. Disponível em: <<http://link.springer.com/10.1007/s00894-014-2331-4> http://www.daxia.com/bibis/upload/406Fast_Inverse_Square_Root.pdf <ftp://lsec.cc.ac.cn/pub/dyh/papers/bfgs-example.pdf> <http://libelemental.org/documentation/element-dev.pdf> <http://www.utim.edu.mx/raycv/mater>>.
- [Stewart 2014]STEWART, J. *MOPAC web site*. 2014. Disponível em: <www.openmopac.net>.
- [Stewart 1990]STEWART, J. J. P. MOPAC: A semiempirical molecular orbital program. *Journal of Computer-Aided Molecular Design*, v. 4, p. 1–103, 1990. ISSN 0920654X.
- [Thiel 2014]THIEL, W. Semiempirical quantum-chemical methods. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, v. 4, n. 2, p. 145–157, 2014. ISSN 17590876.

- [Thiel e Voityuk 1992]THIEL, W.; VOITYUK, A. A. Extension of the MNDO formalism to d orbitals: Integral approximations and preliminary numerical results. *Theoretica Chimica Acta*, v. 81, n. 6, p. 391–404, 1992. ISSN 00405744.
- [Tomov et al. 2010]TOMOV, S. et al. Dense linear algebra solvers for multicore with GPU accelerators. In: *Proceedings of the 2010 IEEE International Symposium on Parallel and Distributed Processing, Workshops and Phd Forum, IPDPSW 2010*. [S.l.: s.n.], 2010. ISBN 9781424465347. ISSN 1878-3449.
- [Valiev et al. 2010]VALIEV, M. et al. NWChem: A comprehensive and scalable open-source solution for large scale molecular simulations. *Computer Physics Communications*, v. 181, n. 9, p. 1477–1489, 2010. ISSN 00104655.
- [Warshel e Levitt 1976]WARSHEL, A.; LEVITT, M. Theoretical studies of enzymic reactions: Dielectric, electrostatic and steric stabilization of the carbonium ion in the reaction of lysozyme. *Journal of Molecular Biology*, v. 103, n. 2, p. 227–249, 1976. ISSN 00222836.
- [Whaley e Dongarra 1998]WHALEY, R. C.; DONGARRA, J. J. Automatically tuned linear algebra software. *Proceedings of the 1998 ACM/IEEE conference on Supercomputing*, p. 1–27, 1998. ISSN 14466368.