Universidade Estadual de Campinas
Instituto de Matemática Estatística e Computação Científica
Departamento de Matemática Aplicada

Viabilidade em Programação Não-Linear: Restauração e Aplicações

Juliano de Bem Francisco[†]
Doutorado em Matemática Aplicada - Campinas - SP

Orientador: Prof. Dr. José Mario Martínez

[†]Este trabalho teve apoio financeiro da FAPESP.

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP

Bibliotecário: Maria Júlia Milani Rodrigues – CRB8a. / 2116

Francisco, Juliano de Bem

F847v Viabilidade em programação não-linear: Restauração e aplicações / Juliano de Bem Francisco -- Campinas, [S.P. : s.n.], 2005.

Orientador: José Mario Martínez

Tese (doutorado) - Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1.Otimização. 2. Programação não-linear. 3. Restauração. 4. Sistemas não-lineares. 5. Estruturas eletrônicas. I. Martínez, José Mario. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.

Título em inglês: Nonlinear programming feasibility: Restoration and applications.

Palavras-chave em inglês (keywords): 1. Optimization; 2. Nonlinear programming; 3. Restoration; 4. Nonlinear systems; 5. Electronic structure.

Área de concentração: Otimização

Titulação: Doutorado em Matemática Aplicada

Banca examinadora: Prof. Dr. José Mario Martínez (UNICAMP)
Profa. Dra. Sandra Santos (UNICAMP)
Prof. Dr. Rogério Custodio (UNICAMP)
Prof. Dr. Clóvis Caesar Gonzaga (UFSC)
Mário Cesar Zambaldi (UFSC)

Data da defesa: 10/02/2005

Viabilidade em Programação Não-Linear: Restauração e Aplicações

Este exemplar corresponde à redação final da tese devidamente corrigida e defendida por **Juliano de Bem Francisco** e aprovada pela comissão julgadora.

Campinas, 10 de fevereiro de 2005.

Prof. Dr. **José Mario Martínez**
Orientador

Banca examinadora:
Prof. Dr. José Mario Martínez (IMECC/UNICAMP)
Prof. Dra. Sandra Santos (IMECC/UNICAMP)
Prof. Dr. Rogério Custodio (IQ/UNICAMP)
Prof. Dr. Clóvis Caesar Gonzaga (MTM/UFSC)
Prof. Dr. Mário César Zambaldi (MTM/UFSC)

Tese apresentada ao Instituto de Matemática Estatística e Computação Científica, UNICAMP, como requisito parcial para a obtenção do título de **Doutor em Matemática Aplicada**.

Tese de Doutorado defendida em 10 de fevereiro de 2005 e aprovada

Pela Banca Examinadora composta pelos Profs. Drs.

Prof. (a). Dr (a). JOSÉ MÁRIO MARTINEZ PEREZ

Prof. (a). Dr (a). ROGÉRIO CUSTÓDIO

Prof. (a). Dr (a). CLÓVIS CAESAR GONZAGA

Prof. (a). Dr (a). MÁRIO CÉSAR ZAMBALDI

Prof. (a) Dr. (a) SANDRA AUGUSTA SANTOS

Aos meus pais, Agileu e Ivonete.
À minha avó, Lorena (em memória)
A Deus.

Agradecimentos

Ao meu professor e amigo José Mario Martínez, que, além de orientar com dedicação esta tese de doutorado, contribuiu expressivamente para minha formação como pesquisador e principalmente como ser humano.

Aos meus pais e meu irmão, por terem me ajudado e contribuído até o presente momento, sendo eles os principais incentivadores e responsáveis pela minha trajetória acadêmica e principalmente pessoal.

À Carina Scandolara, que sempre esteve do meu lado e me ajudou nas horas que mais precisei.

Ao meu amigo Leandro Martínez, por estar sempre disposto a ajudar, sanando, sempre que possível, minhas dúvidas em química teórica. Sem a ajuda dele, a implementação numérica colocada na parte final deste trabalho seria praticamente impossível.

Aos meus amigos e companheiros do IMECC (Mário Salvatierra, João de Deus, Marcos Salvatierra, Marcos Eduardo, João Chela, Dirceu Bagio, Fábio Dorini, Flávio Yano, Laura, Raul, Ricardo, entre outros), que sempre estiveram prontos a ajudar.

À Mario César Zambaldi, que desde que o conheci sempre procurou ajudar e incentivar a transpor as dificuldades.

Aos professores do IMECC/UNICAMP, que contribuíram significativamente para a complementação da minha formação como matemático aplicado.

Agradeço à Fundação de Amparo à Pesquisa do Estado de São Paulo, FAPESP, por fomentar este projeto até o presente momento, tendo um papel fundamental para a realização desta tese de doutorado.

A todos os meus amigos não citados aqui, que me incentivaram e ajudaram, de alguma forma, para a obtenção do título de Doutor.

Para finalizar, agradeço a todas as pessoas que, direta ou indiretamente, tornaram possível a conclusão desta tese de doutorado.

A todos, meu muito obrigado!

Resumo

Algoritmos robustos e numericamente viáveis para resolver problemas de otimização têm sido cada vez mais solicitados em problemas práticos que aparecem em engenharia, química, física, entre outras áreas.

Com isso em mente, este trabalho apresenta um novo método globalmente convergente baseado em região de confiança para resolver sistemas não-lineares indeterminados (mais incógnitas do que equações) com restrições de caixa, podendo, portanto, ser aproveitado para a fase de viabilidade nos algoritmos baseados em restauração periódica. É mostrado que esse método apresenta, sob certas hipóteses, convergência localmente quadrática.

Em uma outra parte deste trabalho é apresentado um novo algoritmo globalmente convergente, o qual se baseia em região de confiança, para resolver problemas de otimização do tipo

$$\min f(x), \quad s.a.\ x \in D,$$

onde $f : \mathbb{R}^n \to \mathbb{R}$ é assumida para ser continuamente diferenciável e $D \subseteq \mathbb{R}^n$, um subconjunto fechado arbitrário. Em vez de considerar a região de confiança explicitamente nos subproblemas, esse método introduz um parâmetro de regularização que busca imitar a região de confiança. Com essa caracterização, os subproblemas consistem em minimizar um modelo quadrático de f sujeito ao subconjunto D.

Uma importante aplicação desse novo algoritmo aparece em química quântica e resultará em um novo algoritmo globalmente convergente, robusto e numericamente viável para calcular estruturas eletrônicas de átomos e moléculas.

Abstract

Robust and numerically feasible algorithms for solving optimization problems have been demanded for solving practice problems that appear in Engineering, Chemistry, Physics and others.

This work present a new globally convergent method based on trust regions for solving box-constrained underdetermined nonlinear systems (more unknowns than equations), that can be used on the feasibility fase of algorithms based on periodic restoration. Under some assumptions, it will be proved locally quadratic convergence.

In other part of this work, a new globally convergent algorithm is introduced, based on trust regions, for solving the optimization problem

$$\min f(x), \quad s.t.\ x \in D,$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and $D \subseteq \mathbb{R}^n$ is an arbitrary closed subset. Instead of considering explicitly the trust region on the subproblems, the method introduces a regularization parameter that mimics the trust region. With this characterization, the subproblems consist on minimizing a quadratic model of f subject to D.

An important application of this new algorithm is on Chemistry, where a robust and numerically feasible globally convergent algorithm for electronic structure calculations is obtained.

Sumário

Introdução

Com o avanço dos computadores, principalmente em relação à velocidade de processamento e à capacidade de armazenamento, muitos problemas de engenharia, química, economia, entre outros, passaram a ser resolvidos de maneira satisfatória. Entretanto, somente os avanços tecnológicos não bastam para que se consiga tal façanha. A escolha do método para resolver um determinado problema é fundamental para que se obtenham soluções satisfatórias.

Como a maioria desses problemas práticos envolve de alguma maneira técnicas de otimização, desenvolver novos algoritmos para problemas particulares de otimização tem se tornado uma atividade cada vez mais freqüente. Com isso, muitos problemas têm sido resolvidos de maneira eficaz e confiável.

O objetivo deste trabalho se enquadra nesse contexto.

Na primeira parte é introduzido um novo algoritmo de pontos interiores para resolver sistemas não-lineares indeterminados com restrição de caixa. Isto porque o conjunto viável de problemas de programação não-linear é geralmente representado por um conjunto de equações sujeito a variáveis canalizadas.

Muitos métodos tradicionais [1, 5, 17, 54] e modernos [38, 39] de otimização exploram essa estrutura e requerem algoritmos específicos para restaurar a viabilidade em toda a iteração. Geralmente, a região viável não é vazia e existem pontos que são interiores com relação às variáveis canalizadas. Portanto, é importante desenvolver procedimentos que resolvam sistemas não-lineares indeterminados sujeitos a restrições de caixa de uma maneira mais eficiente do que os tradicionais métodos para programação não-linear fazem.

Com a finalidade de resolver sistemas não-lineares quadrados com restrições de caixa, Bellavia, Macconi e Morini [6, 7] desenvolveram um algoritmo que adapta para sistemas de equações as idéias do método de otimização com restrições de caixa de Coleman e Li [15].

Neste trabalho é modificado e estendido o algoritmo de Bellavia, Macconi e Morini para o caso indeterminado. Assim, o problema é resolver $F(x) = 0$, com $x \in \Omega$, sendo $F : \mathbb{R}^n \to \mathbb{R}^m$ $(m \leq n)$ e $\Omega \subseteq \mathbb{R}^n$ o conjunto viável gerado pelas restrições canalizadas.

A resolução de sistemas não-lineares indeterminados tem sido endereçada aos métodos

que usam abordagens quasi-Newton [36, 62] e Levenberg-Marquardt [18]. Algumas dessas idéias são incorporadas no presente trabalho. Em particular, a direção de norma mínima de Newton (Newton-pseudoinversa) é tomada como um ponto candidato sempre que possível. Direções de norma mínima quasi-Newton têm sido usadas em outros trabalhos [36, 62].

Métodos para resolver sistemas baseados na pseudoinversa podem ser globalizados usando-se procedimento de região de confiança usual ou de busca linear. Entretanto, quando restrições canalizadas estão presentes, regiões de confiança euclidianas não são adequadas. Na prática, as regiões de confiança afim-escala introduzidas por Coleman e Li [15] são capazes de manipular a caixa de uma maneira mais eficiente. Quando o ponto atual está próximo da fronteira, mas não da solução, a estratégia de Coleman e Li geralmente força um passo maior, uma característica fundamental para um eficiente e prático método de otimização.

O método proposto neste trabalho é comparado com o recente método introduzido por Fukushima, Kanzow e Yamashita [32].

É provada para o método convergência global a pontos estacionários da norma ao quadrado do sistema. Também, como a filosofia é usar a direção newtoniana sempre que possível, é provado sob hipóteses usuais para esse tipo de problema que a convergência é localmente quadrática sempre que o ponto-limite se encontrar no interior da caixa.

Em outra parte deste trabalho é introduzido um novo método globalmente convergente para minimizar uma função diferenciável f restrita a um conjunto arbitrário fechado D.

O algoritmo proposto está baseado, até certo ponto, em métodos de regiões de confiança [40] e nas idéias do método de Levenberg-Marquardt [45], resolvendo o problema de programação não-linear por meio de uma seqüência de subproblemas quadráticos restritos ao conjunto viável D. Entretanto, o ponto-chave do método é a troca da região de confiança por um parâmetro de regularização que penaliza a restrição associada à região de confiança no modelo quadrático dos subproblemas envolvidos. Essa mudança simplifica, na maioria dos casos, os subproblemas envolvidos, pois, geralmente, resolver o modelo quadrático na intersecção de D com a região de confiança é uma tarefa difícil.

As iterações principais usam um modelo quadrático da função objetivo f em torno do iterando atual e minimizam este modelo sujeito ao conjunto D, obtendo um ponto candidato. Então, se este ponto reduzir a função objetivo suficientemente, ou seja, a redução em f é pelo menos uma fração da redução no modelo, este ponto candidato é aceito, mas, se esse decréscimo não for suficiente, o parâmetro de regularização é aumentado, obtendo-se um novo ponto candidato. Felizmente, aumentar o parâmetro de regularização tem o mesmo efeito que reduzir a região de confiança. Esse processo é repetido até que um decréscimo suficiente seja obtido em f.

Sob hipóteses razoavelmente fracas, será provado que todo ponto de acumulação da

seqüência gerada pelo método é um ponto estacionário do problema de programação não-linear em questão.

A principal aplicação desse algoritmo está no cálculo de estruturas eletrônicas, onde, neste caso, f representa a função energia e D um conjunto de restrições de ortonormalidade, o que resulta em um algoritmo globalmente convergente para esse tipo de problema.

Cálculos eletrônicos estão sendo cada vez mais usados em atividades de pesquisa. Vários pacotes computacionais estão disponíveis, os quais fornecem uma grande variedade de algoritmos e ferramentas analíticas, de maneira que não é necessário um entendimento completo dos métodos para se obterem resultados significativos. Para isso acontecer, é necessário que os algoritmos envolvidos se tornem mais rápidos, independentes do usuário e confiáveis.

O primeiro método designado para resolver o problema do cálculo de estruturas eletrônicas foi baseado em uma simples iteração de ponto fixo, que consiste na construção de uma matriz, conhecida como Matriz de Fock, a partir de uma aproximação inicial seguida pela sua diagonalização, obtendo-se um novo iterado. Por se tratar de uma iteração de ponto fixo, esse método tem convergência lenta e instável. Por isso, não é muito utilizado para fins práticos. Entretanto, vários métodos práticos ainda confiam na iteração de ponto fixo de alguma maneira.

Uma característica importante do algoritmo proposto neste trabalho para calcular estruturas eletrônicas é que os subproblemas consistem na resolução de problemas de autovalores.

Como o método incorpora a iteração de ponto fixo na primeira iteração, ele pode ser interpretado como uma globalização deste, embora outros algoritmos sem convergência global pudessem também ser incorporados, de modo a globalizá-los.

O ponto-chave desse algoritmo é que, depois de cada iteração principal, um procedimento de aceleração é admissível, com a única condição de que não aumente a função objetivo. No caso do cálculo de estruturas eletrônicas, foi usada nos experimentos a aceleração DIIS, bem conhecida pela comunidade de químicos teóricos, mas qualquer outro procedimento de aceleração é admitido. Nesse sentido, o algoritmo proposto para o cálculo eletrônico pode também ser interpretado como uma maneira simples e confiável de fornecer convergência global para métodos que são conhecidos por serem eficientes em muitos casos.

Este trabalho está organizado como se segue. No primeiro capítulo são colocados alguns conceitos e as principais idéias dos algoritmos que envolvem processos de restauração. No Capítulo 2 é introduzido o novo algoritmo para sistemas não-lineares indeterminados com restrições de caixa, o qual pode ser aplicado na fase de viabilidade dos algoritmos que envolvem estratégias de restauração. No mesmo capítulo são provadas as propriedades teóricas do método e são mostrados alguns experimentos. No Capítulo 3 é introduzido

um novo algoritmo para problemas de programação não-linear, sendo o conjunto viável um conjunto fechado arbitrário. A análise de convergência é também colocada nesse capítulo. Considerando a aplicação do método do Capítulo 3, são colocadas no Capítulo 4 as idéias principais do problema do cálculo eletrônico, sendo mostrada matematicamente toda a teoria envolvida e, portanto, facilitando-se o entendimento para a comunidade matemática. No Capítulo 5 são colocadas, também com rigor matemático, as idéias dos principais e mais usados métodos para calcular estruturas eletrônicas. No Capítulo 6 é introduzido o novo algoritmo para o cálculo eletrônico, sendo mostrados também os resultados numéricos. Nestes experimentos o método proposto foi confrontado com outros algoritmos bem conhecidos para resolver o problema. Para finalizar, são apresentadas no Capítulo 7 as principais conclusões deste trabalho.

Capítulo 1

A Restauração em Programação Não-Linear

Um problema clássico em otimização numérica é a minimização de uma função contínua sujeita a restrições dadas por equações e inequações, em geral, não-lineares, ou seja,

$$\begin{aligned} \min \quad & f(x) \\ s.a. \quad & h_1(x) = 0 \\ & h_2(x) \leq 0, \end{aligned} \tag{1.1}$$

onde $f : \mathbb{R}^n \to \mathbb{R}$, $h_1 : \mathbb{R}^n \to \mathbb{R}^{m_1}$ e $h_2 : \mathbb{R}^n \to \mathbb{R}^{m_2}$ são, geralmente, funções continuamente diferenciáveis. Esse tipo de problema é conhecido como **problema de programação não-linear** (PNL).

Introduzindo variáveis de folga no problema (1.1), este fica equivalente a:

$$\begin{aligned} \min \quad & f(x) \\ s.a. \quad & h(x) = 0 \\ & x \in \Omega, \end{aligned} \tag{1.2}$$

onde $h : \mathbb{R}^n \to \mathbb{R}^m$ e Ω é um conjunto fechado e convexo.

Umas das idéias mais naturais para resolver problemas dessa natureza é gerar uma seqüência $\{x^k\}_{k\in\mathbb{N}}$ que satisfaz as restrições de modo que a função objetivo f é progressivamente decrescente. Dessa maneira, é esperado que, no limite, uma solução para o problema seja obtida.

Métodos viáveis para resolver problemas de programação não-linear têm considerável interesse em problemas práticos devido ao fato de que, freqüentemente, as restrições envolvem definições ou leis físicas que não podem ser violadas, e a função objetivo está relacionada a algum custo, para o qual se desejam pequenos valores, mas não necessariamente

o menor possível. Portanto, muitas vezes, as soluções viáveis não ótimas são bem-vindas nas aplicações, ao contrário das soluções não viáveis, que não têm aplicabilidade prática.

Entretanto, manter a viabilidade dos iterados quando as restrições são muito não-lineares é uma tarefa difícil. Portanto, quando esse tipo de restrições está presente, é necessário considerar a perda de viabilidade e sua restauração constante. Esse processo de recuperar a viabilidade a partir de um ponto inviável é conhecido como **restauração**.

Geralmente, o processo de restauração envolve a resolução de um sistema não-linear indeterminado sujeito a algum conjunto Ω fechado e convexo, muitas vezes dado por restrições canalizadas. Assim, na restauração, é preciso encontrar x tal que

$$h(x) = 0 \quad \text{e} \quad x \in \Omega. \tag{1.3}$$

Entre os métodos de restauração mais tradicionais, pode-se citar o método de gradiente projetado [54], o de gradiente reduzido generalizado (GRG) [1] e o algoritmo do gradiente com restauração seqüencial (SGRA) [5, 17] com suas variações. A principal diferença entre esses métodos é a maneira com que a restauração é realizada, seguindo roteiros específicos; por exemplo, caminhos ortogonais ou caminhos que consideram apenas as variáveis básicas.

A idéia dos métodos de restauração é decompor a iteração em duas fases, uma relacionada à otimalidade no subespaço tangente e a outra relacionada à viabilidade.

Em resumo, esses métodos consistem em, a partir de um ponto viável x^k, obter um ponto candidato y satisfazendo uma aproximação linear das restrições tal que a função objetivo, o Lagrangeano, avaliada em y seja suficientemente menor do que o correspondente valor em x^k. Como, em geral, y não é viável, um processo de restauração é necessário para obter um ponto (quase) viável z. Se z satisfaz determinada condição de decréscimo suficiente, então o próximo iterado é definido para ser z; caso contrário, a distância entre x^k e y deve ser reduzida. A Figura 1.1 ilustra um passo dos algoritmos de restauração.

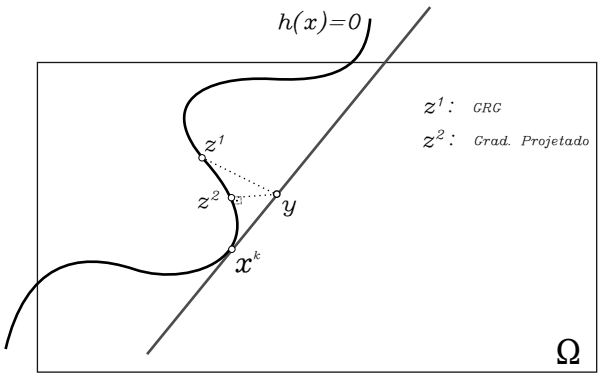

Figura 1.1: A idéia dos algoritmos de restauração para o caso de restrição de igualdade

Uma desvantagem bem conhecida dos métodos viáveis e, conseqüentemente, dos de

restauração é a deficiência para resolver problemas em que as restrições são fortemente não-lineares, o que deve causar passos muito pequenos no algoritmo longe da solução, prejudicando a convergência.

A metodologia de **restauração inexata** procura contornar esse inconveniente relaxando as restrições quando o iterando estiver longe da solução e, então, gradualmente diminui a inviabilidade à medida que o método avança. Dessa maneira, passos suficientemente grandes são permitidos no começo do processo.

1.1 As idéias da restauração inexata

Os métodos de restauração inexata foram recentemente introduzidos [38, 39] para resolver problemas de otimização restrita.

Enquanto nos algoritmos tradicionais de restauração os pontos correntes são (quase) viáveis e os pontos intermediários são os obtidos no processo de minimização no subespaço tangente, nos algoritmos de restauração inexata os pontos inviáveis restaurados inexatamente (y) são intermediários, e os pontos correntes (z), quase sempre inviáveis, são os iterados gerados num processo de minimização de uma função, geralmente a função lagrangeana, nas aproximações tangentes das restrições. A Figura 1.2 ilustra a idéia da restauração inexata. Uma abordagem automática garante que o peso da inviabilidade aumenta à medida que o método se aproxima de uma solução, e, portanto, nas últimas iterações, o algoritmo preserva viabilidade.

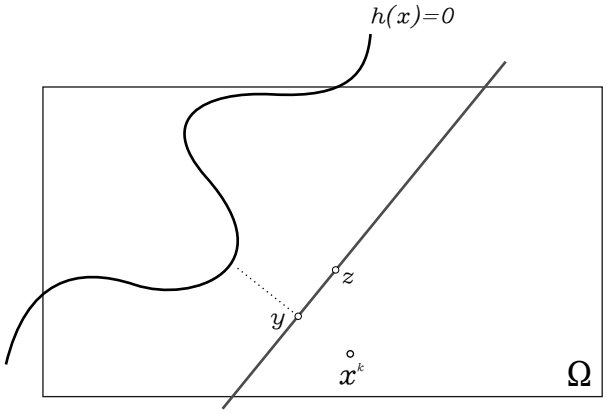

Figura 1.2: A idéia da restauração inexata para o caso de restrição de igualdade

Dessa maneira, para problemas como em (1.2), a fase de viabilidade dos algoritmos de restauração inexata envolve a escolha de um método para resolver problemas como em (1.3). Já a fase de otimalidade envolve a escolha de um método para minimizar a função lagrangeana sujeita a restrições lineares (subespaço tangente das restrições) e a $x \in \Omega$.

O efeito positivo de usar o Lagrangeano na fase de otimalidade vem do fato de que, quando restrições não-lineares estão presentes, o seu comportamento no subespaço tangente imita o comportamento da função objetivo na região viável.

Uma grande vantagem dos métodos de restauração e restauração inexata é a liberdade de escolher o algoritmo em ambas as fases do método, principalmente para a fase de restauração. Assim, características específicas do problema podem ser exploradas e grandes problemas podem ser resolvidos usando-se subalgoritmos apropriados, o que não seria possível para a maioria dos algoritmos de programação não-linear.

Entre as aplicações reais em que mecanismos específicos de restauração podem ser usados estão:

- problemas de dois níveis: as restrições são um problema de minimização, portanto a forma natural de restaurar é resolver (parcialmente) tais problemas;

- problemas com restrições de equilíbrio: quando as restrições são do tipo equilíbrio de Nash, o natural é restaurar resolvendo cíclica ou paralelamente os subproblemas de minimizações envolvidos nas restrições;

- problemas em que a restrição é a resolução de um sistema de equações diferenciais, geralmente uma equação diferencial parcial; e

- problemas de empacotamento de moléculas e outros problemas de química em que buscam recuperar certa localização de átomos, ligados por uma estrutura molecular fixa. Esse problema pode ser generalizado a outros problemas de empacotamento não-linear, não ligados com a química.

Por permitir uma liberdade bastante ampla na escolha do método de restauração é que os algoritmos de restauração inexata podem ser direcionados para esses problemas, em que existe uma maneira natural de restauração, além de fornecer teorias de convergência abrangentes para métodos nelas baseados.

No capítulo que se segue é introduzido um novo método globalmente convergente para resolver sistemas não-lineares indeterminados como em (1.3), podendo, portanto, ser aproveitado para recuperar a viabilidade nos algoritmos que envolvam algum tipo de restauração, em especial os algoritmos de restauração inexata.

Capítulo 2

Algoritmo Newtoniano com Região de Confiança para Restauração

Neste capítulo é introduzido um novo algoritmo de pontos interiores para resolução de sistemas não-lineares indeterminados (SNLI) com restrições de caixa, também conhecidas como restrições canalizadas. Esse algoritmo, globalmente convergente, apresenta, sob certas condições, taxa de convergência localmente quadrática.

O objetivo é investigar o caso em que o sistema não-linear apresenta mais incógnitas (n) do que equações (m). Neste caso, o problema consiste em encontrar pontos na intersecção do conjunto viável (caixa) com uma variedade não-linear de $\mathbb{R}^n$. Problemas desse tipo chegam de numerosas aplicações da vida real, embora o método proposto tenha sido desenvolvido para restaurar a viabilidade nos tradicionais [5, 17] e modernos métodos práticos de otimização, em especial os algoritmos de **restauração inexata** [38, 39].

Nas seções que se seguem são apresentadas as principais características do método. Uma seção foi reservada exclusivamente para expor as propriedades de convergência. Ao final, alguns experimentos computacionais são expostos para mostrar o desempenho do método. Esses experimentos foram realizados tomando conjuntos viáveis de alguns problemas de programação não-linear encontrados na literatura [28, 18].

2.1 Considerações gerais

Nesta seção são colocadas algumas considerações e definições usadas ao longo deste capítulo. O problema geral é definido e alguns comentários relevantes são feitos.

Para começar, considere o conjunto viável Ω (caixa) dado por

$$\Omega = \{x \in \mathbb{R}^n \mid l \leq x \leq u\}, \tag{2.1}$$

onde $l < u$ são os vetores contendo, respectivamente, os limitantes inferiores e superiores satisfazendo $l_i \in \mathbb{R} \cup \{-\infty\}$ e $u_i \in \mathbb{R} \cup \{\infty\}$ para $i = 1, \ldots, n$.

Sendo assim, o problema considerado neste capítulo consiste em encontrar $x \in \Omega$ tal que

$$F(x) = 0, \tag{2.2}$$

onde $F : \Omega \to \mathbb{R}^m$ $(m \leq n)$ é uma aplicação continuamente diferenciável em um conjunto aberto de $\mathbb{R}^n$ contendo Ω. O caso $m = n$ é investigado em vários trabalhos [6, 32, 33, 37]. Na maioria deles, os principais resultados supõem a não-singularidade do Jacobiano de F.

Uma maneira simples de resolver um problema do tipo (2.2) seria transformá-lo em um problema de programação não-linear (PNL) tomando como função objetivo o quadrado da norma euclidiana de F, ou seja, $\|F\|^2$. Assim, uma infinidade de métodos poderia ser usada [20, 45], porém tais abordagens não levariam em conta diferenças e particularidades estruturais entre o problema (2.2) e PNL de maneira geral.

Sistemas não-lineares indeterminados sem restrições podem ser resolvidos por métodos de Newton baseados na pseudo-inversa do Jacobiano, métodos quasi-Newton [36, 62] ou por Levenberg-Marquardt [18]. Porém, a existência de limitantes (caixa) implica que nem todas as soluções serão completamente admissíveis. Mais ainda, em alguns casos, as funções envolvidas estão definidas somente para pontos viáveis. Portanto, cuidados adicionais devem ser considerados para que todos os iterandos se mantenham estritamente viáveis.

O algoritmo introduzido neste capítulo está baseado nos recentes artigos de Bellavia, Macconi e Morini [6, 7] e de Coleman e Li [14, 15]. O algoritmo incorpora o método de Newton de norma mínima, com abordagem de região de confiança baseada no recente método de pontos interiores afim-escala para problemas de otimização restrita [15]. A estratégia de Coleman e Li deriva em subproblemas de região de confiança elípticos cuja geometria depende dos limitantes l e u. A finalidade dessa estratégia é permitir passos maiores dentro da região viável Ω. Em cada iteração do método proposto é requerida uma solução aproximada de um subproblema de minimização quadrática com restrições elípticas, como nas abordagens usuais de região de confiança para problemas irrestritos. Um fato relevante é que esse subproblema não leva explicitamente em consideração as restrições de caixa, mas apenas restrições de norma euclidiana.

Em resumo, o método introduzido por Coleman e Li [15] consiste em usar uma abordagem de região de confiança elíptica para minimizar uma função não-linear sujeita a restrições de caixa, gerando iterandos estritamente viáveis. Os subproblemas de região de confiança não requerem explicitamente as restrições de caixa na sua formulação. As regiões elípticas são definidas por uma matriz afim-escala, que também será usada neste

trabalho e apresentada mais à frente. Coleman e Li [15] introduziram dois algoritmos. Em um deles é necessário resolver exatamente os subproblemas de região de confiança. Sob algumas hipóteses, entre elas a não-singularidade da Hessiana da função objetivo, é estabelecida convergência global com taxa localmente quadrática, desde que o ponto-limite esteja no interior da caixa.

O algoritmo introduzido por Bellavia, Macconi e Morini [6] pode ser visto como uma adaptação do primeiro algoritmo de Coleman e Li [15] para sistemas não-lineares quadrados ($m = n$) com restrições de caixa, aproveitando, portanto, a estrutura do problema. Sendo assim, o método combina a abordagem de região de confiança para sistemas não-lineares com as idéias de Coleman e Li; com isso, as regiões de confiança são elípticas e suas geometrias são definidas pela mesma matriz afim-escala citada anteriormente [15]. Embora o método sugira obter a solução exata do subproblema de região de confiança, os experimentos numéricos mostraram um bom desempenho usando o método **Dog-Leg** [45] para encontrar uma solução aproximada. Da mesma maneira que o caso anterior, o algoritmo apresenta sob certas hipóteses, entre elas a de não-singularidade do Jacobiano de F e viabilidade estrita da solução, convergência global e taxa localmente quadrática.

Como mencionado, diferentemente de outras abordagens [6, 15], o método proposto neste capítulo se destina a resolver sistemas não-lineares indeterminados, problemas do tipo (2.2) com $m \leq n$, podendo, portanto, ser visto como uma extensão do método de Bellavia, Macconi e Morini [6]. A abordagem está baseada nas idéias colocadas acima, porém algumas diferenças devem ser evidenciadas. No algoritmo aqui introduzido é usada a estratégia de um raio de confiança mínimo e são exigidas apenas aproximações para os subproblemas de região de confiança. Uma dificuldade natural do problema é a degenerescência do Jacobiano. Aqui é exigido que este apresente posto completo m. Isso implica outras argumentações para obter os resultados de convergência desejados.

Em uma seção de resultados numéricos, o método proposto por este capítulo é confrontado com a versão globalizada do método introduzido por Kanzow, Yamashita e Fukushima [32, Algoritmo 3.12], embora este tenha sido desenvolvido com o objetivo de resolver problemas mais gerais do que o considerado neste capítulo. O algoritmo de Kanzow, Yamashita e Fukushima incorpora o método Levenberg-Marquardt projetado com o método de gradiente projetado, sendo este último utilizado para a sua globalização. Assim, primeiro, um passo Levenberg-Marquardt projetado é tentado e, se este não apresentar uma decréscimo suficiente em $\|F\|$, o passo do gradiente projetado é desempenhado de forma que o decréscimo seja alcançado. Sob certas condições, entre elas a de "*error bound*", é obtida taxa de convergência localmente quadrática.

São colocadas a seguir algumas notações usadas ao longo deste capítulo e, quando necessário, algumas notações podem ser introduzidas no decorrer do texto.

Para qualquer matriz A, a notação $A^\dagger$ denota sua pseudo-inversa.

Dada uma aplicação G, a notação G_k será usada para denotar $G(x^k)$.

Para qualquer $x \in \mathbb{R}^n$, a norma euclidiana será denotada por $\|x\|$ e a p-norma por $\|x\|_p$.

O k-ésimo termo de uma seqüência será representado por x^k, e para representar o i-ésimo componente de um vetor x será usado $[x]_i$. Algumas vezes, quando claro no contexto, será usado x_i para esse propósito.

Seguindo a notação do Matlab [46], para qualquer $v \in \mathbb{R}^n$, diag(v) representará a matriz diagonal $n \times n$ com o vetor v definindo as entradas da diagonal em sua ordem natural.

A bola aberta de raio ρ e centro em y será denotada por $\mathcal{B}(y, \rho) = \{x \mid \|x - y\| < \rho\}$.

Dada uma caixa $\Omega = \{x \in \mathbb{R}^n \mid l \leq x \leq u\}$, int$(\Omega)$ indicará o interior de Ω.

2.2 Descrição do método

Nesta seção é apresentado o método proposto. Como será visto, o método irá gerar iterandos estritamente viáveis, o que torna o método atrativo para problemas em que a aplicação F é definida somente em Ω.

É definida uma função de mérito, a qual medirá a otimalidade do iterando atual, dizendo se este será aceito ou rejeitado. Problemas do tipo (2.2) sugerem uma função de mérito natural, $f : \mathbb{R}^n \to \mathbb{R}$, dada por

$$f(x) = \frac{1}{2}\|F(x)\|^2. \tag{2.3}$$

Ao longo deste capítulo, $J(x)$ irá denotar o Jacobiano de F avaliado em x. Assim,

$$\nabla f(x) = J(x)^T F(x).$$

Usando a abordagem usual para sistemas não-lineares, o *subproblema de região de confiança* em torno de um iterado x^k será definido usando o modelo quadrático

$$\begin{aligned} m_k(p) &= \frac{1}{2}\|J(x^k)p + F(x^k)\|^2 \\ &= \frac{1}{2}\|F(x^k)\|^2 + F(x^k)^T J(x^k)p + \frac{1}{2}p^T J(x^k)^T J(x^k)p. \end{aligned} \tag{2.4}$$

Portanto, na k-ésima iteração, o subproblema de região de confiança será

$$\begin{array}{ll} \min & m_k(p) \\ s.a. & \|D_k p\| \leq \Delta, \end{array} \tag{2.5}$$

onde $\Delta > 0$ é o raio de confiança e $D_k \equiv D(x^k)$ é a matriz afim-escala [14, 15], a qual será colocada na seqüência. Existem várias abordagens para resolver o subproblema (2.5) [24, 43, 45, 59]. No método aqui introduzido, o subproblema de região de confiança será resolvido apenas aproximadamente.

Dado $x \in \text{int}(\Omega)$, a matriz afim-escala $D(x)$ é definida por

$$\begin{aligned} D(x) &= \begin{bmatrix} |v_1(x)|^{-1/2} & & \\ & \ddots & \\ & & |v_n(x)|^{-1/2} \end{bmatrix} \\ &= diag(|v_1(x)|^{-1/2}, \ldots, |v_n(x)|^{-1/2}), \end{aligned} \tag{2.6}$$

onde

$$v_i(x) = \begin{cases} x_i - u_i, & \text{se } \nabla f(x)_i < 0 \text{ e } u_i < \infty \\ x_i - l_i, & \text{se } \nabla f(x)_i \geq 0 \text{ e } l_i > -\infty \\ -1, & \text{se } \nabla f(x)_i < 0 \text{ e } u_i = \infty \\ 1, & \text{se } \nabla f(x)_i \geq 0 \text{ e } l_i = -\infty. \end{cases} \tag{2.7}$$

É importante perceber que $D(x)$ não está definida na fronteira de Ω, mas $D(x)^{-1}$ pode ser continuamente estendida para pontos dessa fronteira, sendo esta extensão também denotada por $D(x)^{-1}$,

$$D(x)^{-1} = \begin{bmatrix} |v_1(x)|^{1/2} & & \\ & \ddots & \\ & & |v_n(x)|^{1/2} \end{bmatrix},$$

sendo a função $v_i(x)$ dada por (2.7).

Uma observação importante é que, como $D(x)^{-1}$ e $\nabla f(x)$ são aplicações contínuas, segue que $D(x)^{-1}\nabla f(x)$ também é uma aplicação contínua.

Considerando o problema de otimização

$$\begin{array}{ll} \min & f(x) \\ s.a. & x \in \Omega, \end{array} \tag{2.8}$$

suas condições necessárias de primeira ordem são

$$\begin{cases} \nabla f(x)_i = 0 & \text{se} \quad l_i < x_i < u_i \\ \nabla f(x)_i \leq 0 & \text{se} \quad x_i = u_i \\ \nabla f(x)_i \geq 0 & \text{se} \quad x_i = l_i. \end{cases} \tag{2.9}$$

Pontos que satisfazem essas condições serão também chamados de **pontos estacionários**. Com essas considerações, o seguinte lema (retirado de [15, Lema 2.3]), pode ser provado.

Lema 2.1. *Seja $x \in \Omega$. Então, $D(x)^{-1}\nabla f(x) = 0$ se e somente se a condição (2.9) é satisfeita.*

Prova. Pela definição (2.6) da matriz D, resulta que

$$D(x)^{-1}\nabla f(x) = \begin{bmatrix} |v_1(x)|^{1/2}[\nabla f(x)]_1 \\ \vdots \\ |v_n(x)|^{1/2}[\nabla f(x)]_n \end{bmatrix}.$$

Assim, $D(x)^{-1}\nabla f(x) = 0$ se e somente se $|v_i(x)|^{1/2}[\nabla f(x)]_i = 0$ para $i = 1, \ldots, n$. Se $[\nabla f(x)]_i = 0$, então x_i satisfaz a condição (2.9). Portanto, basta analisar o caso quando $|v_i(x)| = 0$. Se $|l_i|, |u_i| = +\infty$ tem-se que $|v_i(x)| = 1$, então esta situação não precisa ser analisada. Assim, será assumido que $|l_i|, |u_i| < +\infty$.

Se um ponto $x \in \Omega$ satisfaz a condição (2.9), imediatamente, pela definição (2.7) de v, segue que $|v_i(x)| = 0$ ou $[\nabla f(x)]_i = 0$ e, portanto, $D(x)^{-1}\nabla f(x) = 0$.

Reciprocamente, suponha que $D(x)^{-1}\nabla f(x) = 0$. Assim, pela observação feita acima, basta considerar $|v_i(x)| = 0$. Pela definição de v segue que $x_i = l_i$ ou $x_i = u_i$. Se $x_i = l_i$, então $[\nabla f(x)]_i \geq 0$ e, portanto, x_i satisfaz a condição (2.9). Se $x_i = u_i$, segue que $[\nabla f(x)]_i < 0$ e x_i também satisfaz a condição (2.9). Logo, x é um ponto estacionário de (2.8), e o lema está provado. ■

Dados um iterando $x^k \in \text{int}(\Omega)$ e uma direção p, é preciso encontrar x^{k+1}, dependendo de x^k e p, de modo que seja estritamente viável. Na seqüência, o procedimento que torna isso possível é estabelecido.

Será definido

$$\lambda(p) = \arg\max\{t \geq 0 \mid x^k + tp \in \Omega\}. \tag{2.10}$$

Se $\lambda(p) \leq 1$, tem-se que $x^k + p \notin \text{int}(\Omega)$, e uma redução no passo ao longo da direção p é necessária para ficar no interior de Ω. Em caso contrário, $x^k + p \in \text{int}(\Omega)$. Para $\theta \in (0, 1)$,

uma constante fixa, será definido

$$\xi(p) = \begin{cases} 1, & \text{se } \lambda(p) > 1, \\ \max\{\theta, 1 - \|p\|\}\,\lambda(p), & \text{caso contrário.} \end{cases} \tag{2.11}$$

Dessa maneira, considerando

$$\alpha(p) = \xi(p)p \tag{2.12}$$

e definindo o próximo iterado por $x^{k+1} = x^k + \alpha(p)$, segue que este é estritamente viável, ou seja, $x^{k+1} \in \text{int}(\Omega)$.

Até o momento, foi assumida a existência de uma direção p de modo que a iteração seguinte pudesse ser definida. Resta, portanto, propor um procedimento que na k-ésima iteração encontre uma direção, a qual será chamada de p^k, de modo que reduza suficientemente o modelo quadrático (2.4) e a função de mérito f definida em (2.3).

Será definida como **direção de máxima descida afim-escala** o vetor

$$d^k = -D_k^{-2}\nabla f(x^k), \tag{2.13}$$

pelo fato de consistir da multiplicação da matriz D_k^{-2} pela direção de máxima descida, a saber, $-\nabla f_k$.

A direção de Cauchy será definida como o vetor p_C^k que minimiza o modelo m_k (2.4) ao longo da direção de máxima descida afim-escala d^k de modo que satisfaça a restrição de região de confiança. Assim,

$$p_C^k = \tau^k d^k = -\tau^k D_k^{-2}\nabla f_k, \tag{2.14}$$

com

$$\tau^k = \arg\min_{\tau>0} \left\{ m_k(\tau d^k) \mid \|\tau D_k d^k\| \leq \Delta \right\},$$

para algum raio de confiança $\Delta > 0$. Essa direção de Cauchy será usada para medir o decréscimo suficiente no modelo quadrático m_k. Vale a pena notar que $x^k + \tau^k d^k$ pode não estar em Ω, já que as restrições de caixa não são consideradas nos subproblemas de região de confiança, como ilustra a Figura 2.1.

A seguir é colocado um resultado conhecido (ver [45, p. 70]) que mostra explicitamente o parâmetro τ^k de (2.14).

Proposição 2.2. *Seja p_C^k a direção de Cauchy (2.14). Então,*

$$\tau^k = \min\left\{ \frac{\|D_k^{-1}\nabla f_k\|^2}{\|J_k D_k^{-2}\nabla f_k\|^2}, \frac{\Delta}{\|D_k^{-1}\nabla f_k\|} \right\}.$$

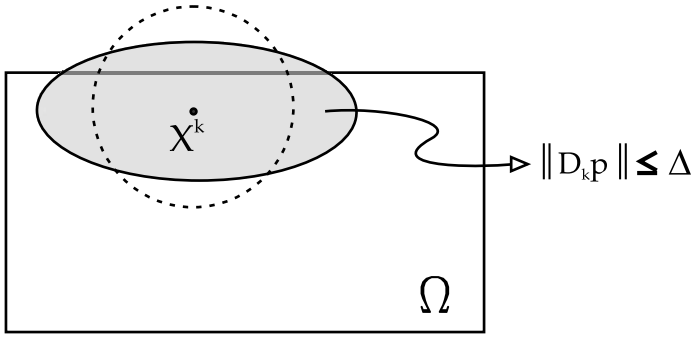

Figura 2.1: A região de confiança e as restrições de caixa nos subproblemas

Prova. Pela definição de τ^k tem-se que

$$\tau^k = \arg\min_{\tau>0} \left\{ m_k(\tau d^k) \mid \|\tau D_k d^k\| \leq \Delta \right\},$$

onde o modelo m_k é dado por (2.4) e a direção d^k por (2.13). Note que existem duas possibilidades para o problema acima. A primeira é quando $\|\tau^k D_k d^k\| = \Delta$, assim

$$\tau^k = \frac{\Delta}{\|D_k d^k\|} = \frac{\Delta}{\|D_k^{-1} \nabla f_k\|}.$$

A outra possibilidade ocorre quando $\|\tau^k D_k d^k\| < \Delta$. Neste caso, τ^k é o minimizador irrestrito do problema

$$\min_{\tau>0} m_k(\tau d^k),$$

onde, pela definição (2.4),

$$\begin{aligned} m_k(\tau d^k) &= f_k + \tau \nabla f_k d^k + \frac{\tau^2}{2} d^k J_k^T J_k d^k \\ &= f_k - \tau \|D_k^{-1} \nabla f_k\|^2 + \frac{\tau^2}{2} \|J_k D^{-2} \nabla f_k\|^2. \end{aligned}$$

Portanto, para este caso,

$$\tau^k = \frac{\|D_k^{-1} \nabla f_k\|^2}{\|J_k D_k^{-2} \nabla f_k\|^2},$$

e o resultado está provado. ■

Definida a direção de Cauchy, algumas novas definições podem ser introduzidas. Essas definições, usuais nas teorias de região de confiança, serão de extrema importância quanto à aceitação ou rejeição de um direção candidata.

Dada uma direção $p \in \mathbb{R}^n$, serão definidas a redução efetiva e a redução prevista

respectivamente por

$$\text{ared}(p) \;\; = \;\; f(x^k) - f(x^k + \alpha(p))$$

e

$$\text{pred}(p) \;\; = \;\; m_k(0) - m_k(\alpha(p)).$$

Para efeito de convergência global, uma direção candidata p é submetida à condição de decréscimo suficiente do modelo m_k com relação ao decréscimo atingido pela direção de Cauchy. Portanto, a condição

$$\rho_C^k(p) = \frac{\text{pred}(p)}{\text{pred}(p_C^k)} = \frac{m_k(0) - m_k(\alpha(p))}{m_k(0) - m_k(\alpha(p_C^k))} \geq \beta_1 \tag{2.15}$$

deve ser testada para uma constante $\beta_1 \in (0, 1)$ devidamente fixada. Note que $\rho_C^k(p) \geq \beta_1$ quando $p = p_C^k$.

A condição do decréscimo suficiente com relação ao ponto de Cauchy (2.15) não necessariamente garante que o modelo quadrático m_k seja uma boa representação da função f ao longo da direção p, portanto a condição

$$\rho_f^k(p) = \frac{\text{ared}(p)}{\text{pred}(p)} = \frac{f(x^k) - f(x^k + \alpha(p))}{m_k(0) - m_k(\alpha(p))} \geq \beta_2, \tag{2.16}$$

também deve ser testada para uma constante $\beta_2 \in (0, 1)$ devidamente escolhida para ser fixa.

Na k-ésima iteração, a escolha da direção p^k deve ser explicada com um pouco mais de detalhes. Para isso, será considerada a direção newtoniana de norma mínima,

$$p_N^k = -J(x^k)^\dagger F(x^k). \tag{2.17}$$

Algoritmos para problemas irrestritos que se baseiam no procedimento iterativo $x^{k+1} = x^k + p_N^k$ apresentam, sob certas hipóteses, taxa de convergência localmente quadrática [62]. Esta é, portanto, a razão pela qual o algoritmo proposto considera a direção p_N^k antes de qualquer outra. Assim, se esta direção satisfaz $\|D_k p_N^k\| \leq \Delta$ e ainda as condições (2.15) e (2.16), ela é aceita, e o novo iterado é definido por $x^{k+1} = x^k + \alpha(p_N^k)$. Caso a direção newtoniana de norma mínima seja rejeitada, o algoritmo procede com a abordagem de região de confiança, encontrando uma solução aproximada para o subproblema (2.5) que satisfaça as condições (2.15) e (2.16). Sob certas hipóteses, será provado mais adiante que, a partir de uma certa iteração, todas as direções p^k serão direções newtonianas de norma

mínima e $\alpha(p_N^k) = p_N^k$, o que garantirá convergência localmente quadrática.

Como é bem sabido [45], o subproblema (2.5) pode ser transformado num subproblema de região de confiança usual:

$$\begin{array}{ll} \min & \widetilde{m}_k(\widetilde{p}) \\ s.a. & \|\widetilde{p}\| \leq \Delta, \end{array} \tag{2.18}$$

com $\widetilde{p} = D_k p$ e $\widetilde{m}_k(\widetilde{p}) = m_k(D_k^{-1}p)$. Aqui,

$$\widetilde{m}_k(w) = \frac{1}{2}\|F(x^k)\|^2 + {F_k}^T J_k D_k^{-1} w + \frac{1}{2} w^T D_k^{-1} {J_k}^T J_k D_k^{-1} w.$$

Como pode ser observado, (2.18) é um subproblema de região de confiança padrão, típico de problemas irrestritos, não tendo a necessidade de lidar explicitamente com as restrições de caixa.

Com as considerações feitas anteriormente, o algoritmo proposto pode ser estabelecido.

Algoritmo 2.1 (Algoritmo para SNLI com restrição de caixa). *Seja* $x^0 \in int(\Omega)$. *Escolha* $\Delta_{min} > 0$, $\Delta > \Delta_{min}$ *e* $\theta,\ \beta_1,\ \beta_2,\ \delta_0,\ \delta_1 \in (0,1)$ *tal que* $\delta_0 < \delta_1$.
Faça $k \leftarrow 0$.

Passo 1. *Calcule* J_k, F_k *e* D_k.

Passo 2. *Se* $\|\nabla f_k\| = 0$, *pare!* x^k *é ponto estacionário de* $\min_{x\in\Omega} f(x)$.

Passo 3. *Calcule a direção newtoniana de norma mínima,*

$$p_N^k = -J\left(x^k\right)^{\dagger} F\left(x^k\right).$$

Passo 4. *Se*

$$\left\|D_k p_N^k\right\| \leq \Delta \text{ e } \rho_C^k\left(p_N^k\right) \geq \beta_1, \tag{2.19}$$

defina $p^k = p_N^k$.
Senão, encontre p^k *tal que* $\|D_k p^k\| \leq \Delta$ *e* $\rho_C^k(p^k) \geq \beta_1$, *com* p_C^k *dado pela Equação (2.14).*

Passo 5. *Se*

$$\rho_f^k(p^k) \geq \beta_2, \tag{2.20}$$

faça $x^{k+1} = x^k + \alpha(p^k)$ *e vá para o Passo* 6.
Senão, escolha $\Delta_{novo} \in [\delta_0\Delta, \delta_1\Delta]$, *faça* $\Delta = \Delta_{novo}$ *e volte para o Passo* 4.

Passo 6. *Faça* $\Delta_k = \Delta$, $k \leftarrow k+1$, *escolha* $\Delta > \Delta_{min}$ *e volte para o Passo 1.*

Na seqüência, são colocados os principais resultados de convergência do algoritmo, onde, sob certas hipóteses, são mostradas sua boa definição, convergência a pontos estacionários do problema (2.8) e convergência localmente quadrática, desde que, para este último caso, o ponto-limite esteja no interior de Ω.

2.3 Análise de convergência

Antes de qualquer resultado, algumas hipóteses sobre o problema são colocadas. Estas hipóteses, usuais para esse tipo de problema, são necessárias para obter os resultados de convergência.

Considere o conjunto de nível

$$L = \{x \in \Omega \mid f(x) \leq f(x^0)\}$$

e assuma que L é limitado. Suponha também que:

H1 - J é Lipschitz-contínuo em Ω com constante $2\gamma_0$. Assim, para todo $x, y \in \Omega$,

$$\|J(x) - J(y)\| \leq 2\gamma_0\|x - y\|;$$

H2 - $J(x)$ tem posto completo m para todo $x \in L$.

Pelo fato de F ser uma aplicação diferenciável em Ω, tem-se que existe $\gamma_1 > 0$ tal que, para todo $x,\ y \in \Omega$,

$$\|F(x) - F(y)\| \leq \gamma_1\|x - y\|. \tag{2.21}$$

Também, pela continuidade de J e compacidade de L, resulta que $\|J(x)\|$ é limitada para todo $x \in L$. Assim, como conseqüência da hipótese H2, tem-se que

$$\|J(x)^\dagger\| = \|J(x)^T(J(x)J(x)^T)^{-1}\| \leq \mu \tag{2.22}$$

para todo $x \in L$ e alguma constante $\mu > 0$. Já, como conseqüência da hipótese H1, segue que, para todo $x,\ y \in L$,

$$\begin{aligned} \|F(x) - F(y) - J(y)(x - y)\| &= \|\int_0^1 [J(y + t(x - y)) - J(y)](x - y)dt\| \\ &\leq 2\gamma_0(\int_0^1 tdt)\|x - y\|^2 \\ &= \gamma_0\|x - y\|^2. \end{aligned} \tag{2.23}$$

Os resultados que se seguem fornecem propriedades teóricas do algoritmo. Estas serão importantes para os resultados subseqüentes, em que resultados de convergência serão provados. A demonstração do lema que vem a seguir segue próxima a de um resultado dado por Bellavia *et al.* [6, Lema 3.1].

Lema 2.3. *Suponha que a hipótese H1 é satisfeita e considere p tal que*

$$\|J(x^k)\alpha(p) + F(x^k)\| \leq \|F(x^k)\|.$$

Então,

$$|ared\,(p) - pred\,(p)\,| \leq \varepsilon^k(p)\,\|\alpha\,(p)\|^2\,,$$

onde

$$\varepsilon^k(p) = \gamma_0\left\|F(x^k)\right\| + \frac{1}{2}\gamma_0^2\,\|\alpha\,(p)\|^2\,.$$

Prova. Pelo teorema de Taylor, segue que

$$F(x^k + \alpha(p)) = F(x^k) + \int_0^1 J(x^k + \xi\alpha(p))\alpha(p)d\xi$$

para algum $\xi \in (0,1)$. Defina

$$s_k = \int_0^1 (J(x^k + \xi\alpha(p)) - J(x^k))\alpha(p)d\xi$$

e note que $\|s_k\| \leq \gamma_0\|\alpha(p)\|^2$. Assim,

$$\begin{aligned}
2|\text{ared}(p) - \text{pred}(p)| &= 2|m_k(\alpha(p)) - f(x^k + \alpha(p))| \\
&= |\|F(x^k) + J(x^k)\alpha(p)\|^2 - \|F(x^k) + J(x^k)\alpha(p) + s_k\|^2| \\
&\leq 2\|F(x^k) + J(x^k)\alpha(p)\|\|s_k\| + \|s_k\|^2 \\
&\leq 2\gamma_0\|F(x^k)\|\|\alpha(p)\|^2 + \gamma_0^2\|\alpha(p)\|^4.
\end{aligned}$$

Portanto,

$$|\text{ared}(p) - \text{pred}(p)| \leq \varepsilon^k(p)\|\alpha(p)\|^2,$$

e a tese segue. ■

A seguir, é colocado um lema técnico cujo procedimento usado para demonstrá-lo é praticamente o mesmo que os colocados em [6, Lema 3.3] e [15, Lema 3.1], sendo necessárias apenas leves modificações.

Lema 2.4. *Se p^k satisfaz $\rho_C^k(p^k) \geq \beta_1$, então*

$$pred(p^k) \geq \frac{1}{2}\beta_1 \|D_k^{-1}\nabla f_k\| \min\left\{\Delta, \frac{\|D_k^{-1}\nabla f_k\|}{\|D_k^{-1}{J_k}^T J_k D_k^{-1}\|}, \frac{\theta\|D_k^{-1}\nabla f_k\|}{\|\nabla f_k\|_\infty}\right\}, \tag{2.24}$$

onde θ é a constante usada na definição de $\xi(p^k)$ em (2.11).

Prova. Primeiro será provado que

$$\lambda(d^k) \geq \frac{1}{\|\nabla f_k\|_\infty}, \tag{2.25}$$

onde d^k é dado por (2.13). Para isso, note que a Equação (2.10) pode ser reescrita como

$$\lambda(p) = \min\left\{\max\left\{\frac{l_i - [x^k]_i}{p_i}, \frac{u_i - [x^k]_i}{p_i}\right\} \mid 1 \leq i \leq n\right\}.$$

Como $\lambda(d^k) > 0$ e as componentes de d^k têm o mesmo sinal de $-\nabla f_k$, resulta que

$$\lambda(d^k) = \frac{|[v_k]_j|}{|[d^k]_j|} = \frac{|[v_k]_j|}{|[v_k]_j||[\nabla f_k]_j|} = \frac{1}{|[\nabla f_k]_j|}$$

para algum j, de onde (2.25) segue.

Para provar (2.24), basta encontrar um limitante inferior para $\text{pred}(p_C^k)$, já que, por hipótese, vale que $\rho_C^k(p^k) \geq \beta_1$.

Pela Proposição 2.2, segue que

$$\tau^k = \min\left\{\frac{\|D_k^{-1}\nabla f_k\|^2}{\|J_k D_k^{-2}\nabla f_k\|^2}, \frac{\Delta}{\|D_k^{-1}\nabla f_k\|}\right\}.$$

Suponha que $\xi(p_C^k) = 1$, onde $\xi(p_C^k)$ é dado por (2.11), então $\alpha(p_C^k) = p_C^k$. Assim,

$$\begin{aligned}\text{pred}(p_C^k) &= \text{pred}(\tau^k d^k) = m_k(0) - m_k(\tau^k d^k) \\ &= \tau^k\|D_k^{-1}\nabla f_k\|^2 - \frac{1}{2}(\tau^k)^2\|J_k D_k^{-2}\nabla f_k\|^2. \end{aligned}\tag{2.26}$$

Então, se $\tau^k = \|D_k^{-1}\nabla f_k\|^2/\|J_k D_k^{-2}\nabla f_k\|^2$, tem-se que

$$\begin{aligned}\text{pred}(p_C^k) &= \frac{\|D_k^{-1}\nabla f_k\|^4}{\|J_k D_k^{-2}\nabla f_k\|^2} - \frac{1}{2}\frac{\|D_k^{-1}\nabla f_k\|^4}{\|J_k D_k^{-2}\nabla f_k\|^2} \\ &= \frac{1}{2}\frac{\|D_k^{-1}\nabla f_k\|^4}{\|J_k D_k^{-2}\nabla f_k\|^2}. \end{aligned}\tag{2.27}$$

Como

$$\begin{aligned}\|J_k D_k^{-2}\nabla f_k\|^2 &= \nabla f_k^T D_k^{-2} J_k^T J_k D_k^{-2}\nabla f_k \\ &\leq \|D_k^{-1}\nabla f_k\|^2 \|D_k^{-1} J_k^T J_k D_k^{-1}\|,\end{aligned}$$

segue, pela Equação (2.27), que

$$\text{pred}(p_C^k) \geq \frac{1}{2}\frac{\|D_k^{-1}\nabla f_k\|^2}{\|D_k^{-1} J_k^T J_k D_k^{-1}\|}. \tag{2.28}$$

Se $\tau^k = \Delta/\|D_k^{-1}\nabla f_k\|$, então, como na demonstração da Proposição 2.2, tem-se que $\tau^k \leq \|D_k^{-1}\nabla f_k\|^2/\|J_k D_k^{-2}\nabla f_k\|^2$, e assim

$$\begin{aligned}\text{pred}(p_C^k) &= \tau^k(\|D_k^{-1}\nabla f_k\|^2 - \frac{1}{2}\tau^k\|J_k D_k^{-2}\nabla f_k\|^2) \\ &\geq \frac{1}{2}\tau^k\|D_k^{-1}\nabla f_k\|^2 = \frac{1}{2}\Delta\|D_k^{-1}\nabla f_k\|.\end{aligned} \tag{2.29}$$

Suponha que $\xi(p_C^k) \neq 1$, então $\alpha(p_C^k) = \alpha(d^k) = \xi(d^k)d^k$. Note que $\xi(d^k) \leq \tau^k \leq \|D_k^{-1}\nabla f_k\|^2/\|J_k D_k^{-2}\nabla f_k\|^2$. Assim, pelas equações (2.11) e (2.25), tem-se que

$$\begin{aligned}\text{pred}(p_C^k) &= m_k(0) - m_k(\alpha(d^k)) \\ &= \xi(d^k)(\|D_k^{-1}\nabla f_k\|^2 - \frac{1}{2}\xi(d^k)\|J_k D_k^{-2}\nabla f_k\|^2) \\ &\geq \frac{1}{2}\xi(d^k)\|D_k^{-1}\nabla f_k\|^2 \geq \frac{1}{2}\theta\lambda(d^k)\|D_k^{-1}\nabla f_k\|^2 \\ &\geq \frac{\theta}{2}\frac{\|D_k^{-1}\nabla f_k\|^2}{\|\nabla f_k\|_\infty}.\end{aligned} \tag{2.30}$$

Portanto, por (2.28), (2.29) e (2.30), o resultado segue. ■

O próximo lema confirma a boa definição do algoritmo proposto. Sua prova é semelhante à do Lema 3.4, de [6].

Lema 2.5. *Suponha que a hipótese H1 é satisfeita. Se na k-ésima iteração* $\|\nabla f_k\| \neq 0$, *então o Algoritmo (2.1) está bem definido.*

Prova. Para provar que está bem definido, basta mostrar que, após um número finito de passos, o algoritmo encontra uma direção p^k de modo que a condição $\rho_f^k(p^k) \geq \beta_2$ é satisfeita para algum Δ suficientemente pequeno. Com efeito, suponha que

$$\Delta \leq \min\left\{\frac{\|D_k^{-1}\nabla f_k\|}{\|D_k^{-1}{J_k}^T{J_k}^T D_k^{-1}\|}, \frac{\theta\|D_k^{-1}\nabla f_k\|}{\|\nabla f_k\|_\infty}\right\}.$$

Portanto, pelo Lema 2.4,

$$\Delta \leq \widetilde{C}_k \text{pred}(p^k),$$

onde $\widetilde{C}_k = 2/(\beta_1 \|D_k^{-1}\nabla f_k\|)$. Usando o fato que $\|D_k p^k\| \leq \Delta$, segue que

$$\|\alpha(p^k)\| \leq \|p^k\| \leq \|D_k^{-1}\|\Delta \leq \|D_k^{-1}\|\widetilde{C}_k \text{pred}(p^k). \tag{2.31}$$

Como $m_k(\alpha(p^k)) \leq m_k(0)$, segue pelo Lema 2.3, por (2.31) e o fato de que $\|\alpha(p^k)\| \leq \|D_k^{-1}\|\Delta$, que

$$\begin{aligned} |\text{ared}(p^k) - \text{pred}(p^k)| &\leq \varepsilon^k(p^k)\|\alpha(p^k)\|^2 \\ &\leq \varepsilon^k(p^k)\|D_k^{-1}\|^2\widetilde{C}_k\Delta\text{pred}(p^k). \end{aligned}$$

Então,

$$|\rho_f^k(p^k) - 1| \leq \varepsilon^k(p^k)\|D_k^{-1}\|^2\widetilde{C}_k\Delta.$$

Mas como

$$\varepsilon^k(p^k) \leq \gamma_0\|F_k\| + \frac{1}{2}\gamma_0^2\|D_k^{-1}\|\Delta,$$

resulta que

$$\lim_{\Delta\to 0} |\rho_f^k(p^k) - 1| = 0.$$

Portanto, existe Δ^* tal que $\rho_f^k(p^k) \geq \beta_2$ para todo $\Delta \leq \Delta^*$. Tomando

$$\Delta \leq \min\left\{\Delta^*, \frac{\|D_k^{-1}\nabla f_k\|}{\|D_k^{-1}{J_k}^T{J_k}^T D_k^{-1}\|}, \frac{\theta\|D_k^{-1}\nabla f_k\|}{\|\nabla f_k\|_\infty}\right\},$$

a condição $\rho_f^k(p^k) \geq \beta_2$ vale, e a prova está completa. ■

O resultado a seguir mostra que $\|D(x^k)^{-1}\|$ é limitada, onde $\{x^k\}$ é a seqüência gerada pelo Algoritmo 2.1.

Lema 2.6. *Considere a seqüência de iterados $\{x^k\}$ do Algoritmo 2.1. Então, existe $\chi_D > 0$ tal que*

$$\|D(x^k)^{-1}\| \leq \chi_D.$$

Prova. Como a seqüência $\{x^k\} \subset L$, resulta que $\{x^k\}$ é limitada. Então, pela definição (2.7), segue que a seqüência $\{|v_i(x^k)|\}$ é limitada para todo $i = 1, \ldots, n$. Assim, pela definição (2.6), a matriz $D(x^k)^{-1}$ é também limitada, ou seja, existe $\chi_D > 0$ tal que $\|D(x^k)^{-1}\| \leq \chi_D$. ■

O primeiro resultado de convergência é dado a seguir, o qual está relacionado à convergencia global do algoritmo. Note que, pela suposição H1, tem-se que o gradiente da função mérito f é limitado e Lipschitz-contínuo em L.

Teorema 2.7. *Suponha que H1 é satisfeita e assuma que a seqüência $\{x^k\}$ é infinita. Então, todos os pontos de acumulação de $\{x^k\}$ são pontos estacionários de*

$$\min_{x\in\Omega} f(x).$$

Prova. Seja x^* um ponto de acumulação de $\{x^k\}$. Assim, existe $\mathcal{N}_1$, um subconjunto infinito de $\mathbb{N}$, tal que

$$\lim_{k\in\mathcal{N}_1} x^k = x^*.$$

Primeiro, será mostrado que

$$\lim_{k\in\mathcal{N}_1} \left\|D_k^{-1}\nabla f_k\right\| = 0. \tag{2.32}$$

A prova de (2.32) é por contradição. Pela hipótese de contradição, segue que existe $\epsilon > 0$ tal que $\|D_k^{-1}\nabla f_k\| \geq \epsilon$ para um conjunto infinito de índices $\mathcal{N}_2 \subseteq \mathcal{N}_1$.

Pelo Lema 2.4 e a Equação (2.16), segue que, para todo $k \in \mathcal{N}_1$,

$$\begin{array}{l} \text{ared}(p^k) = f(x^k) - f(x^{k+1}) \geq \beta_2 \text{pred}(p^k) \\ \geq \frac{1}{2}\beta_1\beta_2\|D_k^{-1}\nabla f_k\| \min\left\{\Delta_k, \frac{\|D_k^{-1}\nabla f_k\|}{\|D_k^{-1}{J_k}^T J_k D_k^{-1}\|}, \frac{\theta\|D_k^{-1}\nabla f_k\|}{\|\nabla f_k\|_\infty}\right\}. \end{array} \tag{2.33}$$

Como $\|J(x)\|$ é limitada em L, existe χ_g tal que $\|J(x)^T J(x)\| \leq \chi_g$ e $\|\nabla f(x)\|_\infty \leq \chi_g$ para todo $x \in L$. Também $\|D(x)^{-1}\|$ é limitada em L, então existe $\chi_f > 0$ tal que $\|D_k^{-1}{J_k}^T J_k D_k^{-1}\| \leq \chi_f$ para todo $k \in \mathcal{N}_1$. Assim, pela Equação (2.33),

$$f(x^k) - f(x^{k+1}) \geq \frac{1}{2}\beta_1\beta_2\epsilon \min\left\{\Delta_k, \frac{\epsilon}{\chi_f}, \frac{\theta\epsilon}{\chi_g}\right\} \tag{2.34}$$

para todo $k \in \mathcal{N}_2$.

Como $\{f(x^k)\}_{k\in\mathbb{N}}$ é uma seqüência não crescente e limitada inferiormente,

$$\lim_{k\to\infty}(f(x^k) - f(x^{k+1})) = 0.$$

Assim, pela Equação (2.34), resulta que

$$\lim_{k\in\mathcal{N}_2} \Delta_k = 0.$$

Como em cada iteração o primeiro raio de região de confiança testado Δ é maior que Δ_{min}, segue que, para todo $k \in \mathcal{N}_2$ suficientemente grande, existem $\bar{\Delta}_k$ e $\bar{p}^k \equiv \bar{p}(\bar{\Delta}_k)$ tal

que $\lim_{k\in\mathcal{N}_2}\bar{\Delta}_k = 0$, $\rho_C^k(\bar{p}^k) \geq \beta_1$, $\|D_k\bar{p}^k\| \leq \bar{\Delta}_k$ e $\rho_f^k(\bar{p}^k) < \beta_2$.

Pelo Lema 2.6, tem-se que $\|D_k^{-1}\| \leq \chi_D$ para todo $k \in \mathbb{N}$. Assim, para todo $k \in \mathcal{N}_2$,

$$\|\alpha(\bar{p}^k)\| \leq \|\bar{p}^k\| \leq \chi_D\|D_k\bar{p}^k\| \leq \chi_D\bar{\Delta}_k.$$

Pelo Lema 2.3,

$$|\text{ared}(\bar{p}^k) - \text{pred}(\bar{p}^k)| \quad \leq \quad \varepsilon^k(\bar{p}^k)\chi_D^2\bar{\Delta}_k^2 = \eta^k\bar{\Delta}_k$$

para todo $k \in \mathcal{N}_2$, onde $\eta^k = \varepsilon^k(\bar{p}^k)\chi_D^2\bar{\Delta}_k$. Pelo Lema 2.4 e hipótese de contradição, para $k \in \mathcal{N}_2$ suficientemente grande, segue que

$$\text{pred}(\bar{p}^k) \geq \frac{1}{2}\beta_1\epsilon\bar{\Delta}_k$$

e assim

$$|\text{ared}(\bar{p}^k) - \text{pred}(\bar{p}^k)| \quad \leq \frac{2}{\beta_1\epsilon}\eta^k\text{pred}(\bar{p}^k),$$

ou seja,

$$|\rho_f^k(\bar{p}^k) - 1| \leq \frac{2}{\beta_1\epsilon}\eta^k$$

para $k \in \mathcal{N}_2$ suficientemente grande.

Como $\{\varepsilon^k(\bar{p}^k)\}_k$ é uma seqüência limitada, tem-se que

$$\lim_{k\in\mathcal{N}_2}\eta^k = 0$$

e assim

$$\lim_{k\in\mathcal{N}_2}|\rho_f^k(\bar{p}^k) - 1| = 0,$$

o que contradiz o fato de $\rho_k^f(\bar{p}^k) < \beta_2$ para $k \in \mathcal{N}_2$ suficientemente grande. Logo,

$$\lim_{k\in\mathcal{N}_1}\|D_k^{-1}\nabla f_k\| = 0.$$

Portanto, pelo Lema 2.1, que diz que um ponto $z \in \Omega$ é estacionário de (2.8) se e somente se $\|D(z)^{-1}\nabla f(z)\| = 0$, o resultado segue. ■

O teorema anterior não resolve completamente o problema central deste capítulo, a saber, o problema (2.2), já que nem todo ponto estacionário do PNL (2.8), $\min_{x\in\Omega} f(x)$, anula a aplicação F. O resultado que se segue mostra condições as quais o ponto-limite anula a aplicação F.

Corolário 2.8. *Assuma que as hipóteses H1 e H2 são satisfeitas. Suponha que a seqüência* $\{x^k\}$ *gerada pelo Algoritmo 2.1 é infinita. Seja* $x^* \in int(\Omega)$ *um ponto-limite de* $\{x^k\}$*, então* $F(x^*) = 0$.

Prova. Pelo Teorema 2.7, x^* é um ponto estacionário do problema (2.8) com $x^* \in \text{int}(\Omega)$. Então,

$$J(x^*)^T F(x^*) = 0.$$

Portanto, pela suposição H2, segue que $F(x^*) = 0$, e o corolário está provado. ■

A seguir, são colocados alguns resultados intermediários que serão importantes na demonstração da convergência localmente quadrática do algoritmo.

Lema 2.9. *Seja* $z \in int(\Omega)$*. Então, existe* $r > 0$ *e* $\mathcal{D} > 0$ *tal que* $\|D(x)\| < \mathcal{D}$ *para todo* $x \in \mathcal{B}(z, r) \subset int(\Omega)$.

Prova. Como $z \in \text{int}(\Omega)$, existe $r \in (0, 1]$ tal que $\mathcal{B}(z, 2r) \subset \text{int}(\Omega)$. Seja $\mathcal{D} = \sqrt{1/r}$, assim, para todo $x \in \mathcal{B}(z, r)$,

$$|l_i - x_i|,\ |u_i - x_i| > r \ \text{ para } \ i = 1, \ldots, n.$$

Portanto, pela definição de $D(x)$ (Equação (2.6)), segue que $\|D(x)\| < \sqrt{1/r} = \mathcal{D}$. ■

Lema 2.10. *Assuma que as hipóteses H1 e H2 são satisfeitas. Seja* $K \subseteq \mathbb{N}$ *um subconjunto infinito de índices tal que*

$$\lim_{k \in K} x^k = x^* \in int(\Omega)$$

e

$$F(x^*) = 0.$$

Então, existe $k_0 \in \mathbb{N}$ *tal que, para* $k \geq k_0$ *e* $k \in K$*, o passo de Newton* p_k^N *dado por (2.17) satisfaz (2.19) e (2.20).*

Prova. Pela continuidade de F,

$$\lim_{k \in K} \|F_k\| = \|F(x^*)\| = 0. \tag{2.35}$$

Como a seqüência $\{x^k\}_{k \in K}$ é limitada e $x^* \in \text{int}(\Omega)$, segue pelo Lema 2.9 que existe $\mathcal{D} > 0$ tal que $\|D_k\| \leq \mathcal{D}$ para todo $k \in K$. Assim,

$$\left\|D_k p_N^k\right\| \leq \|D_k\| \left\|-J\left(x^k\right)^\dagger F_k\right\| \leq \mu \mathcal{D} \|F_k\|$$

e, então,

$$\lim_{k \in K} \|D_k p_N^k\| = 0.$$

Assim, existe $k_1 \in \mathbb{N}$ tal que, para todo $k \geq k_1$ e $k \in K$,

$$\|D_k p_N^k\| \leq \Delta_{min}.$$

Por (2.35) e (2.22), resulta que

$$\lim_{k \in K} \|p_N^k\| \leq \mu \lim_{k \in K} \|F_k\| = 0.$$

Pelas definições (2.10) e (2.11), respectivamente de $\lambda(p)$ e $\xi(p)$, mais o fato de que $x^* \in \text{int}(\Omega)$, segue que existe $k_2 \in \mathbb{N}$ tal que $\lambda(p_N^k) > 1$ para todo $k \geq k_2$ e $k \in K$. Conseqüentemente, $\xi(p_N^k) = 1$ e, portanto, $\alpha(p_N^k) = p_N^k$ para $k \geq k_2$ e $k \in K$. Será provado que p_N^k satisfaz as condições $\rho_C^k(p_N^k) \geq \beta_1$ e $\rho_f^k(p_N^k) \geq \beta_2$, dadas respectivamente por (2.19) e (2.20), para k suficientemente grande. Seja $\widetilde{k} = \max\{k_1, k_2\}$. Note que, para todo $k \geq \widetilde{k}$,

$$\begin{aligned} \text{pred}(p_N^k) &= m_k(0) - m_k(p_N^k) \\ &= \frac{1}{2}\|F_k\|^2 - \frac{1}{2}\|J_k(-J_k^\dagger F_k) + F_k\|^2 \\ &= \frac{1}{2}\|F_k\|^2 \end{aligned} \tag{2.36}$$

e

$$\text{pred}(p_C^k) \leq m_k(0) = \frac{1}{2}\|F_k\|^2.$$

Assim, $\rho_C^k(p_N^k) \geq \beta_1$. Como $\lim_{k \in K} \|F_k\| = \lim_{k \in K} \|p_N^k\| = 0$, segue, pela definição de $\varepsilon^k(p)$ no Lema 2.3, que

$$\lim_{k \in K} \varepsilon^k(p_N^k) = 0.$$

Portanto, como $\|J_k p_N^k + F_k\| = 0 \leq \|F_k\|$, segue, por este mesmo lema, que

$$|\text{ared}(p_N^k) - \text{pred}(p_N^k)| \leq \varepsilon^k(p_N^k)\|p_N^k\|^2.$$

Então, dividindo-se a equação acima por $\text{pred}(p_N^k)$ e usando a Equação (2.36), resulta que

$$\left|\frac{\text{ared}(p_N^k)}{\text{pred}(p_N^k)} - 1\right| \leq 2\varepsilon^k(p_N^k)\frac{\|p_N^k\|^2}{\|F_k\|^2}$$

para todo $k \geq \widetilde{k}$, $k \in K$.

Pela definição de p_N^k, tem-se que $\|p_N^k\| \leq \mu\|F_k\|$. Assim, para todo $k \geq \widetilde{k}$, $k \in K$,

$$|\rho_f^k(p_N^k) - 1| \leq 2\mu^2\varepsilon^k(p_N^k).$$

Como $\lim_{k\in K} \varepsilon^k(p_N^k) = 0$, resulta que

$$\lim_{k\in K} \rho_f^k(p_N^k) = 1.$$

Portanto, existe $k_0 \geq \widetilde{k}$ tal que $\rho_f^k(p_N^k) \geq \beta_2$ para todo $k \geq k_0$, $k \in K$, completando a prova. ■

Lema 2.11. *Suponha que valem H1 e H2, e que*

$$F(x^*) = 0,$$

onde $x^* \in int(\Omega)$. Então, existe $\epsilon > 0$ tal que, sempre que $\|x^k - x^*\| \leq \epsilon$, o passo de Newton p_N^k satisfaz (2.19) e (2.20).

Prova. Assuma que a tese não é verdadeira. Então, para todo $\epsilon > 0$, existe um iterado x^k (k dependendo de ϵ) tal que $\|x^k - x^*\| \leq \epsilon$ e pelo menos uma das condições (2.19) ou (2.20) não é satisfeita por p_N^k. Isso implica que existe uma quantidade infinita de x^k com essa propriedade. Com isso em mente, é construída uma subseqüência de $\{x^k\}$ como se segue.
Tome $x^{j(1)}$ tal que $\|x^{j(1)} - x^*\| \leq 1$ e que pelo menos uma das condições, (2.19) ou (2.20), não seja satisfeita para $p_N^{j(1)}$. Para $k > 1$, $x^{j(k)}$ é tal que $\|x^{j(k)} - x^*\| \leq 1/k$, onde pelo menos uma das condições, (2.19) ou (2.20), não é satisfeita para $p_N^{j(k)}$ e $j(k) > j(k-1)$. O fato de que existem infinitos iterados cuja distância a x^* é menor que $1/k$ e que não satisfazem (2.19) ou (2.20) garante que é possível escolher $j(k) > j(k-1)$.

Por construção, a seqüência $\{x^{j(k)}\}$ é uma subseqüência de $\{x^k\}$, converge para x^* e cada um de seus elementos não satisfaz pelo menos uma das condições (2.19) ou (2.20), contradizendo o Lema 2.10. ■

O lema que se segue afirma que, para o ϵ do Lema anterior e $x^* \in \text{int}(\Omega)$ tal que $F(x^*) = 0$, existe $\delta \equiv \delta_\epsilon > 0$ tal que, se $x^{k_0} \in \mathcal{B}(x^*, \delta)$, a seqüência $\{x^k\}$ gerada pelo Algoritmo 2.1 é do tipo newtoniana e está inteiramente contida na bola $\mathcal{B}(x^*, \epsilon)$ a partir do índice k_0.

Lema 2.12. *Suponha que as hipóteses H1 e H2 valem e considere $x^* \in int(\Omega)$ tal que $F(x^*) = 0$. Tome $\epsilon > 0$ como dado pela tese do Lema 2.11. Então, existe $\delta > 0$ ($\delta < \epsilon$), dependendo de ϵ, tal que, se $x^{k_0} \in \mathcal{B}(x^*, \delta)$ para algum índice k_0 da seqüência $\{x^k\}$ gerada pelo Algoritmo 2.1, esta seqüência satisfaz*

$$x^k \in \mathcal{B}(x^*, \epsilon)$$

e

$$x^{k+1} = x^k + p_N^k$$

para todo $k \geq k_0$.

Prova. Defina

$$c_1 = \gamma_0 \mu,$$

$$\delta = \min\left\{\frac{\epsilon}{2(1+2\mu\gamma_1)}, \frac{1}{2c_1\mu\gamma_1}\right\}, \tag{2.37}$$

onde γ_1 é definido em (2.21).

Tome $x^{k_0} \in \mathcal{B}(x^*, \delta)$. Será provado, por indução, que para todo $k \geq k_0$

$$x^k \in \mathcal{B}(x^*, \epsilon).$$

Note que

$$\delta \leq \frac{\epsilon}{2(1+2\mu\gamma_1)} \leq \frac{\epsilon}{2},$$

então $x^{k_0} \in \mathcal{B}(x^*, \epsilon)$.

Pela hipótese indutiva, tem-se que

$$x^j \in \mathcal{B}(x^*, \epsilon)$$

para todo $j \in \{k_0, k_0+1, \ldots, k\}$. O objetivo é provar que $x^{k+1} \in \mathcal{B}(x^*, \epsilon)$.

Pelo Lema 2.11 e hipótese indutiva,

$$x^{j+1} = x^j + p_N^j \tag{2.38}$$

para todo $j \in \{k_0, k_0+1, \ldots, k\}$, onde $p_N^j = -J(x^j)^\dagger F(x^j)$. Assim, por (2.22),

$$\|p_N^j\| \leq \mu \|F(x^j)\| \tag{2.39}$$

para todo $j \in \{k_0, k_0+1, \ldots, k\}$.

Como $J(x^k)J(x^k)^\dagger = I_{m\times m}$, por (2.23) segue que

$$\begin{aligned} \|F(x^{j+1})\| &= \|F(x^{j+1}) - F(x^j) - J(x^j)p_N^j\| \\ &\leq \gamma_0 \|p_N^j\|^2 \end{aligned}$$

para todo $j \in \{k_0, k_0+1, \ldots, k\}$. Assim, por (2.39),

$$\|p_N^j\| \leq \gamma_0 \mu \|p_N^{j-1}\|^2 = c_1 \|p_N^{j-1}\|^2 \tag{2.40}$$

para todo $j \in \{k_0+1, \ldots, k\}$.

Agora, da hipótese de indução e (2.38),

$$\begin{aligned}
\|x^{k+1} - x^*\| &= \|x^k + p_N^k - x^*\| \leq \|x^k - x^*\| + \|p_N^k\| \\
&= \|x^{k-1} + p_N^{k-1} - x^*\| + \|p_N^k\| \\
&\leq \|x^{k-1} - x^*\| + \|p_N^k\| + \|p_N^{k-1}\| \\
&\vdots \\
&\leq \|x^{k_0} - x^*\| + \sum_{j=k_0}^{k} \|p_N^j\|. \qquad (2.41)
\end{aligned}$$

Por (2.40),

$$\begin{aligned}
\|p_N^j\| &\leq c_1\|p_N^{j-1}\|^2 \leq c_1 c_1^2 \|p_N^{j-2}\|^{2^2} \\
&\vdots \\
&\leq c_1 c_1^2 \ldots c_1^{2^{j-k_0-1}} \|p_N^{k_0}\|^{2^{j-k_0}} \\
&= c_1^{\sum_{i=0}^{j-k_0-1} 2^i} \|p_N^{k_0}\|^{2^{j-k_0}} \qquad (2.42)
\end{aligned}$$

para todo $j \in \{k_0+1, \ldots, k\}$.

Mas, por (2.21) e (2.22),

$$\begin{aligned}
\|p_N^{k_0}\| &= \|J(x^{k_0})^\dagger F(x^{k_0})\| \leq \mu\|F(x^{k_0})\| \\
&= \mu\|F(x^{k_0}) - F(x^*)\| \leq \mu\gamma_1\|x^{k_0} - x^*\|.
\end{aligned}$$

Assim, como $\sum_{i=0}^{j-k_0-1} 2^i = 2^{j-k_0} - 1$, resulta de (2.41) e (2.42) que

$$\begin{aligned}
\|x^{k+1} - x^*\| &\leq \|x^{k_0} - x^*\| + \sum_{j=k_0}^{k} c_1^{2^{j-k_0}-1} \|p_N^{k_0}\|^{2^{j-k_0}} \\
&\leq \|x^{k_0} - x^*\| + \sum_{j=k_0}^{k} c_1^{2^{j-k_0}-1} (\mu\gamma_1)^{2^{j-k_0}} \|x^{k_0} - x^*\|^{2^{j-k_0}} \\
&\leq \delta + \mu\gamma_1\delta \sum_{j=k_0}^{k} (c_1\mu\gamma_1\delta)^{2^{j-k_0}-1}. \qquad (2.43)
\end{aligned}$$

Então, por (2.37) e (2.43),

$$\begin{aligned}\|x^{k+1} - x^*\| &\leq \delta + \mu\gamma_1\delta \sum_{j=k_0}^{k} (\frac{1}{2})^{2^{j-k_0}-1} \\ &\leq \delta(1 + \mu\gamma_1 \sum_{j=k_0}^{k} (\frac{1}{2})^{j-k_0}) \leq \delta(1 + 2\mu\gamma_1) \\ &\leq \frac{\epsilon}{2}.\end{aligned}$$

Portanto, $x^{k+1} \in \mathcal{B}(x^*, \epsilon)$ e $x^{k+1} = x^k + p_N^k$ para todo $k \geq k_0$, completando a prova. ■

O resultado a seguir, introduzido por Walker e Watson [62, Teorema 2.1], aqui deixado sem prova, será fundamental para o resultado que mostra a convergência localmente quadrática, a saber, Teorema 2.14. Para isso, dado $\eta > 0$, será definido

$$\Omega_\eta = \{y \in \Omega \mid \|x - y\| \leq \eta \Rightarrow x \in \Omega\}. \tag{2.44}$$

Teorema 2.13. *Seja $F : \mathbb{R}^n \to \mathbb{R}^m$ $(m \leq n)$ uma aplicação satisfazendo H1 e H2 e suponha que Ω_η é dado por (2.44) para algum $\eta > 0$. Então, existe $\varepsilon > 0$ dependendo de γ_0, μ e η tal que se $x^0 \in \Omega_\eta$ e $\|F(x^0)\| \leq \varepsilon$, os iterandos determinados pelo método de norma mínima de Newton, a saber, $x^{k+1} = x^k + p_N^k$, estão bem definidos e convergem a um ponto $x^* \in \Omega$ tal que $F(x^*) = 0$. Além disso, existe uma constante $M > 0$ tal que*

$$\|x^{k+1} - x^*\| \leq M\|x^k - x^*\|^2.$$

O resultado que mostra a convergência localmente quadrática do Algoritmo 2.1 é apresentado a seguir.

Teorema 2.14. *Suponha que as hipóteses H1 e H2 valem e seja $x^* \in int(\Omega)$ um ponto de acumulação da seqüência $\{x^k\}$ gerada pelo Algoritmo 2.1. Então, $F(x^*) = 0$, e a taxa de convergência é localmente quadrática.*

Prova. O fato de $F(x^*) = 0$ segue do Corolário 2.8.

Pelo Lema 2.12, existe $\delta > 0$ tal que, se $\|x^{k_0} - x^*\| \leq \delta$, então $x^{k+1} = x^k + p_N^k$ para todo $k \geq k_0$. Como x^* é um ponto-limite de $\{x^k\}$, segue que $x^{k+1} = x^k + p_N^k$ para k suficientemente grande. Assim, pelo Teorema 2.13, a seqüência converge quadraticamente para alguma solução. Como x^* é um ponto-limite, ela converge para x^*, o que completa a prova. ■

Na seqüência, são descritos alguns resultados numéricos realizados com o Algoritmo 2.1. Esses resultados serão importantes para avaliar o comportamento do método para diversos tipos de problema.

2.4 Resultados numéricos

Nesta seção, são apresentados os experimentos numéricos realizados com o método. O algoritmo proposto é comparado com o método globalizado recentemente introduzido por Kanzow, Yamashita e Fukushima [32], chamado aqui de Algoritmo KYF. Esse método, proposto para resolver sistemas não-lineares quadrados e indeterminados sujeitos a restrições mais gerais do que caixa, apresenta sob certas hipóteses, entre elas a hipótese de "*error bound*", taxa de convergência localmente quadrática. O algoritmo está baseado no método de Levenberg-Marquardt com gradiente projetado. Esse algoritmo, extraído de [32, Algoritmo 3.12], é colocado a seguir.

Algoritmo 2.2 (KYF). *Seja* $x^0 \in \Omega$. *Escolha* $\mu > 0$, $p \geq 1$, $\beta, \sigma, \gamma \in (0,1)$.
Faça $k \leftarrow 0$.

Passo 1. *Se* $F(x^k) = 0$, *pare.*

Passo 2. *Faça* $\mu_k = \mu \|F_k\|^2$ *e calcule* d_U^k *solução de*

$$(J_k^T J_k + \mu_k I) d_U = -J_k^T F_k. \tag{2.45}$$

Passo 3. *Se* $\|F(P_\Omega(x^k + d_U^k))\| \leq \gamma \|F_k\|$ *(*P_Ω *é a projeção em* Ω*), faça* $x^{k+1} = P_\Omega(x^k + d_U^k)$, $k \leftarrow k+1$ *e volte para o Passo 1. Senão, vá para o Passo 4.*

Passo 4. *Defina* $s_{LM}^k = P_\Omega(x^k + d_U^k) - x^k$. *Se* $\nabla f(x^k)^T s_{LM}^k \leq -\rho \|s_{LM}^k\|^p$, *calcule* $t^k = \max\{\beta^i \mid i = 0, 1, \ldots\}$ *tal que*

$$f(x^k + t^k s_{LM}^k) \leq f(x^k) + t^k \sigma \nabla f(x^k)^T s_{LM}^k,$$

defina $x^{k+1} = x^k + t^k s_{LM}^k$, *faça* $k \leftarrow k+1$ *e volte para o Passo 1. Caso contrário, vá para o Passo 5.*

Passo 5. *Calcule* $t^k = \max\{\beta^i \mid i = 0, 1, ...\}$ *tal que*

$$f(x^k(t^k)) \leq f(x^k) + \sigma \nabla f(x^k)^T (x^k(t^k) - x^k),$$

onde $x^k(t) = P_\Omega(x^k - t \nabla f(x^k))$. *Faça* $x^{k+1} = x^k(t^k)$, $k \leftarrow k+1$ *e volte para o Passo 1.*

Para a realização dos testes do Algoritmo KYF, foram tomados os valores de μ iguais a 10^{-1}, 10^{-5} e 10^{-7}. Seguindo a sugestão de Kanzow, Yamashita e Fukushima [32], uma outra fórmula de atualização de μ_k, chamada neste trabalho de $\bar{\mu}$, foi também testada. Iniciando-se com $\mu_0 = \frac{1}{2}10^{-8}\|F(x^0)\|$, a fórmula é dada por

$$\mu_k = \min\{\mu_{k-1}, \|F(x^k)\|^2\},$$

o que não altera as propriedades teóricas do algoritmo. Os demais parâmetros foram declarados como em [32] ($\beta = 0.9,\ \rho = 10^{-8},\ p = 2.1,\ \gamma = 0.99995$ e $\sigma = 10^{-4}$). O objetivo é confrontar o desempenho do método proposto com o Algoritmo KYF, mas antes é preciso esclarecer alguns detalhes da implementação, a qual foi realizada com o software Matlab 5.3.

Para os testes foram considerados alguns problemas sugeridos por Dan, Fukushima e Yamashita [18] e também um conjunto de problemas definidos por conjuntos viáveis de problemas de programação não-linear do livro de Hock e Schittkowski [28], sendo estes problemas da forma

$$F(x) = 0, \qquad x \in \Omega,$$

onde Ω é uma caixa e $F : \mathbb{R}^n \to \mathbb{R}^m$. A Tabela 2.1 mostra os dados dos problemas. Na primeira coluna é colocada a numeração dos problemas que será usada nas tabelas seguintes; na segunda coluna, de onde o problema foi retirado. A terceira e a quarta colunas apresentam a dimensão de cada problema, respectivamente m e n. Os Problemas 1, 3, 6 e 7 são originalmente irrestritos, ou seja, $\Omega = \mathbb{R}^n$. Assim, foram introduzidas as restrições artificiais $0 \leq x_i \leq 2.5$ para todo i. Assim, apenas os Problemas 5, 13 e 14 são mantidos irrestritos. Nos Problemas 10 e 11 as variáveis x_1 e x_2 são ilimitadas superiormente, já no Problema 4 todas as variáveis são ilimitadas superiormente. O Problema 13 corresponde a um sistema linear; e o 14, a um sistema quadrático. Foram também testados ambos os métodos para o problema definido por

$$F(x_1, x_2) = x_2 - \frac{1}{100}x_1, \quad \text{com} \quad -\infty \leq x_1 \leq \infty \ \text{ e } \ x_2 \geq 0,$$

o qual foi chamado de Problema 15.

A seguir, são colocados os detalhes e parâmetros fundamentais da implementação.

Para ambos os métodos, o Jacobiano da aplicação F foi aproximadamente calculado usando-se o método de diferenças finitas, ou seja,

$$[J(x)]_{ij} \approx \frac{[F(x + he_j) - F(x)]_i}{h},$$

Tabela 2.1: Tabela com os dados dos problemas testados

Prob.	Fonte	m	n
1	Problema 46 de [28]	2	5
2	Problema 53 de [28]	3	5
3	Problema 56 de [28]	4	7
4	Problema 63 de [28]	2	3
5	Problema 75 de [28]	3	4
6	Problema 77 de [28]	2	5
7	Problema 79 de [28]	3	5
8	Problema 81 de [28]	3	5
9	Problema 87 de [28]	4	6
10	Problema 107 de [28]	6	9
11	Problema 109 de [28]	6	9
12	Problema 111 de [28]	3	10
13	Problema 2 de [18]	150	300
14	Problema 4 de [18]	150	300
15	–	1	2

para algum h devidamente escolhido para ser o passo da diferença finita. A precisão para aceitar x^k como solução é $\|F(x^k)\| \leq 10^{-6}$. O número máximo de iterações admitidas e avaliações da aplicação F (não levando em consideração as avaliações do cálculo do Jacobiano numérico) é, respectivamente, 5000 iterações e 10000 avaliações. Além disso, a implementação do método proposto (Algoritmo 2.1) pode ser encerrada por algum dos seguintes motivos:

- quando o raio de confiança Δ é menor 10^{-8};
- quando
$$\|D_k^{-1}\nabla f(x^k)\| < 10^{-10},$$
o que acusa que a seqüência de iterados $\{x^k\}$ está se aproximando de um ponto estacionário do problema $\min_{x\in\Omega} f(x)$; ou
- quando $\|D_k^{-1}\|_\infty \leq 7.45 \times 10^{-155}$. Neste caso é dito que D_k^{-1} é numericamente singular, chamado nos experimentos de SINGUL.

Os valores dos parâmetros para inicializar o Algoritmo 2.1 são: $\Delta = \|D(x^0)^{-1}\nabla f(x^0)\|$, $\Delta_{min} = 5 \times 10^{-4}$, $\theta = 0.99995$ (parâmetro para deixar o iterado estritamente viável, definido na Equação (2.11)), $\beta_1 = 0.1$, $\beta_2 = 0.25$ e $\delta_1 = 0.25$.

Sendo assim, a escolha do Δ_{novo} no Passo 5 do Algoritmo 2.1 é feita como se segue:

$$\Delta_{novo} = \min\{\delta_1\Delta, \frac{1}{2}\|D_k\alpha(p_k)\|\}.$$

Já no Passo 6, se $\rho_f^k(p^k) \geq 0.75$, então

$$\Delta = \max\{\Delta_{min}, \Delta, 2\|D_k\alpha(p^k)\|\}.$$

Caso contrário,

$$\Delta = \max\{\Delta_{min}, \Delta\}.$$

No Passo 4 do algoritmo é preciso encontrar uma direção p^k que satisfaça as condições $\rho_C^k(p^k) \geq \beta_1$ e $\|D^k p^k\| \leq \Delta$. Para isso foi utilizado primeiro o método **Dog-Leg**. Uma vez tendo em mãos as direções de Cauchy e Newton, a direção Dog-Leg, chamada aqui de p_d^k, pode ser facilmente calculada [45]:

$$p_d^k = \begin{cases} \dfrac{-\Delta D_k^{-2}\nabla f_k}{\|D_k^{-1}\nabla f_k\|}, & \text{se } \|D_k p_C^k\| \geq \Delta \\ p_C^k + (\mu - 1)(p_N^k - p_C^k), & \text{caso contrário ,} \end{cases}$$

sendo μ a solução positiva da equação

$$\|D_k(p_C^k + (\mu - 1)(p_N^k - p_C^k))\|^2 = \Delta^2.$$

Caso essa direção p_d^k não satisfaça a condição $\rho_C^k(p_d^k) \geq \beta_1$, a direção de Cauchy é aceita, ou seja, $p^k = p_C^k$, e prontamente a condição é satisfeita.

Para encontrar a direção de norma mínima de Newton (2.17) foi usada a fatoração QR.

A aproximação inicial estritamente viável foi escolhida para ser $x^0 = (l + u)/2$, exceto nas seguintes situações:

- nos Problemas 2 e 8 foi usado $x^0 = l + \frac{1}{4}(u - l)$, pelo fato de a aproximação inicial acima ser um ponto estacionário de $\min_{x\in\Omega} f(x)$;
- nos Problemas 13 e 14 foi tomado $x^0 = 150(1, \ldots, 1)^T$, que é uma das aproximações iniciais sugeridas em [32]; e
- no Problema 15 foi considerado $x_0 = (-\frac{1}{2}, \frac{1}{2})$.

Após as considerações colocadas anteriormente, os resultados numéricos podem ser apresentados.

A Tabela 2.2 apresenta os resultados para o Algoritmo 2.1. A dimensão m e n do problema é colocada na segunda coluna, seguida pelo número de iterações (It). Para se ter uma idéia de quanto o algoritmo recuperou a solução, são mostrados na quarta e quinta colunas, respectivamente, o valor da norma euclidiana da aplicação F na aproximação inicial ($\|F(x^0)\|$) e o valor da norma euclidiana na solução aproximada ($\|F(x^*)\|$). A

sexta coluna mostra o número de avaliações da aplicação F (F-aval), desprezando as avaliações feitas no cálculo do Jacobiano numérico. Esses valores são interessantes para que se possa fazer uma análise do custo computacional do método. A última coluna apresenta a quantidade de direções aceitas dos diferentes tipos, a saber, direção de Cauchy (p_C^k), Dog-Leg (p_d^k) e Newton (p_N^k). A notação MAXIT indica que foi atingido o máximo de iterações. MAXFUN indica que foi atingido o máximo de avaliações permitidas, e SINGUL, para declarar que D_k^{-1} é numericamente singular.

Tabela 2.2: Tabela com resultados numéricos do Algoritmo 2.1 para os problemas da Tabela 2.1

Prob	(m,n)	It	$\|F(x^0)\|$	$\|F(x^*)\|$	F-aval	$(p_C^k)/\ (p_d^k)/(p_N^k)$
1	(2,5)	5	3.2	5.3×10^{-10}	6	0 / 0 / 5
2	(3,5)	1	1.0×10	3.2×10^{-15}	2	0 / 0 / 1
3	(4,7)	5	4.4	1.1×10^{-12}	6	0 / 0 / 5
4	(2,3)	5	6.4×10	2.0×10^{-9}	6	0 / 0 / 5
5	(3,4)	7	2.7×10^{3}	9.9×10^{-10}	10	0 / 2 / 5
6	(2,5)	4	4.4	4.3×10^{-7}	5	0 / 0 / 4
7	(3,5)	3	1.6	1.4×10^{-7}	4	0 / 0 / 3
8	(3,5)	308	1.1×10	9.9×10^{-7}	315	290 / 5 / 13
9	(4,6)	SINGUL	4.6×10^{3}	4.4×10^{3}	6	4 /0 /1
10	(6,9)	229	5.8	9.9×10^{-7}	230	225 / 1/ 3
11	(6,9)	MAXIT	5.2×10^{4}	2.9×10^{3}	5001	4997 / 0 /3
12	(3,10)	16	9.5	8.8×10^{-7}	17	0 / 0 / 16
13	(150,300)	2	6.5×10^{3}	0.0	3	0 / 0 / 2
14	(150,300)	11	2.7×10^{5}	8.3×10^{-12}	12	0 / 0 / 11
15	(1,2)	53	5.1×10^{-1}	5.8×10^{-14}	54	51 / 0 / 2

A Tabela 2.3 apresenta o desempenho do Algoritmo KYF. Os dados apresentados são praticamente os mesmos da Tabela 2.2, sendo nesta omitida a dimensão do problema (m e n) e o valor de $\|F(x^0)\|$, já que são os mesmos da Tabela 2.2. Na segunda coluna estão os diferentes valores de μ usados no KYF, a saber, 10^{-1}, 10^{-5}, 10^{-7}, incluindo também a fórmula de atualização de μ_k sugerida por Kanzow, Yamashita e Fukushima [32], chamada aqui de $\bar{\mu}$. A quantidade de direções aceitas no algoritmo para cada problema, que são direções Levenberg-Marquardt (LM), direções de busca linear (LS) e gradiente projetado (GP), é mostrada na sexta coluna. Finalmente, para efeito de comparação, nas três últimas colunas são repetidos alguns valores da Tabela 2.2: número de iterações, valor de $\|F\|$ na solução aproximada e número total de avaliações de F.

A partir da Tabela 2.2 podem ser feitos alguns comentários a respeito do desempenho do Algoritmo 2.1. De modo geral, o método mostrou um bom desempenho, apresentando problemas de convergência somente nos Problemas 9 e 11. Pode ser observado um baixo

número de avaliações da aplicação F, praticamente uma avaliação por iteração. Foi observada em quase todos os problemas convergência localmente quadrática, confirmando a teoria apresentada. Para o Problema 12, apesar de o método ter apresentado um bom desempenho, a convergência foi prejudicada pelo fato de o Jacobiano possuir posto (quase) deficiente na solução aproximada, encontrando-se o menor valor singular do Jacobiano na ordem de 10^{-7}. Já o mal desempenho no Problema 8 se deve ao fato de a solução aproximada se encontrar praticamente na fronteira de Ω.

Comparando os resultados das Tabelas 2.2 e 2.3, pode-se perceber que ambos os métodos apresentaram o mesmo comportamento na maioria dos problemas testados para os valores de $\mu = 10^{-5}$, $\mu = 10^{-7}$ e $\bar{\mu}$. Porém, o Algoritmo KYF de Kanzow, Yamashita e Fukushima não conseguiu resolver os Problemas 5, 9, 11, 13 e 14 para $\mu = 10^{-1}$, nem o Problema 15 para nenhum valor de μ, atingindo o número máximo de iterações. No Problema 11 o Algoritmo KYF excedeu o número máximo de avaliações para $\mu = 10^{-7}$ e $\bar{\mu}$. No Problema 8 podem-se notar muitas avaliações de F quando comparado com o Algoritmo 2.1. Este algoritmo resolveu o problema em número maior de iterações, mas com um número menor de avaliações de F. Para os Problemas 13 e 14, que são respectivamente um sistema linear e um sistema quadrático, o algoritmo também teve um desempenho inferior ao método proposto neste capítulo. Pode-se notar que, quanto menor o parâmetro μ, o desempenho do KYF tende a melhorar, porém, para estes casos, o sistema linear (2.45) pode se tornar mal condicionado. Entretanto, uma abordagem que busca contornar o mal condicionamento, adotada neste trabalho, pode ser encontrada em [45, p. 264]. Neste caso, o sistema linear (2.45) é transformado num problema de quadrados mínimos linear,

$$\min \left\| \begin{bmatrix} J_k \\ \sqrt{\mu_k} I \end{bmatrix} d_U + \begin{bmatrix} F_k \\ 0 \end{bmatrix} \right\|^2,$$

o qual pode ser resolvido de várias maneiras. Neste trabalho foi usado o comando "\" do Matlab.

De maneira geral, o Algoritmo 2.1 se comportou muito bem, sempre apresentando um número de avaliações da aplicação F próximo ao número de iterações, mostrando poucas reduções da região de confiança para se obter uma direção de decréscimo suficiente, o que é bastante satisfatório. Os experimentos mostraram rápida convergência local se a solução se encontra no interior de Ω e se o menor valor singular do Jacobiano, avaliado na solução, não estiver próximo de zero, como mostrado nos resultados teóricos.

Tabela 2.3: Tabela de resultados numéricos realizados com o Algoritmo 2.2 para os problemas da Tabela 2.1 ($\mu = 10^{-1}$, $\mu = 10^{-5}$, $\mu = 10^{-7}$ e $\bar{\mu}$)

	Algoritmo KYF					Algoritmo 2.1		
Prob	μ	It	$\|F(x^*)\|$	F-aval	LM/LS/GP	It	$\|F(x^*)\|$	F-aval
1	10^{-1}	5	6.3×10^{-10}	6	5 / 0 / 0	5	5.3×10^{-10}	6
	10^{-5}	5	5.3×10^{-10}	6	5 / 0 / 0			
	10^{-7}	5	5.3×10^{-10}	6	5 / 0 / 0			
	$\bar{\mu}$	5	5.3×10^{-10}	6	5 / 0 / 0			
2	10^{-1}	7	1.5×10^{-12}	8	7 / 0 / 0	1	3.2×10^{-15}	2
	10^{-5}	2	6.5×10^{-11}	3	2 / 0 / 0			
	10^{-7}	2	1.2×10^{-12}	3	2 / 0 / 0			
	$\bar{\mu}$	2	1.9×10^{-14}	3	2 / 0 / 0			
3	10^{-1}	4	3.1×10^{-11}	5	4 / 0 / 0	5	1.1×10^{-12}	6
	10^{-5}	5	1.1×10^{-12}	6	5 / 0 / 0			
	10^{-7}	5	1.1×10^{-12}	6	5 / 0 / 0			
	$\bar{\mu}$	5	1.1×10^{-12}	6	5 / 0 / 0			
4	10^{-1}	6	1.8×10^{-9}	7	6 / 0 / 0	5	2.0×10^{-9}	6
	10^{-5}	5	2.8×10^{-8}	6	5 / 0 / 0			
	10^{-7}	5	3.0×10^{-8}	6	5 / 0 / 0			
	$\bar{\mu}$	5	3.0×10^{-8}	6	5 / 0 / 0			
5	10^{-1}	MAXIT	2.1×10^{3}	9996	5 / 4995 / 0	7	9.9×10^{-10}	10
	10^{-5}	31	5.8×10^{-10}	33	30 / 1 / 0			
	10^{-7}	13	5.6×10^{-11}	86	7 / 6 / 0			
	$\bar{\mu}$	20	1.0×10^{-11}	152	8 / 12 / 0			
6	10^{-1}	4	3.3×10^{-7}	5	4 / 0 / 0	4	4.3×10^{-7}	5
	10^{-5}	4	4.3×10^{-7}	5	4 / 0 / 0			
	10^{-7}	4	4.3×10^{-7}	5	4 / 0 / 0			
	$\bar{\mu}$	4	4.3×10^{-7}	5	4 / 0 / 0			
7	10^{-1}	3	1.1×10^{-7}	4	3 / 0 / 0	3	1.4×10^{-7}	4
	10^{-5}	3	1.4×10^{-7}	4	3 / 0 / 0			
	10^{-7}	3	1.4×10^{-7}	4	3 / 0 / 0			
	$\bar{\mu}$	3	1.4×10^{-7}	4	3 / 0 / 0			
8	10^{-1}	27	1.6×10^{-10}	482	9 / 10 / 8	308	9.9×10^{-7}	315
	10^{-5}	21	1.6×10^{-7}	472	5 / 9 / 7			
	10^{-7}	25	2.0×10^{-11}	618	5 / 11 / 9			
	$\bar{\mu}$	21	3.1×10^{-8}	489	5 / 9 / 7			
9	10^{-1}	MAXIT	4.4×10^{3}	9999	2 / 4998 / 0	SINGUL	4.4×10^{3}	6
	10^{-5}	237	8.4×10^{-7}	238	237 / 0 / 0			
	10^{-7}	93	8.8×10^{-7}	94	93 / 0 / 0			
	$\bar{\mu}$	92	9.1×10^{-7}	93	92 / 0 / 0			
10	10^{-1}	310	1.0×10^{-6}	313	309 / 1 / 0	229	9.9×10^{-7}	230
	10^{-5}	255	9.9×10^{-7}	258	254 / 1 / 0			
	10^{-7}	245	9.6×10^{-7}	249	244 / 1 / 0			
	$\bar{\mu}$	245	9.7×10^{-7}	249	244 / 1 / 0			
11	10^{-1}	MAXIT	4.7×10^{4}	9999	2 / 4998 / 0	MAXIT	2.9×10^{3}	5001
	10^{-5}	565	9.9×10^{-7}	566	565 / 0 / 0			
	10^{-7}	49	1.3×10^{4}	MAXFUN	1 / 4 / 44			
	$\bar{\mu}$	49	1.3×10^{4}	MAXFUN	1 / 4 / 44			
12	10^{-1}	25	6.1×10^{-7}	26	25 / 0 / 0	16	8.8×10^{-7}	17
	10^{-5}	16	8.8×10^{-7}	17	16 / 0 / 0			
	10^{-7}	16	8.8×10^{-7}	17	16 / 0 / 0			
	$\bar{\mu}$	50	9.8×10^{-7}	51	50 / 0 / 0			
13	10^{-1}	MAXIT	5.9×10^{3}	5001	0 / 5000 / 0	2	0.0	3
	10^{-5}	13	0.0	14	13 / 0 / 0			
	10^{-7}	4	3.0×10^{-15}	5	4 / 0 / 0			
	$\bar{\mu}$	6	1.7×10^{-8}	7	6 / 0 / 0			
14	10^{-1}	MAXIT	2.5×10^{5}	5001	0 / 5000 / 0	11	8.3×10^{-12}	12
	10^{-5}	20	4.9×10^{-8}	21	20 /0/ 0			
	10^{-7}	11	9.6×10^{-11}	12	11 /0/ 0			
	$\bar{\mu}$	29	6.2×10^{-8}	30	29 /0/ 0			
15	10^{-1}	MAXIT	3.0×10^{-3}	5001	5000 / 0 / 0	53	5.8×10^{-14}	54
	10^{-5}	MAXIT	3.0×10^{-3}	5001	5000 / 0 / 0			
	10^{-7}	MAXIT	3.0×10^{-3}	5001	5000 / 0 / 0			
	$\bar{\mu}$	MAXIT	3.0×10^{-3}	5001	5000 / 0 / 0			

Capítulo 3

Algoritmo do Tipo Levenberg-Marquardt para Programação Não-Linear

No capítulo anterior foi introduzido um algoritmo para resolver sistemas não-lineares indeterminados com restrições de caixa, visando a resolver a fase de viabilidade dos algoritmos de restauração para programação não-linear. Neste capítulo é apresentado um novo método para o problema de programação não-linear com função objetivo f diferenciável e conjunto viável arbitrário fechado Γ. Embora seja um método novo, pode ser enquadrado nos métodos do tipo Levenberg-Marquardt, estando as argumentações dos resultados teóricos inspiradas nas do algoritmo introduzido por Martínez e Santos [40]. Assim, a análise de convergência do método proposto é uma modificação da apresentada nesse artigo. O método gera uma seqüência de iterandos viáveis $\{x^k\}$ satisfazendo $f(x^{k+1}) < f(x^k)$. Em um dos principais resultados teóricos deste capítulo é provado que todo ponto de acumulação da seqüência gerada pelo método satisfaz as condições necessárias de primeira ordem, ou seja, é um ponto estacionário do problema de programação não-linear.

Uma aplicação direta do algoritmo proposto é um método globalizado numericamente viável e eficiente para o cálculo de estruturas eletrônicas, o qual será introduzido no Capítulo 6. Neste caso, a função objetivo é o funcional energia, e o conjunto viável é formado por restrições de ortonormalidade, teoria a qual é apresentada no Capítulo 4. Esta aplicação pode ser interpretada como uma abordagem de globalização do bem conhecido algoritmo de ponto fixo SCF, que será visto mais à frente. Sendo assim, será garantida convergência a pontos estacionários do funcional energia para qualquer que seja a aproximação inicial viável considerada.

3.1 Considerações gerais

Como colocado, o objetivo deste capítulo é resolver o problema de programação não-linear

$$\begin{aligned} \min \quad & f(x) \\ s.a. \quad & x \in \Gamma, \end{aligned} \tag{3.1}$$

onde $f : \mathbb{R}^n \to \mathbb{R}$ é uma função continuamente diferenciável sobre algum conjunto aberto contendo $\Gamma \subset \mathbb{R}^n$, sendo este um conjunto fechado arbitrário.

Vários métodos, baseados em abordagens de região de confiança [20, 43, 45, 59], têm sido sugeridos para resolver problemas desse tipo [12, 40, 41, 47, 61]. O método introduzido neste capítulo pode ser visto como uma variação do método introduzido por Martínez e Santos [40]. A idéia-chave do método é considerar um parâmetro de regularização para penalizar a restrição de região de confiança, podendo, portanto, ser enquadrado como um método do tipo Levenberg-Marquardt. Diferentemente dos outros métodos [12, 47, 61], este não faz aproximação linear de Γ.

A principal diferença entre o método aqui sugerido e o método de Martínez e Santos é que, neste último, resolvem-se subproblemas em que o conjunto viável é a intersecção de uma bola euclidiana (região de confiança) com Γ, enquanto no método introduzido neste capítulo resolvem-se subproblemas restritos somente ao conjunto Γ, o que muitas vezes é mais atrativo, como no caso do cálculo de estruturas eletrônicas, em que a resolução desses subproblemas se resume à decomposição em valores singulares de uma matriz [25].

A seguir, são colocadas algumas notações usadas ao longo deste capítulo e, quando necessário, outras serão introduzidas ao longo do texto.

Para representar o gradiente de f avaliado em x, é usado $g(x)$, ou seja, $g(x) \equiv \nabla f(x)$.

Para o k-ésimo termo de uma seqüência será reservada a notação x^k; já o i-ésimo componente de um vetor, digamos x, será representado por $[x]_i$. Quando claro no contexto, os colchetes serão omitidos.

Dada uma aplicação G, G_k representa $G(x^k)$.

Dada uma matriz simétrica definida positiva $A \in \mathbb{R}^{n\times n}$, $\|\cdot\|_A$ denota a norma definida por A. Assim, dado $x \in \mathbb{R}$, $\|x\|_A = \sqrt{x^T A x}$. Para a norma euclidiana será reservada a notação $\|\cdot\|$.

Um cálculo fácil mostra que, dada uma matriz $B \in \mathbb{R}^{n\times n}$ simétrica,

$$-\|B\|\|x\|^2 \leq \lambda_{min}(B)\|x\|^2 \leq x^T B x \leq \lambda_{max}(B)\|x\|^2 \leq \|B\|\|x\|^2 \tag{3.2}$$

para todo $x \in \mathbb{R}^n$, onde $\lambda_{min}(B)$ e $\lambda_{max}(B)$ denotam, respectivamente, o menor e o maior autovalor da matriz B.

A seguir, são colocadas algumas definições usuais em teoria de otimização, as quais serão úteis mais adiante.

Definição 3.1. *Dado $x \in \Gamma$, será chamada de* **curva em Γ partindo de** *x uma função contínua $\alpha : [0, b] \to \Gamma$ tal que $b > 0$ e $\alpha(0) = x$.*

Definição 3.2. *Dado $x \in \Gamma$, será chamada de* **curva em Γ de classe C^k partindo de** *x uma função contínua $\alpha : [0, b] \to \Gamma$ tal que $b > 0$, $\alpha(0) = x$ e $\alpha \in C^k([0, b])$.*

Com essas definições, o bem conhecido resultado de condição necessária de primeira ordem para problemas do tipo (3.1) pode ser colocado.

Teorema 3.3. *Seja x^* um minimizador local de (3.1) e α uma curva qualquer em Γ de classe C^1 partindo de x^*. Então,*

$$g(x^*)^T \alpha'(0) \geq 0.$$

Prova. Defina $\phi : [0, b] \to \mathbb{R}$ por $\phi(t) = f(\alpha(t))$. Como x^* é um minimizador local, existe $b_1 \in (0, b)$ tal que

$$\phi(t) \geq \phi(0)$$

para todo $t \in [0, b_1]$. Assim,

$$\frac{\phi(t) - \phi(0)}{t} \geq 0$$

para todo $t \in [0, b_1]$. Então, tomando o limite, tem-se que $\phi'(0) \geq 0$, mas pela regra da cadeia

$$\phi'(t) = \nabla f(\alpha(t))^T \phi'(t) = g(\alpha(t))^T \phi'(t).$$

Portanto, $g(x^*)^T \alpha'(0) = \phi'(0) \geq 0$. ■

Um ponto $x^* \in \Gamma$ é chamado de **ponto estacionário** de (3.1) se, para toda curva α em Γ de classe C^1 partindo de x^*, tem-se que $g(x^*)^T \alpha'(0) \geq 0$.

3.2 Descrição do método

Nesta seção os principais passos do método são apresentados e o algoritmo é colocado. Sejam M, $\gamma > 0$ constantes fixas, defina

$$\mathcal{B} = \{B \in \mathbb{R}^{n \times n} \mid B = B^T \text{ e } \|B\| \leq M\} \tag{3.3}$$

e

$$\mathcal{B}_+ = \{B \in \mathcal{B} \mid v^T B v \geq \gamma \|v\|^2 \text{ para todo } v \in \mathbb{R}^n\}. \tag{3.4}$$

Note que $\mathcal{B}$ e $\mathcal{B}_+$ são conjuntos fechados.

Lema 3.4. *Seja $A \in \mathcal{B}_+$. Então, A é simétrica definida positiva e*

$$\gamma \|x\|^2 \leq \|x\|_A^2 \leq M \|x\|^2$$

para todo $x \in \mathbb{R}^n$.

Prova. Como $A \in \mathcal{B}_+$, A é simétrica e $v^T A v \geq \gamma \|v\|^2$ para todo $v \in \mathbb{R}^n$. Assim, considerando v_i um autovetor unitário ($\|v_i\| = 1$) qualquer de A associado ao autovalor λ_i, tem-se que

$$\lambda_i = v_i^T A v_i \geq \gamma \|v_i\|^2 = \gamma > 0.$$

Portanto, todos os autovalores de A são positivos e, então, A é simétrica definida positiva.

Para a segunda parte da demonstração, note, pela definição de $\mathcal{B}_+$, que

$$\gamma \|x\|^2 \leq x^T A x = \|x\|_A^2$$

para todo $x \in \mathbb{R}$. Por outro lado, como A é simétrica definida positiva,

$$x^T A x = \|x\|_A^2 \leq \|A\| \|x\|^2 \leq M \|x\|^2,$$

concluindo a demonstração do lema. ■

Dado um iterando $x^k \in \mathbb{R}^n$ e $\rho \geq 0$, considere o modelo quadrático em torno de x^k,

$$Q^k(x, \rho) = g(x^k)^T (x - x^k) + \frac{1}{2}(x - x^k)^T (B_\rho^k + \rho A_\rho^k)(x - x^k), \tag{3.5}$$

onde $B_\rho^k \in \mathcal{B}$ e $A_\rho^k \in \mathcal{B}_+$. Note que B_ρ^k e A_ρ^k dependem de x^k e de ρ, que serve como um parâmetro de regularização.

O modelo (3.5) busca aproximar $f(x) - f(x^k)$ em torno de x^k, já que apresenta o mesmo gradiente que $f(x)$ quando avaliado em $x = x^k$.

Dado $x \in \mathbb{R}^n$, será definido

$$\text{ared}(x^k, x) = f(x^k) - f(x)$$

e

$$\text{pred}(x^k, x) = Q^k(x^k, 0) - Q^k(x, 0) = -Q^k(x, 0).$$

Sendo assim, a partir de um iterando x^k, o próximo iterado é construído como se segue.

Sejam $\beta_1 \in (0, \frac{1}{2}]$ e $1 < \zeta_1 < \zeta_2 < \infty$ constantes fixas. Comece com $\rho = 0$. Sendo assim, o algoritmo proposto consiste, primeiro, em encontrar uma solução global, chamada de x_ρ^k, do subproblema

$$\begin{aligned} \min \quad & Q^k(x, \rho) \\ s.a. \quad & x \in \Gamma, \end{aligned} \tag{3.6}$$

onde $Q^k(x, \rho)$ é o modelo quadrático dado por (3.5). Note que, se x^k não é solução de (3.6), necessariamente tem-se que $\text{pred}(x^k, x_\rho^k) > 0$. Se a condição de decréscimo suficiente é satisfeita, ou seja,

$$\frac{\text{ared}(x^k, x_\rho^k)}{\text{pred}(x^k, x_\rho^k)} \geq \beta_1, \tag{3.7}$$

escolha $x^{k+1} \in \Gamma$ de modo que $f(x^{k+1}) \leq f(x_\rho^k)$ e faça $\rho^k = \rho$. Senão, se $\rho = 0$, escolha $\rho_{novo} > 0$. Caso contrário, escolha $\rho_{novo} \in [\zeta_1\rho, \zeta_2\rho]$. Faça $\rho = \rho_{novo}$, escolha novas matrizes $B_\rho^k \in \mathcal{B}$ e $A_\rho^k \in \mathcal{B}_+$ e resolva novamente (3.6) com esse novo ρ, continuando esse processo até encontrar ρ e x_ρ^k que satisfaçam a condição de decréscimo suficiente (3.7). Por fim, escolha $x^{k+1} \in \Gamma$ de modo que $f(x^{k+1}) \leq f(x_\rho^k)$ e faça $\rho^k = \rho$.

O fato de o modelo quadrático (3.5) ser uma aproximação de primeira ordem de f garante a boa definição do algoritmo, ou seja, necessariamente a iteração acima *termina*, como será mostrado mais adiante. Com esse mesmo argumento, em um outro resultado será mostrado que todo ponto de acumulação da seqüência gerada por esse algoritmo é um ponto estacionário de (3.1).

Após essas considerações o algoritmo pode ser estabelecido.

Algoritmo 3.1 (Algoritmo Modelo). *Tome $x^0 \in \Gamma$ e considere M e γ definidos em (3.3) e (3.4). Escolha $\rho_b > 0 \in \mathbb{R}$, ζ_1, $\zeta_2 \in \mathbb{R}$ com $1 < \zeta_1 < \zeta_2 < +\infty$ e $\beta_1 \in (0, \frac{1}{2}]$.*
Faça $k \leftarrow 0$.

Passo 1. *Calcule f_k, g_k e faça $\rho = 0$.*

Passo 2. *Escolha $B_\rho^k \in \mathcal{B}$ e $A_\rho^k \in \mathcal{B}_+$.*

Passo 3. *Considere $Q^k(x, \rho)$ definida em (3.5) e encontre x_ρ^k solução global de*

$$\begin{aligned} \min \quad & Q^k(x, \rho) \\ s.a. \quad & x \in \Gamma. \end{aligned}$$

Se $Q^k(x_\rho^k, \rho) = 0$, pare! x_ρ^k é um ponto estacionário de (3.1).

Passo 4. *Se*

$$\frac{ared(x^k, x^k_\rho)}{pred(x^k, x^k_\rho))} = \frac{f(x^k) - f(x^k_\rho)}{-Q^k(x^k_\rho, 0)} \geq \beta_1,$$

defina $\rho^k = \rho$, *escolha* $x^{k+1} \in \Gamma$ *tal que* $f(x^{k+1}) \leq f(x^k_\rho)$, *faça* $k \leftarrow k+1$ *e volte para o Passo 1.*

Senão, se $\rho = 0$, *escolha* $\rho_{novo} \in (0, \rho_b]$. *Se* $\rho > 0$, *escolha* $\rho_{novo} \in [\zeta_1\rho, \zeta_2\rho]$. *Faça* $\rho = \rho_{novo}$ *e volte para o Passo 2.*

3.3 Análise de convergência

Nesta seção são estabelecidos os resultados teóricos do Algoritmo (3.1). Entre eles encontra-se o que confirma a boa definição do algoritmo, ou seja, após um número finito de passos o algoritmo encontra, no Passo 3, ρ e x^k_ρ de modo que a condição de decréscimo suficiente seja satisfeita. Já um outro resultado, o mais importante desta seção, diz respeito à convergência do algoritmo, garantindo que, sob certas hipóteses, todo ponto de acumulação é um ponto estacionário de (3.1).

Daqui em diante, neste capítulo, considere o conjunto de nível

$$L = \{x \in \Gamma \mid f(x) \leq f(x^0)\}$$

e suponha que L é limitado.

O primeiro resultado mostra que, se $Q^k(x^k_\rho, \rho) = 0$, então x^k é um ponto estacionário do problema (3.1), justificando o Passo 3 do Algoritmo 3.1.

Lema 3.5. *Se o Algoritmo 3.1 pára no Passo 3, ou seja,* $Q^k(x^k_\rho, \rho) = 0$, *então* x^k *é um ponto estacionário do problema (3.1),*

$$\begin{array}{ll} \min & f(x) \\ s.a. & x \in \Gamma. \end{array}$$

Prova. Se $Q^k(x^k_\rho, \rho) = 0$, então x^k é solução de

$$\begin{array}{ll} \min & Q^k(x, \rho) \\ s.a. & x \in \Gamma. \end{array}$$

Assim, para toda curva α em Γ de classe C^1 partindo de x^k, tem-se, pelo Teorema 3.3, que

$$\nabla_x Q^k(x^k, \rho)^T \alpha'(0) = g(x^k)^T \alpha'(0) \geq 0.$$

Portanto, x^k é um ponto estacionário de (3.1). ■

A definição que se segue será indispensável nos principais resultados teóricos desta seção, principalmente nos que dizem respeito à boa definição e convergência a pontos estacionários.

Definição 3.6. *Dados $A \in \mathcal{B}_+$ e $\alpha : [0,b] \to \mathbb{R}^n$, com $\alpha \in C^1([0,b])$ e $\alpha'(0) \neq 0$, será definido, para $\delta \geq 0$,*

$$\begin{aligned} \tau(\alpha, A, \delta) &= \min\{t \in [0,b] \mid \sqrt{(\alpha(t)-\alpha(0))^T A(\alpha(t)-\alpha(0))} = \delta\} \\ &= \min\{t \in [0,b] \mid \|\alpha(t)-\alpha(0)\|_A = \delta\}. \end{aligned}$$

O seguinte lema técnico é uma variação do Lema 2.1, apresentado em [40]. Este resultado coloca algumas propriedades de $\tau(\alpha, A, \delta)$, as quais serão usadas mais adiante.

Lema 3.7. *Suponha $b > 0$ e, para cada $k \in \mathbb{N}$, considere $\alpha^k : [0,b] \to \mathbb{R}^n$, $\alpha : [0,b] \to \mathbb{R}^n$, $\alpha^k, \alpha \in C^1([0,b])$ para todo $k \in \mathbb{N}$ com $\alpha'(0) \neq 0$ e*

$$\lim_{k\to\infty} \|(\alpha^k)' - \alpha'\|_\infty = 0, \tag{3.8}$$

onde $\|\beta\|_\infty = max\{\|\beta(t)\| \mid t \in [0,b]\}$. Então, existem $c_1, c_2, \bar{\delta} > 0$ e $k_0 \in \mathbb{N}$ tais que $\tau(\alpha^k, A, \delta)$ e $\tau(\alpha, A, \delta)$ estão bem definidos e

$$\begin{cases} c_1\delta \leq \tau(\alpha^k, A, \delta) \leq c_2\delta \\ c_1\delta \leq \tau(\alpha, A, \delta) \leq c_2\delta \end{cases} \tag{3.9}$$

para toda matriz $A \in \mathcal{B}_+$, $\delta \in [0, \bar{\delta}]$ e todo $k \geq k_0$.

Prova. Como $\alpha'(0) \neq 0$, existe $i \in \{1, \ldots, n\}$ tal que $[\alpha']_i \neq 0$. Assuma, sem perda de generalidade, que $[\alpha']_i > 0$. Tome $b_1 \in [0,b]$ tal que

$$[\alpha']_i \geq 2\epsilon > 0$$

para todo $t \in [0, b_1]$. Pela hipótese (3.8), $(\alpha^k)'$ converge uniformemente para α' em $[0, b_1]$, então existe $k_0 \in \mathbb{N}$ tal que

$$[(\alpha^k)']_i(t) \geq \epsilon > 0$$

para todo $t \in [0, b_1]$ e $k \geq k_0$.

Agora, se $A \in \mathcal{B}_+$ e $t \in [0, b_1]$, tem-se, para algum $c > 0$, que

$$\begin{aligned} \|\alpha^k(t) - \alpha^k(0)\|_A &\geq \sqrt{\gamma}\|\alpha^k(t) - \alpha^k(0)\| \\ &\geq \sqrt{\gamma}c|[\alpha^k]_i(t) - [\alpha^k]_i(0)| = \sqrt{\gamma}c|\int_0^t [(\alpha^k)']_i(w)dw| \\ &= \sqrt{\gamma}c\int_0^t [(\alpha^k)']_i(w)dw \geq \sqrt{\gamma}c\epsilon t, \end{aligned} \tag{3.10}$$

onde γ é a constante da definição (3.4) de $\mathcal{B}_+$.

Da mesma maneira, segue que

$$\begin{aligned} \|\alpha(t) - \alpha(0)\|_A &\geq \sqrt{\gamma}\|\alpha(t) - \alpha(0)\| \\ &= \sqrt{\gamma}c\int_0^t [\alpha']_i(w)dw \geq \sqrt{\gamma}c\epsilon t. \end{aligned} \tag{3.11}$$

Pela convergência uniforme de $(\alpha^k)'$, existe $e > 0$ tal que

$$\|(\alpha^k)'(t)\| \leq e$$

e

$$\|\alpha'(t)\| \leq e$$

para todo $t \in [0, b_1]$ e $k \geq k_0$.

Assim, pelo Lema 3.4, dada uma matriz $A \in \mathcal{B}_+$, $t \in [0, b_1]$ e $k \geq k_0$,

$$\begin{aligned} \|\alpha^k(t) - \alpha^k(0)\|_A &\leq \sqrt{M}\|\alpha^k(t) - \alpha^k(0)\| \\ &= \sqrt{M}\|\int_0^t (\alpha^k)'(w)dw\| \\ &\leq \sqrt{M}\int_0^t \|(\alpha^k)'(w)\|dw \leq \sqrt{M}et, \end{aligned} \tag{3.12}$$

onde M é a constante usada para definir o subconjunto $\mathcal{B}$ em (3.3).

Da mesma maneira,

$$\begin{aligned} \|\alpha(t) - \alpha(0)\|_A &\leq \sqrt{M}\|\alpha(t) - \alpha(0)\| \\ &\leq \sqrt{M}\int_0^t \|\alpha'(w)\|dw \leq \sqrt{M}et. \end{aligned} \tag{3.13}$$

Defina $\bar{\delta} = \sqrt{\gamma}c\epsilon b_1$. Assim, por (3.10) e (3.11), tem-se, para todo $A \in \mathcal{B}_+$, que

$$\|\alpha^k(b_1) - \alpha^k(0)\|_A \geq \bar{\delta}$$

e

$$\|\alpha(b_1) - \alpha(0)\|_A \geq \bar{\delta}.$$

Assim, para todo $\delta \in [0, \bar{\delta}]$, toda matriz $A \in \mathcal{B}_+$ e $k \geq k_0$, existem $\tilde{t}$ e $\tilde{\tilde{t}} \in [0, b_1]$ tais que

$$\|\alpha^k(\tilde{t}) - \alpha^k(0)\|_A = \delta$$

e

$$\|\alpha(\tilde{\tilde{t}}) - \alpha(0)\|_A = \delta.$$

Por continuidade, segue que $\tau(\alpha^k, A, \delta), \tau(\alpha, A, \delta) \in [0, b_1]$ estão bem definidos para todo $\delta \in [0, \bar{\delta}]$ e toda matriz $A \in \mathcal{B}_+$. Por (3.10), (3.11), (3.12) e (3.13), tem-se, para toda $A \in \mathcal{B}_+$, $t \in [0, b_1]$ e $k \geq k_0$, que

$$\sqrt{\gamma} c\epsilon t \leq \|\alpha^k(t) - \alpha^k(0)\|_A \leq \sqrt{M} e t$$

e

$$\sqrt{\gamma} c\epsilon t \leq \|\alpha(t) - \alpha(0)\|_A \leq \sqrt{M} e t.$$

Então,

$$\frac{\delta}{\sqrt{M}e} \leq \tau(\alpha^k, A, \delta) \leq \frac{\delta}{\sqrt{\gamma} c\epsilon}$$

e

$$\frac{\delta}{\sqrt{M}e} \leq \tau(\alpha, A, \delta) \leq \frac{\delta}{\sqrt{\gamma} c\epsilon}$$

para todo $\delta \in [0, \bar{\delta}]$, $A \in \mathcal{B}_+$ e $k \geq k_0$. Portanto, (3.9) segue com $c_1 = \frac{1}{\sqrt{M}e}$ e $c_2 = \frac{1}{\sqrt{\gamma} c\epsilon}$. ■

Lema 3.8. *Tome $x^k \in \Gamma$, $\rho \geq 0$, $B \equiv B_\rho^k \in \mathcal{B}$ e $A \equiv A_\rho^k \in \mathcal{B}_+$. Seja x_ρ^k solução global do subproblema (3.6),*

$$\begin{array}{ll} \min & Q^k(x, \rho) \\ s.a. & x \in \Gamma, \end{array}$$

onde $Q^k(x, \rho) = g_k^T(x - x^k) + \frac{1}{2}(x - x^k)^T(B + \rho A)(x - x^k)$. Considere $y \in \Gamma$ tal que

$$\|y - x^k\|_A = \|x_\rho^k - x^k\|_A. \tag{3.14}$$

Então,

$$Q^k(x_\rho^k, 0) \quad \leq \quad Q^k(y, 0). \tag{3.15}$$

Prova. Por hipótese, x_ρ^k é o minimizador global, então

$$Q^k(x_\rho^k, \rho) \leq Q^k(y, \rho),$$

ou seja,

$$\begin{aligned} g_k^T(x_\rho^k - x^k) + \frac{1}{2}(x_\rho^k - x^k)^T(B + \rho A)(x_\rho^k - x^k) \quad &\leq \\ g_k^T(y - x^k) + \frac{1}{2}(y - x^k)^T(B + \rho A)(y - x^k). \end{aligned} \tag{3.16}$$

Mas, por (3.14),

$$(x_\rho^k - x^k)^T A(x_\rho^k - x^k) \quad = \quad (y - x^k)^T A(y - x^k).$$

Então, subtraindo $\frac{1}{2}\rho\|x_\rho^k - x^k\|_A^2$ de ambos os lados de (3.16), segue que

$$g_k^T(x_\rho^k - x^k) + \frac{1}{2}(x_\rho^k - x^k)^T B(x_\rho^k - x^k) \quad \leq \quad g_k^T(y - x^k) + \frac{1}{2}(y - x^k)^T B(y - x^k),$$

ou seja,

$$Q^k(x_\rho^k, 0) \leq Q^k(y, 0),$$

completando a prova. ■

O próximo lema é fundamental para provar a boa definição do algoritmo.

Lema 3.9. *Seja $x^k \in \Gamma$ um ponto não estacionário de (3.1). Considere, para cada $\rho \geq 0$ fixo, $B_\rho^k \in \mathcal{B}$, $A_\rho^k \in \mathcal{B}_+$ e chame x_ρ^k uma solução global do problema*

$$\begin{aligned} \min \quad & Q^k(x, \rho) \\ s.a. \quad & x \in \Gamma, \end{aligned}$$

sendo $Q^k(x, \rho)$ dada por (3.5). Então,

$$\lim_{\rho \to \infty} \|x_\rho^k - x^k\| = 0. \tag{3.17}$$

Prova. Como x^k não é um ponto estacionário de (3.1), para todo $\rho \geq 0$ fixo, resulta que

$$Q^k(x_\rho^k, \rho) < 0.$$

Suponha que a tese (3.17) não é válida, então existe uma seqüência $\{\rho_i\}_{i \in \mathbb{N}} \subset \mathbb{R}$ com

$\lim_{i\to\infty} \rho_i = \infty$, $\epsilon > 0$ e $i_0 \in \mathbb{N}$ tal que

$$\|x^k_{\rho_i} - x^k\| > \epsilon \tag{3.18}$$

para todo $i \geq i_0$.

Pela desigualdade de Cauchy-Schwarz e compacidade de L, existe $M_1 > 0$ tal que $\|g_k\| \leq M_1$. Então, para todo $x \in \Gamma$,

$$-M_1\|x - x^k\| \leq -\|g_k\|\|x - x^k\| \leq g_k^T(x - x^k) \leq \|g_k\|\|x - x^k\| \leq M_1\|x - x^k\|.$$

Também, por (3.2),

$$-M\|x - x^k\|^2 \leq (x - x^k)^T B^k_{\rho_i}(x - x^k)$$

para todo $x \in \Gamma$ e $B^k_{\rho_i} \in \mathcal{B}$, onde a constante $M > 0$ é definida em (3.3).

Então, definindo $M_2 \equiv M_1/\epsilon + M/2 > 0$, por (3.18) e definição de $\mathcal{B}_+$, para todo $i \geq i_0$, segue que

$$\begin{aligned}
Q^k(x^k_{\rho_i}, \rho_i) &= g_k^T(x^k_{\rho_i} - x^k) + \frac{1}{2}(x^k_{\rho_i} - x^k)^T(B^k_{\rho_i} + \rho_i A^k_{\rho_i})(x^k_{\rho_i} - x^k) \\
&\geq -M_1\|x^k_{\rho_i} - x^k\| - \frac{M}{2}\|x^k_{\rho_i} - x^k\|^2 + \frac{1}{2}\rho_i\gamma\|x^k_{\rho_i} - x^k\|^2 \\
&= \left(-\frac{M_1}{\|x^k_{\rho_i} - x^k\|} - \frac{M}{2} + \frac{1}{2}\rho_i\gamma\right)\|x^k_{\rho_i} - x^k\|^2 \\
&> \left(-\left(\frac{M_1}{\epsilon} + \frac{M}{2}\right) + \frac{1}{2}\rho_i\gamma\right)\|x^k_{\rho_i} - x^k\|^2 \\
&= \left(-M_2 + \frac{1}{2}\rho_i\gamma\right)\|x^k_{\rho_i} - x^k\|^2.
\end{aligned} \tag{3.19}$$

Mas como $\lim_{i\to\infty} \rho_i = \infty$, existe $i_1 \in \mathbb{N}$ ($i_1 \geq i_0$) tal que $\rho_i > \frac{4M_2}{\gamma}$ para todo $i \geq i_1$, e então, por (3.19),

$$\begin{aligned}
0 &> Q^k(x^k_{\rho_i}, \rho_i) \\
&> (-M_2 + 2M_2)\|x^k_{\rho_i} - x^k\|^2 \\
&> M_2\epsilon^2 > 0
\end{aligned}$$

para todo $i \geq i_1$, o que é um absurdo. Portanto, a tese (3.17) segue. ■

O próximo teorema mostra que o Algoritmo 3.1 está bem definido, ou seja, após um número finito de iterações, o algoritmo encontra ρ e x^k_ρ de modo que a condição de decréscimo suficiente (3.7) seja satisfeita. Sua demonstração está baseada na demonstração do Teorema 2.3 de [40].

Teorema 3.10. *Se x^k não é um ponto estacionário do problema (3.1),*

$$\begin{aligned} \min \quad & f(x) \\ s.a. \quad & x \in \Gamma, \end{aligned}$$

então x^{k+1} está bem definido no Algoritmo 3.1.

Prova. Como x^k não é ponto estacionário de (3.1), existe uma curva α em Γ de classe C^1 partindo de x^k com $\alpha'(0) \neq 0$ tal que

$$(f \circ \alpha)'(0) = g_k^T \alpha'(0) < 0.$$

Pelo Lema 3.7, existe $\bar{\delta}$ tal que $\tau(\alpha, A, \delta)$ está bem definido e existem c_1 e $c_2 > 0$ tal que

$$c_1 \delta \leq \tau(\alpha, A, \delta) \leq c_2 \delta$$

para todo $\delta \in [0, \bar{\delta}]$ e $A \in \mathcal{B}_+$.

Será denotado $\delta_\rho^k \equiv \sqrt{(x_\rho^k - x^k)^T A_\rho^k (x_\rho^k - x^k)}$, onde A_ρ^k é escolhida no Passo 2 do Algoritmo 3.1 e x_ρ^k definido no Passo 3. Assim, pelo Lema 3.4,

$$\gamma \|x_\rho^k - x^k\|^2 \leq (\delta_\rho^k)^2 \leq M \|x_\rho^k - x^k\|^2, \tag{3.20}$$

já que $A_\rho^k \in \mathcal{B}_+$. Pelo Lema 3.9 tem-se que $\lim_{\rho \to \infty} \|x_\rho^k - x^k\| = 0$, e então, por (3.20),

$$\lim_{\rho \to \infty} \delta_\rho^k = 0. \tag{3.21}$$

Defina $\tau_\rho^k \equiv \tau(\alpha, A_\rho^k, \delta_\rho^k)$. Assim, por (3.21), existe $\bar{\rho} \geq 0$ tal que $\delta_\rho^k \in [0, \bar{\delta}]$ para todo $\rho \geq \bar{\rho}$, e então τ_ρ^k está bem definido e

$$c_1 \delta_\rho^k \leq \tau_\rho^k \leq c_2 \delta_\rho^k \tag{3.22}$$

para todo $\rho \geq \bar{\rho}$.

Pela Definição (3.6),

$$\begin{aligned} \sqrt{(\alpha(\tau_\rho^k) - \alpha(0))^T A_\rho^k (\alpha(\tau_\rho^k) - \alpha(0))} &= \delta_\rho^k \\ &= \sqrt{(x_\rho^k - x^k)^T A_\rho^k (x_\rho^k - x^k)}. \end{aligned}$$

Então, pelos Lemas 3.4 e 3.8, para todo $\rho \geq \bar{\rho}$, tem-se que

$$\begin{aligned} Q^k(x_\rho^k, 0) &\leq Q^k(\alpha(\tau_\rho^k), 0) \\ &= g_k^T(\alpha(\tau_\rho^k) - \alpha(0)) + \frac{1}{2}(\alpha(\tau_\rho^k) - \alpha(0))^T B_\rho^k(\alpha(\tau_\rho^k) - \alpha(0)) \\ &\leq g_k^T(\alpha(\tau_\rho^k) - \alpha(0)) + \frac{1}{2}M\|\alpha(\tau_\rho^k) - \alpha(0)\|^2, \end{aligned} \tag{3.23}$$

onde B_ρ^k é escolhida no Passo 2 do Algoritmo 3.1. Também, pelo Lema 3.4,

$$\|x_\rho^k - x^k\| \leq \frac{1}{\sqrt{\gamma}}\delta_\rho^k.$$

Assim, por (3.22), (3.23) e o fato de que $Q^k(x_\rho^k, 0) < 0$,

$$\begin{aligned} \frac{Q^k(x_\rho^k, 0)}{\|x_\rho^k - x^k\|} &\leq \sqrt{\gamma}\frac{Q^k(x_\rho^k, 0)}{\delta_\rho^k} \\ &\leq \sqrt{\gamma}c_1\frac{Q^k(x_\rho^k, 0)}{\tau_\rho^k} \\ &\leq \sqrt{\gamma}c_1\frac{g_k^T(\alpha(\tau_\rho^k) - \alpha(0)) + \frac{1}{2}M\|\alpha(\tau_\rho^k) - \alpha(0)\|^2}{\tau_\rho^k} \end{aligned} \tag{3.24}$$

para todo $\rho \geq \bar{\rho}$.

Como $\lim_{\rho\to\infty} \delta_\rho^k = 0$, por (3.22) segue que $\lim_{\rho\to\infty} \tau_\rho^k = 0$. Assim,

$$\lim_{\rho\to\infty} \frac{\alpha(\tau_\rho^k) - \alpha(0)}{\tau_\rho^k} = \alpha'(0) \tag{3.25}$$

e

$$\lim_{\rho\to\infty} \|\alpha(\tau_\rho^k) - \alpha(0)\| = 0. \tag{3.26}$$

Então, por (3.24), (3.25) e (3.26),

$$\limsup_{\rho\to\infty} \frac{Q^k(x_\rho^k, 0)}{\|x_\rho^k - x^k\|} \leq c_1\sqrt{\gamma}g_k^T\alpha'(0) < 0.$$

Portanto, existe $\rho_1 \geq \bar{\rho}$ tal que

$$\frac{Q^k(x_\rho^k, 0)}{\|x_\rho^k - x^k\|} \leq \frac{c_1\sqrt{\gamma}}{2}g_k^T\alpha'(0) \equiv c_3 < 0 \tag{3.27}$$

para todo $\rho \geq \rho_1$.

Então, se $\rho \geq \rho_1$, por (3.27) segue que

$$\begin{aligned}
\left|\frac{\text{ared}(x^k, x^k_\rho)}{\text{pred}(x^k, x^k_\rho)} - 1\right| &= \left|\frac{f(x^k_\rho) - f(x^k) - Q^k(x^k_\rho, 0)}{Q^k(x^k_\rho, 0)}\right| \\
&= \frac{|f(x^k_\rho) - f(x^k) - g_k^T(x^k_\rho - x^k) - \frac{1}{2}(x^k_\rho - x^k)^T B^k_\rho (x^k_\rho - x^k)|}{-Q^k(x^k_\rho, 0)} \\
&\leq \frac{|f(x^k_\rho) - f(x^k) - g_k^T(x^k_\rho - x^k)|}{-Q^k(x^k_\rho, 0)} + \frac{M}{2}\frac{\|x^k_\rho - x^k\|^2}{-Q^k(x^k_\rho, 0)} \\
&\leq \frac{|f(x^k_\rho) - f(x^k) - g_k^T(x^k_\rho - x^k)|}{|c_3|\|x^k_\rho - x^k\|} + \frac{M}{2|c_3|}\|x^k_\rho - x^k\|.
\end{aligned}$$

Portanto, como f é diferenciável e $\lim_{\rho\to\infty} \|x^k_\rho - x^k\| = 0$,

$$\lim_{\rho\to\infty}\left|\frac{\text{ared}(x^k, x^k_\rho)}{\text{pred}(x^k, x^k_\rho)} - 1\right| = 0.$$

Assim, depois de um número finito de aumentos de ρ, existe x^k_ρ tal que a condição de decréscimo suficiente (3.7) é verificada. Então, x^{k+1} está bem definido. ■

A definição que se segue foi retirada de [40]. Essa definição coloca um conceito mais fraco de regularidade do que o de independência linear (LICQ) [45], usual da teoria de otimização.

Definição 3.11. *Um ponto $x \in \Gamma$ será chamado de fracamente regular se, para toda curva $\alpha : [0, b] \to \Gamma$ de classe C^1 partindo de x e para toda seqüência $\{x^k\}_{k\in K_1} \subset \Gamma$ convergindo para x, onde $K_1 \subseteq \mathbb{N}$ é um subconjunto infinito de índices, existe $b_1 \in (0, b)$ e $\alpha^k : [0, b_1] \to \Gamma$ $(k \in K_1)$, uma seqüência de curvas de classe C^1 partindo de x^k tal que*

$$\lim_{k\in K_1} \|(\alpha^k)' - \alpha'\|_\infty = 0,$$

onde

$$\|\beta\|_\infty = \max\{\|\beta(t)\| \mid t \in [0, b_1]\}.$$

A hipótese de regularidade fraca definida acima é mais fraca do que a hipótese de regularidade usual de programação não-linear [45], como mostra o resultado a seguir, retirado de [40, p. 36].

Proposição 3.12. *Assuma que $\Gamma = \{x \in \mathbb{R}^n \mid h(x) = 0 \text{ e } c(x) \leq 0\}$, onde $h : \mathbb{R}^n \to \mathbb{R}^m$, $c : \mathbb{R}^n \to \mathbb{R}^p$ e $f, h \in C^1(\mathbb{R}^n)$. Se $\bar{x}$ é um ponto regular de Γ, então $\bar{x}$ é fracamente regular.*

Prova. Ver o apêndice em [40, p. 36]. ■

A recíproca da Proposição 3.12 não é válida. De fato, considerando $\Gamma = \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1^2 - x_2 \geq 0 \text{ e } x_2 \geq 0\}$, o ponto $(0, 0)$ é fracamente regular, mas não é regular.

De agora em diante será assumido que existe c_4 tal que

$$|f(z) - f(x) - g(x)^T(z - x)| \leq c_4\|z - x\|^2 \tag{3.28}$$

para todo $z, x \in \Gamma$.

O resultado que se segue é semelhante ao Lema 3.9. Suas conseqüências serão fundamentais para o Teorema 3.14.

Lema 3.13. *Seja $K_1 \in \mathbb{N}$ um subconjunto infinito de índices e considere $\{x^k\}_{k\in K_1} \subset \Gamma$ uma seqüência de pontos não estacionários do problema (3.1), $\{\tilde{\rho}^k\}_{k\in K_1} \subset \mathbb{R}$ uma seqüência não negativa com $\lim_{k\in K_1} \tilde{\rho}^k = \infty$, $\{B^k_{\tilde{\rho}^k}\}_{k\in K_1} \subset \mathcal{B}$ e $\{A^k_{\tilde{\rho}^k}\}_{k\in K_1} \subset \mathcal{B}_+$ seqüências de matrizes. Para cada $k \in K_1$, considere $\tilde{x}^k$ uma solução global de*

$$\begin{aligned} \min \quad & Q^k(x, \tilde{\rho}^k) \\ s.a. \quad & x \in \Gamma, \end{aligned}$$

onde $Q^k(x, \rho)$ é dada por (3.5). Então,

$$\lim_{k\in K_1} \|\tilde{x}^k - x^k\| = 0. \tag{3.29}$$

Prova. Por hipótese, $\{x^k\}_{k\in K_1}$ não são pontos estacionários de (3.1). Então, para todo $\tilde{\rho}^k$ fixo ($k \in K_1$),

$$Q^k(\tilde{x}^k, \tilde{\rho}^k) < 0.$$

Suponha que a tese (3.29) não é válida. Então, existe $\epsilon > 0$, $K_2 \subseteq K_1$ um subconjunto infinito de índices e $k_0 \in K_2$ tal que

$$\|\tilde{x}^k - x^k\| > \epsilon \tag{3.30}$$

para todo $k \in K_3 \equiv \{k \in K_2 \mid k \geq k_0\}$.

Pela continuidade de g e compacidade de L, existe $M_1 > 0$ tal que

$$\|g(x)\| \leq M_1$$

para todo $x \in L$. Então, pela desigualdade de Cauchy-Schwarz e por (3.2), para todo

$x \in \Gamma$ e $k \in K_3$,

$$-M_1\|x - x^k\| \leq -\|g_k\|\|x - x^k\| \leq g_k^T(x - x^k) \leq \|g_k\|\|x - x^k\| \leq M_1\|x - x^k\|$$

e

$$-M\|x - x^k\|^2 \leq (x - x^k)^T B^k_{\tilde{\rho}^k}(x - x^k),$$

onde $M > 0$ é constante definida em $\mathcal{B}$.

Defina $M_2 = (M_1/\epsilon + M/2) > 0$. Então, por (3.30) e definição de $\mathcal{B}_+$, para todo $k \in K_3$ tem-se

$$\begin{aligned}
Q^k(\tilde{x}^k, \tilde{\rho}^k) &= g_k^T(\tilde{x}^k - x^k) + \frac{1}{2}(\tilde{x}^k - x^k)^T(B^k_{\tilde{\rho}^k} + \tilde{\rho}^k A^k_{\tilde{\rho}^k})(\tilde{x}^k - x^k) \\
&\geq -M_1\|\tilde{x}^k - x^k\| - \frac{M}{2}\|\tilde{x}^k - x^k\|^2 + \frac{1}{2}\tilde{\rho}^k\gamma\|\tilde{x}^k - x^k\|^2 \\
&= \left(-\frac{M_1}{\|\tilde{x}^k - x^k\|} - \frac{M}{2} + \frac{1}{2}\tilde{\rho}^k\gamma\right)\|\tilde{x}^k - x^k\|^2 \\
&> \left(-\left(\frac{M_1}{\epsilon} + \frac{M}{2}\right) + \frac{1}{2}\tilde{\rho}^k\gamma\right)\|\tilde{x}^k - x^k\|^2 \\
&= \left(-M_2 + \frac{1}{2}\tilde{\rho}^k\gamma\right)\|\tilde{x}^k - x^k\|^2. \qquad (3.31)
\end{aligned}$$

Como $\lim_{k\in K_3} \tilde{\rho}^k = \infty$, existe $k_1 \in K_3$ tal que $\tilde{\rho}^k > \frac{4M_2}{\gamma}$ para todo $k \in K_4 \equiv \{k \in K_3 \mid k \geq k_1\}$, e então, por (3.31),

$$\begin{aligned}
0 &> Q^k(\tilde{x}^k, \tilde{\rho}^k) \\
&> M_2\|\tilde{x}^k - x^k\|^2 \\
&> M_2\epsilon^2 > 0
\end{aligned}$$

para todo $k \in K_4$, o que é um absurdo. Portanto,

$$\lim_{k\in K_1} \|\tilde{x}^k - x^k\| = 0$$

e o resultado segue. ■

A seguir, o principal resultado desta seção é colocado. Neste resultado será provado que todo ponto de acumulação fracamente regular do Algoritmo 3.1 é ponto estacionário de (3.1). A demonstração está inspirada na do Teorema 3.2 de [40].

Teorema 3.14. *Seja $\{x^k\}$ uma seqüência infinita gerada pelo Algoritmo 3.1 e considere $x^* \in \Gamma$ fracamente regular um ponto de acumulação de $\{x^k\}$. Então, x^* é um ponto*

estacionário do problema (3.1),

$$\begin{aligned} \min \quad & f(x) \\ s.a. \quad & x \in \Gamma. \end{aligned}$$

Prova. Por hipótese, x^* é um ponto de acumulação de $\{x^k\}$, então existe $K_1 \subset \mathbb{N}$, um subconjunto infinito de índices, tal que

$$\lim_{k \in K_1} x^k = x^*.$$

A prova está dividida em duas partes. Primeiro, será considerado o caso em que a seqüência $\{\rho^k\}$ apresenta uma subseqüência que tende ao infinito e depois o caso em que $\{\rho^k\}$ é limitado superiormente, ou seja,

$$\sup_{k \in K_1} \rho^k \quad = \quad \infty \tag{3.32}$$

e

$$\sup_{k \in K_1} \rho^k \quad < \quad \infty. \tag{3.33}$$

Suponha que vale (3.32). Então, existe $K_2 \subseteq K_1$, um subconjunto infinito de índices, tal que

$$\lim_{k \in K_2} \rho^k = \infty. \tag{3.34}$$

Assim, existe $k_2 \in \mathbb{N}$ tal que $\rho^k > \rho_b$ para todo $k \in K_3 = \{k \in K_2 \mid k \geq k_2\}$. Mas para cada $k \in K_3$ é tentado $\rho \in [0, \rho_b]$, então, para todo $k \in K_3$, existem $\tilde{\rho}^k \in [\frac{\rho^k}{\zeta_2}, \frac{\rho^k}{\zeta_1}]$ com

$$\lim_{k \in K_3} \tilde{\rho}^k = \infty$$

e $\widetilde{x}^k \equiv x^k_{\tilde{\rho}^k}$, solução (global) de

$$\begin{aligned} \min \quad & Q^k(x, \tilde{\rho}^k) \\ s.a. \quad & x \in \Gamma, \end{aligned}$$

tal que

$$f(\widetilde{x}^k) > f(x^k) + \beta_1 Q^k(\widetilde{x}^k, 0).$$

Para $k \in K_3$ e $A^k_{\tilde{\rho}^k} \in \mathcal{B}_+$, será definido $\tilde{\delta}^k \equiv \sqrt{(\widetilde{x}^k - x^k)^T A^k_{\tilde{\rho}^k}(\widetilde{x}^k - x^k)}$, então, pelo Lema 3.4, segue que

$$\gamma \|\widetilde{x}^k - x^k\|^2 \leq (\tilde{\delta}^k)^2 \leq M \|\widetilde{x}^k - x^k\|^2 \tag{3.35}$$

para todo $k \in K_3$.

Também, pelo Lema 3.13,

$$\lim_{k\in K_3} \|\widetilde{x}^k - x^k\| = 0$$

e, então,

$$\lim_{k\in K_3} \tilde{\delta}^k = 0. \tag{3.36}$$

Suponha que x^* não é um ponto estacionário. Então, existe $\alpha : [0, b] \to \Gamma$, uma curva de classe C^1 partindo de x^*, tal que

$$g(x^*)^T \alpha'(0) < 0.$$

Como x^* é fracamente regular e $\lim_{k\in K_1} x^k = x^*$, existe $b_1 \in [0, b]$, $\alpha^k : [0, b_1] \to \Gamma$ $(k \in K_1)$ curvas em Γ de classe C^1 partindo de x^k tal que

$$\lim_{k\in K_1} \|(\alpha^k)' - \alpha'\| = 0. \tag{3.37}$$

Assim, pelo Lema 3.7, existe $\bar{\delta}$ e $k_3 \in K_1$ tal que $\tau(\alpha, A, \delta)$ e $\tau(\alpha^k, A, \delta)$ estão bem definidos para todo $k \in K_4 \equiv \{k \in K_3 \mid k \geq k_3\}$, $\delta \in [0, \bar{\delta}]$ e toda matriz $A \in \mathcal{B}_+$ e, além disso, valem as desigualdades (3.9), ou seja, existem $c_1, c_2 > 0$ tais que

$$c_1\delta \leq \tau(\alpha, A, \delta) \leq c_2\delta$$

e

$$c_1\delta \leq \tau(\alpha^k, A, \delta) \leq c_2\delta$$

para todo $k \in K_4$, $\delta \in [0, \bar{\delta}]$ e $A \in \mathcal{B}_+$.

Por (3.36), existe $k_4 \in K_3$ tal que $\tilde{\delta}^k \in [0, \bar{\delta}]$ para todo $k \in K_5 = \{k \in K_4 \mid k \geq k_4\}$.

Para cada $k \in K_5$, defina $\tilde{\tau}^k = \tau(\alpha^k, A^k_{\tilde{\rho}^k}, \tilde{\delta}^k)$. Assim, $\tilde{\tau}^k$ está bem definido para todo $k \in K_5$ e também

$$c_1\tilde{\delta}^k \leq \tilde{\tau}^k \leq c_2\tilde{\delta}^k. \tag{3.38}$$

Pela Definição (3.6),

$$(\alpha^k(\tilde{\tau}^k) - \alpha^k(0))^T A^k_{\tilde{\rho}^k}(\alpha^k(\tilde{\tau}^k) - \alpha^k(0)) = (\tilde{\delta}^k)^2 = (\tilde{x}^k - x^k)^T A^k_{\tilde{\rho}^k}(\tilde{x}^k - x^k)$$

para todo $k \in K_5$. Então, pelo Lema 3.8,

$$\begin{aligned} Q^k(\tilde{x}^k, 0) &\leq Q^k(\alpha^k(\tilde{\tau}^k), 0) \\ &= g_k^T(\alpha^k(\tilde{\tau}^k) - \alpha^k(0)) + \frac{1}{2}(\alpha^k(\tilde{\tau}^k) - \alpha^k(0))^T B^k_{\tilde{\rho}^k}(\alpha^k(\tilde{\tau}^k) - \alpha^k(0)) \\ &\leq g_k^T(\alpha^k(\tilde{\tau}^k) - \alpha^k(0)) + \frac{1}{2}M\|\alpha^k(\tilde{\tau}^k) - \alpha^k(0)\|^2, \end{aligned} \tag{3.39}$$

onde a matriz $B^k_{\tilde{\rho}^k}$ é definida no Passo 2 do Algoritmo 3.1.

Por (3.35),

$$\|\tilde{x}^k - x^k\| \geq \frac{1}{\sqrt{\gamma}}\tilde{\delta}^k,$$

então, por (3.38), (3.39) e o fato de que $Q^k(\tilde{x}^k, 0) < 0$,

$$\begin{aligned} \frac{Q^k(\tilde{x}^k, 0)}{\|\tilde{x}^k - x^k\|} &\leq \sqrt{\gamma}\frac{Q^k(\tilde{x}^k, 0)}{\tilde{\delta}^k} \\ &\leq \sqrt{\gamma}c_1\frac{Q^k(\tilde{x}^k, 0)}{\tilde{\tau}^k} \\ &\leq \sqrt{\gamma}c_1[\frac{g_k^T(\alpha^k(\tilde{\tau}^k) - \alpha^k(0))}{\tilde{\tau}^k} + \frac{1}{2}M\frac{\|\alpha^k(\tilde{\tau}^k) - \alpha^k(0)\|^2}{\tilde{\tau}^k}] \end{aligned} \tag{3.40}$$

para todo $k \in K_5$.

Por (3.37), tem-se que

$$\begin{aligned} \left\|\frac{\alpha^k(\tilde{\tau}^k) - \alpha^k(0)}{\tilde{\tau}^k} - \alpha'(0)\right\| &= \left\|\frac{1}{\tilde{\tau}^k}\int_0^{\tilde{\tau}^k}(\alpha^k)'(w)dw - \alpha'(0)\right\| \\ &= \left\|\frac{1}{\tilde{\tau}^k}\int_0^{\tilde{\tau}^k}[(\alpha^k)'(w) - \alpha'(0)]dw\right\| \\ &\leq \left\|\frac{1}{\tilde{\tau}^k}\int_0^{\tilde{\tau}^k}[(\alpha^k)'(w) - \alpha'(w)]dw\right\| + \\ &\quad \left\|\frac{1}{\tilde{\tau}^k}\int_0^{\tilde{\tau}^k}[\alpha'(w) - \alpha'(0)]dw\right\| \\ &\leq \frac{1}{\tilde{\tau}^k}\left[\int_0^{\tilde{\tau}^k}\|(\alpha^k)'(w) - \alpha'(w)\|dw + \|\int_0^{\tilde{\tau}^k}\alpha'(w) - \alpha'(0)dw\|\right] \\ &\leq \|(\alpha^k)' - \alpha'\|_\infty + \frac{\|\alpha(\tilde{\tau}^k) - \alpha(0) - \alpha'(0)\tilde{\tau}^k\|}{\tilde{\tau}^k} \end{aligned}$$

para todo $k \in K_5$. Mas por (3.36) e (3.38), tem-se que

$$\lim_{k \in K_5} \tilde{\tau}^k = 0$$

e, então, pela diferenciabilidade de α, segue que

$$\lim_{k \in K_5} \left\| \frac{\alpha^k(\tilde{\tau}^k) - \alpha^k(0)}{\tilde{\tau}^k} - \alpha'(0) \right\| = 0.$$

Como conseqüência imediata,

$$\lim_{k \in K_5} g_k^T \frac{(\alpha^k(\tilde{\tau}^k) - \alpha^k(0))}{\tilde{\tau}^k} = g(x^*)^T \alpha'(0)$$

e

$$\lim_{k \in K_5} \|\alpha^k(\tilde{\tau}^k) - \alpha^k(0)\| = 0.$$

Portanto, por (3.40) e o de fato que $g(x^*)^T \alpha'(0) < 0$,

$$\limsup_{k \in K_5} \frac{Q^k(\tilde{x}^k, 0)}{\|\tilde{x}^k - x^k\|} \leq c_1 \sqrt{\gamma} g(x^*)^T \alpha'(0) < 0.$$

Logo, existe $k_5 \in K_5$ tal que, para todo $k \in K_6 = \{k \in K_5 \mid k \geq k_5\}$,

$$\frac{Q^k(\tilde{x}^k, 0)}{\|\tilde{x}^k - x^k\|} \leq \frac{c_1 \sqrt{\gamma}}{2} g(x^*)^T \alpha'(0) \equiv c_3 < 0$$

e, então,

$$\begin{aligned}
\left| \frac{\text{ared}(x^k, \tilde{x}^k)}{\text{pred}(x^k, \tilde{x}^k)} - 1 \right| &= \left| \frac{f(\tilde{x}^k) - f(x^k) - Q^k(\tilde{x}^k, 0)}{Q^k(\tilde{x}^k, 0)} \right| \\
&= \frac{|f(\tilde{x}^k) - f(x^k) - g_k^T(\tilde{x}^k - x^k) - \frac{1}{2}(\tilde{x}^k - x^k)^T B^k_{\tilde{\rho}^k}(\tilde{x}^k - x^k)|}{-Q^k(\tilde{x}^k, 0)} \\
&\leq \frac{|f(\tilde{x}^k) - f(x^k) - g_k^T(\tilde{x}^k - x^k)|}{-Q^k(\tilde{x}^k, 0)} + \frac{M}{2} \frac{\|\tilde{x}^k - x^k\|^2}{-Q^k(\tilde{x}^k, 0)} \\
&\leq \frac{|f(\tilde{x}^k) - f(x^k) - g_k^T(\tilde{x}^k - x^k)|}{|c_3| \|\tilde{x}^k - x^k\|} + \frac{M}{2|c_3|} \|\tilde{x}^k - x^k\|.
\end{aligned}$$

Portanto, por (3.28) e o fato de que

$$\lim_{k \in K_6} \|\tilde{x}^k - x^k\| = 0,$$

segue que

$$\lim_{k\in K_6}\left|\frac{\text{ared}(x^k,\tilde{x}^k)}{\text{pred}(x^k,\tilde{x}^k)}-1\right|=0,$$

o que contradiz o fato de

$$f(\tilde{x}^k)>f(x^k)+\beta_1 Q^k(\tilde{x}^k,0),$$

e, portanto, x^* é um ponto estacionário neste caso.

Será analisado agora o caso quando vale (3.33), ou seja, $\{\rho^k\}$ é uma seqüência limitada. Como

$$\lim_{k\in K_1} x^k = x^*$$

e $\{f(x^k)\}$ é uma seqüência monótona decrescente e limitada inferiormente, segue que

$$\lim_{k\in K_1} f(x^{k+1})-f(x^k)=0.$$

Pelo Passo 4 do Algoritmo 3.1,

$$f(x^{k+1})\leq f(x^k_{\rho^k})\leq f(x^k)+\beta_1 Q^k(x^k_{\rho^k},0)$$

e, então,

$$\lim_{k\in K_1} Q^k(x^k_{\rho^k},0)=0,$$

e como

$$Q^k(x^k_{\rho^k},0)\leq Q^k(x^k_{\rho^k},\rho^k)<0,$$

tem-se que

$$\lim_{k\in K_1} Q^k(x^k_{\rho^k},\rho^k)=0. \tag{3.41}$$

Como $\|A^k_{\rho^k}\|,\|B^k_{\rho^k}\|\leq M$ para todo $k\in K_1$, existe $B^*\in\mathcal{B}$, $A^*\in\mathcal{B}_+$ e $K_7\subseteq K_1$, um subconjunto infinito de índices, tal que

$$\lim_{k\in K_7} B^k_{\rho^k}=B^*$$

e

$$\lim_{k\in K_7} A^k_{\rho^k}=A^*.$$

Seja $\bar{\rho}=\sup_{k\in K_7}\rho^k$ e tome $\bar{x}$ solução de

$$\begin{array}{ll}\min & g(x^*)^T(x-x^*)+\dfrac{1}{2}(x-x^*)^T(B^*+\bar{\rho}A^*)(x-x^*)\\ s.a. & x\in\Gamma.\end{array} \tag{3.42}$$

Então, por (3.41),

$$
\begin{aligned}
&g(x^*)^T(\bar{x}-x^*)+\frac{1}{2}(\bar{x}-x^*)^T(B^*+\bar{\rho}A^*)(\bar{x}-x^*)= \\
&\lim_{k\in K_7}[g(x^k)^T(\bar{x}-x^k)+\frac{1}{2}(\bar{x}-x^k)^T(B^k_{\rho^k}+\bar{\rho}A^k_{\rho^k})(\bar{x}-x^k)]\geq \\
&\lim_{k\in K_7}[g(x^k)^T(\bar{x}-x^k)+\frac{1}{2}(\bar{x}-x^k)^T(B^k_{\rho^k}+\rho^k A^k_{\rho^k})(\bar{x}-x^k)]= \\
&\lim_{k\in K_7} Q^k(\bar{x},\rho^k)\geq \lim_{k\in K_7} Q^k(x^k_{\rho^k},\rho^k)=0.
\end{aligned}
$$

Portanto, zero é minimizador global de (3.42) e, assim, pelo Lema 3.5, x^* é um ponto estacionário também neste caso, completando a prova. ■

Uma das principais aplicações do algoritmo introduzido neste capítulo aparece em química computacional, que é o problema do cálculo de estruturas eletrônicas, o qual será abordado daqui em diante por este trabalho.

Capítulo 4

Cálculo de Estruturas Eletrônicas como um Problema de Programação Não-Linear

Neste capítulo são apresentados as principais teorias relacionadas com o cálculo de estruturas. O objetivo é mostrar uma surpreendente ligação deste problema com um problema de programação não-linear. Os resultados apresentados neste capítulo podem facilmente serem encontrados na literatura correspondente [26, 58], entretanto, essas abordagens exige do leitor uma intuição química para se convencer de sua veracidade. Com efeito, neste trabalho foi necessário desenvolver, com um rigor matemático, todas as demonstrações envolvidas.

4.1 Considerações gerais

Uma maneira de calcular estruturas eletrônicas de átomos e moléculas para um sistema com N elétrons consiste em encontrar N autofunções associadas a N autovalores de uma equação em dimensão infinita,

$$\mathcal{F}\chi_a = \varepsilon_a \chi_a, \tag{4.1}$$

conhecida como **Equação de Hartree-Fock**. O operador $\mathcal{F}$ é hermitiano e não-linear, chamado de operador de Fock (ver [60] para mais detalhes). As autofunções χ_a são conhecidas como orbitais *spins*, e os autovalores ε_a representam os níveis de energia. O interesse em encontrar o estado fundamental do sistema químico, ou seja, o estado onde o sistema apresenta menor energia, leva à necessidade de encontrar as N autofunções χ_a associadas aos N menores autovalores ε_a. Muitas propriedades teóricas do problema (4.1) têm sido desenvolvidas, algumas destas diretamente relacionadas à existência de soluções,

como feito por P. L. Lions [35].

Daqui em diante será considerado o caso "Closed-Shell" restrito [53, 60], significando que o sistema eletrônico apresenta uma quantidade par de elétrons, podendo cada elétron assumir dois valores de *spins* possíveis, a saber, α e β. Com efeito, pela simetria do sistema com relação aos *spins*, basta considerar somente N elétrons nas equações. Para tanto, será considerado o problema em dimensão finita, também conhecido como equações de **Hartree-Fock-Roothaan** [53, 58, 60]. A partir dessas considerações se chegará a um problema de programação não-linear com propriedades bastante interessantes.

A seguir, são colocadas as definições e considerações usadas ao longo do texto, o que ajudará a entender melhor o problema.

Como foi colocado, este trabalho considera o caso "Closed Shell"restrito. Portanto, daqui em diante, $2N$ representa o número total de elétrons, e M, o de núcleos do sistema eletrônico.

Considere $\{g_i\}_{i=1}^{K}$ $(K \geq 2N)$ um conjunto linearmente independente (LI) de funções reais em $L_2(\mathbb{R}^3) \cap C_0^2(\mathbb{R}^3)$ tais que as integrais

$$\int_{\mathbb{R}^3}\int_{\mathbb{R}^3} g_\mu(r_1)g_\nu(r_1)\frac{1}{r_{12}}g_\sigma(r_2)g_\lambda(r_2)dr_1dr_2$$

e

$$\int_{\mathbb{R}^3} g_\mu(r)[h(g_\nu)(r)]dr$$

existam, sendo h o operador (4.2) definido à frente e r_{12} a distância entre os elétrons. Assim, para cada i, $g_i : \mathbb{R}^3 \to \mathbb{R}$. Aqui $L_2(\mathbb{R}^3) \cap C_0^2(\mathbb{R}^3)$ denota o conjunto de funções duas vezes continuamente diferenciáveis de quadrado integrável em $\mathbb{R}^3$ que se anulam no infinito. Um exemplo de um conjunto de funções $\{g_i\}$ com essa propriedade seriam as funções gaussianas dadas por

$$g_{a,b}(x) = ae^{-b\|x-z\|^2},$$

com a e b constantes e $z \in \mathbb{R}^3$. Essas funções são bastante conhecidas no cálculo de estruturas eletrônicas (para mais detalhes sobre funções de base, ver [22, 26, 29, 30]).

Para cada núcleo j $(j = 1, \ldots, M)$, sua carga será representada por Z_j.

Seja h o operador tal que, para todo $r \in \mathbb{R}^3$ e para toda função ϑ no espaço de funções considerado,

$$h(\vartheta)(r) = -\frac{1}{2}\nabla^2\vartheta(r) - \sum_{j=1}^{M}\frac{Z_j}{\|r-\bar{r}_j\|}\vartheta(r), \tag{4.2}$$

onde $\bar{r}_j$ são as coordenadas do núcleo j e ∇^2 denota o Laplaciano. Um fato que deve ser

observado é que $h(\vartheta)(r)$ não está definida se r assume as coordenadas de algum núcleo. A importância disso está relacionada apenas com a boa definição das integrais (4.5). Do ponto de vista químico, a primeira parcela do operador h corresponde à energia cinética, e a segunda, à energia potencial de atração entre elétrons e núcleos.

Para cada elétron i, sua posição no espaço será chamada de r_i, ou seja, $r_i = (x_i, y_i, z_i)$, sendo (x_i, y_i, z_i) as coordenadas do elétron i em $\mathbb{R}^3$. Para denotar sua distância a um elétron j, será usado r_{ij}. Assim,

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}.$$

Para facilitar a notação, defina

$$(\mu\nu|\sigma\lambda) = \int_{\mathbb{R}^3}\int_{\mathbb{R}^3} g_\mu(r_1)g_\nu(r_1)\frac{1}{r_{12}}g_\sigma(r_2)g_\lambda(r_2)dr_1dr_2, \tag{4.3}$$

onde $dr_i = dx_idy_idz_i$, $i = 1, 2$. Note que as integrais acima nem sempre existem, então, serão assumidas suas existências para $\mu, \nu, \sigma, \lambda = 1, \ldots, K$, como mencionado anteriormente, na definição do conjunto $\{g_i\}_{i=1}^K$.

Como usual, $\langle\cdot,\cdot\rangle$ denota o produto interno. Assim, são colocadas a seguir algumas notações usadas ao longo do texto que simplificarão as equações subseqüentes.

Para cada $X \in \mathbb{R}^{K\times N}$ e $\mu, \nu = 1 \ldots, K$, sejam

$$\begin{aligned}
G(X)_{\mu\nu} &= \sum_{b=1}^{N}\sum_{\sigma,\lambda=1}^{K} X_{\sigma b}X_{\lambda b}[2(\mu\nu|\sigma\lambda) - (\mu\lambda|\sigma\nu)], &(4.4)\\
H_{\mu\nu} &= \langle g_\mu, hg_\nu\rangle = \int_{\mathbb{R}^3} g_\mu(r)[h(g_\nu)(r)]dr, &(4.5)\\
B_{\mu\nu}^{\sigma\lambda} &= [2(\mu\nu|\sigma\lambda) - (\mu\lambda|\sigma\nu)], &(4.6)\\
S_{\mu\nu} &= \langle g_\mu, g_\nu\rangle = \int_{\mathbb{R}^3} g_\mu(r)g_\nu(r)dr, &(4.7)
\end{aligned}$$

então, $G : \mathbb{R}^{K\times N} \to \mathbb{R}^{K\times K}$, $B \in \mathbb{R}^{K\times K\times K\times K}$ e $H, S \in \mathbb{R}^{K\times K}$.

Na seqüência são apresentados alguns resultados bem conhecidos na literatura [26, 58, 60], mostrando a positividade da matriz S e a simetria da matriz G (ver [25] para mais detalhes sobre matrizes definidas positivas).

Lema 4.1. *Considere a definição (4.3). Então,*

$$(\mu\nu|\sigma\lambda) = (\nu\mu|\sigma\lambda) = (\sigma\lambda|\nu\mu).$$

Prova. A demonstração segue diretamente da definição. ■

Proposição 4.2. $B_{\mu\nu}^{\sigma\lambda} = B_{\sigma\lambda}^{\mu\nu}$.

Prova. Pelo lema anterior e por (4.6), segue que

$$B_{\mu\nu}^{\sigma\lambda} = [2(\mu\nu|\sigma\lambda) - (\mu\lambda|\sigma\nu)] = [2(\sigma\lambda|\mu\nu) - (\sigma\nu|\mu\lambda)] = B_{\sigma\lambda}^{\mu\nu}.$$

■

Proposição 4.3. *A matriz S definida em (4.7) é simétrica definida positiva (SDP).*

Prova. A simetria de S segue diretamente da definição (4.7). Basta provar, portanto, a positividade. Seja $v_i \in \mathbb{R}^K$ (com $\|v_i\| = 1$) um autovetor de S com autovalor λ_i, então $Sv_i = \lambda_i v_i$. Assim,

$$\sum_{j=1}^{K} S_{\mu j}[v_i]_j = \lambda_i [v_i]_\mu.$$

Multiplicando ambos os lados dessa equação por $[v_i]_\mu$ e somando com relação a μ, resulta que

$$\sum_{\mu=1}^{K}\sum_{j=1}^{K} S_{\mu j}[v_i]_\mu [v_i]_j = \lambda_i \sum_{\mu=1}^{K} [v_i]_\mu^2 = \lambda_i \|v_i\|^2 = \lambda_i.$$

Mas, pela definição (4.7),

$$\begin{aligned}
\sum_{\mu=1}^{K}\sum_{j=1}^{K} S_{\mu j}[v_i]_\mu [v_i]_j &= \sum_{\mu=1}^{K}\sum_{j=1}^{K} \Big(\int_{\mathbb{R}^3} g_\mu(r) g_j(r) dr\Big)[v_i]_\mu [v_i]_j \\
&= \int_{\mathbb{R}^3} \Big(\sum_{\mu=1}^{K} [v_i]_\mu g_\mu(r)\Big)\Big(\sum_{j=1}^{K} [v_i]_j g_j(r)\Big) dr \\
&= \int_{\mathbb{R}^3} \varphi_i(r)\varphi_i(r) dr > 0,
\end{aligned}$$

onde $\varphi_i = \sum_{j=1}^{K} [v_i]_j g_j$. Então, $\lambda_i > 0$ e, como λ_i é um autovalor qualquer, segue que S é simétrica definida positiva. ■

Proposição 4.4. *Seja $X \in \mathbb{R}^{K\times N}$. Então, a matriz $G(X)$ é simétrica.*

Prova. Pela definição (4.4),

$$\begin{aligned}
G(X)_{\mu\nu} &= 2\sum_{b=1}^{N}\sum_{\sigma,\lambda=1}^{K} X_{\sigma b} X_{\lambda b} \int_{\mathbb{R}^3}\int_{\mathbb{R}^3} g_\mu(r_1) g_\nu(r_1) \frac{1}{r_{12}} g_\sigma(r_2) g_\lambda(r_2) dr_1 dr_2 \\
&- \sum_{b=1}^{N}\sum_{\sigma,\lambda=1}^{K} X_{\sigma b} X_{\lambda b} \int_{\mathbb{R}^3}\int_{\mathbb{R}^3} g_\mu(r_1) g_\lambda(r_1) \frac{1}{r_{12}} g_\sigma(r_2) g_\nu(r_2) dr_1 dr_2.
\end{aligned}$$

Se a primeira parcela dessa equação for chamada de $G_1(X)_{\mu\nu}$ e a segunda de $G_2(X)_{\mu\nu}$, segue que $G(X) = G_1(X) - G_2(X)$. De imediato, segue que $G_1(X)$ é simétrica, portanto, para provar a simetria de $G(X)$, basta provar que $G_2(X)$ é simétrica. Mas

$$\begin{aligned}
G_2(X)_{\mu\nu} &= \sum_{b=1}^{N}\sum_{\sigma,\lambda=1}^{K} X_{\sigma b}X_{\lambda b}\int_{\mathbb{R}^3}\int_{\mathbb{R}^3} g_\mu(r_1)g_\lambda(r_1)\frac{1}{r_{12}}g_\sigma(r_2)g_\nu(r_2)dr_1dr_2 \\
&= \sum_{b=1}^{N}\int\int g_\mu(r_1)(\sum_{\lambda=1}^{K} X_{\lambda b}g_\lambda(r_1))\frac{1}{r_{12}}(\sum_{\sigma=1}^{K} X_{\sigma b}g_\sigma(r_2))g_\nu(r_2)dr_1dr_2 \\
&= \sum_{b=1}^{N}\int\int g_\mu(r_1)\varphi_b(r_1)\frac{1}{r_{12}}\varphi_b(r_2)g_\nu(r_2)dr_1dr_2 \\
&= \sum_{b=1}^{N}\int\int g_\nu(r_1)\varphi_b(r_1)\frac{1}{r_{12}}\varphi_b(r_2)g_\mu(r_2)dr_1dr_2 = G_2(X)_{\nu\mu},
\end{aligned}$$

onde $\varphi_b = \sum_{i=1}^{K} X_{ib}g_i$. Portanto, $G_2(X)$ é simétrica e, conseqüentemente, $G(X)$ também. ■

Dado $X \in \mathbb{R}^{K\times N}$, será definida a aplicação $F : \mathbb{R}^{K\times N} \to \mathbb{R}^{K\times K}$ por

$$F(X) = H + G(X). \tag{4.8}$$

A matriz $F(X)$ é chamada de **Matriz de Fock**.

Algumas propriedades de F devem ser evidenciadas. Essas propriedades serão fundamentais ao longo do texto, principalmente nas demonstrações dos resultados teóricos subseqüentes.

4.2 Propriedades da Matriz de Fock

Nesta seção são colocadas algumas propriedades importantes da matriz F. De imediato, segue um resultado bem conhecido, usado freqüentemente em muitos textos de química quântica [26, 58, 60].

Proposição 4.5. *Seja $X \in \mathbb{R}^{K\times N}$ e considere a Matriz de Fock $F(X)$ como definida em (4.8). Então, $F(X)$ é simétrica.*

Prova. Pela Proposição 4.4 segue que $G(X)$ é simétrica. Portanto, para a demonstração

estar completa, basta provar que a matriz H é simétrica. Note que

$$\begin{aligned} H_{\mu\nu} &= \int_{\mathbb{R}^3} g_\mu(r)[h(g_\nu)(r)]dr \\ &= \int_{\mathbb{R}^3} g_\mu(r)\nabla^2 g_\nu(r)dr - \sum_{j=1}^{M}\int_{\mathbb{R}^3} \frac{Z_j}{\|r-\bar{r}_j\|} g_\mu(r)g_\nu(r)dr. \end{aligned}$$

Como para o operador Laplaciano ∇^2 vale que

$$\int_{\mathbb{R}^3} g_\mu(r)\nabla^2 g_\nu(r)dr = \int_{\mathbb{R}^3} g_\nu(r)\nabla^2 g_\mu(r)dr,$$

(ver [44, exemplo 6, p. 497]), segue que H é simétrica, provando a proposição. ■

No decorrer do trabalho será necessário conhecer as derivadas de cada elemento da aplicação matricial F. O resultado a seguir trata essencialmente disso.

Lema 4.6. *Seja $F(X)$ dada por (4.8), então*

$$\frac{\partial F(X)_{\mu\nu}}{\partial X_{ij}} = \sum_{\sigma=1}^{K} X_{\sigma j}(B_{\mu\nu}^{\sigma i} + B_{\mu\nu}^{i\sigma}).$$

Prova. Note que

$$\begin{aligned} F(X)_{\mu\nu} &= H_{\mu\nu} + \sum_{b=1}^{N}\sum_{\sigma,\lambda=1}^{K} X_{\sigma b}X_{\lambda b}B_{\mu\nu}^{\sigma\lambda} \\ &= H_{\mu\nu} + \sum_{\substack{b=1\\ b\neq j}}^{N}\sum_{\sigma,\lambda=1}^{K} X_{\sigma b}X_{\lambda b}B_{\mu\nu}^{\sigma\lambda} + \sum_{\sigma,\lambda=1}^{K} X_{\sigma j}X_{\lambda j}B_{\mu\nu}^{\sigma\lambda}. \end{aligned}$$

Também, os dois primeiros termos não dependem de X_{ij} e

$$\begin{aligned} \sum_{\sigma,\lambda=1}^{K} X_{\sigma j}X_{\lambda j}B_{\mu\nu}^{\sigma\lambda} &= \sum_{\lambda\neq i}\sum_{\sigma=1}^{K} X_{\sigma j}X_{\lambda j}B_{\mu\nu}^{\sigma\lambda} + X_{ij}\sum_{\sigma=1}^{K} X_{\sigma j}B_{\mu\nu}^{\sigma i} \\ &= \sum_{\lambda\neq i}\sum_{\sigma\neq i} X_{\sigma j}X_{\lambda j}B_{\mu\nu}^{\sigma\lambda} + \\ &\quad X_{ij}\sum_{\sigma\neq i} X_{\sigma j}B_{\mu\nu}^{\sigma i} + X_{ij}\sum_{\lambda\neq i} X_{\lambda j}B_{\mu\nu}^{i\lambda} + X_{ij}^2 B_{\mu\nu}^{ii}. \end{aligned}$$

Então,

$$\begin{aligned}\frac{\partial F(X)_{\mu\nu}}{\partial X_{ij}} &= 2X_{ij}B^{ii}_{\mu\nu} + \sum_{\sigma\neq i} X_{\sigma j}B^{\sigma i}_{\mu\nu} + \sum_{\lambda\neq i} X_{\lambda j}B^{i\lambda}_{\mu\nu} \\ &= \sum_{\sigma=1}^{K} X_{\sigma j}(B^{i\sigma}_{\mu\nu} + B^{\sigma i}_{\mu\nu}).\end{aligned}$$

■

Na seqüência, uma propriedade de invariância por matrizes unitárias é apresentada. Essa propriedade será fundamental ao longo deste capítulo. Antes de colocar o resultado, será necessário fazer algumas considerações e definições.

Seja $U \in \mathbb{R}^{N\times N}$ tal que $U^TU = UU^T = I$, onde $I \in \mathbb{R}^{N\times N}$ representa a matriz identidade. Uma matriz U com essa propriedade é chamada de matriz unitária [25]. Dado $X \in \mathbb{R}^{K\times N}$, considere a transformação unitária pela direita $X' = XU$, ou seja,

$$X'_{\sigma b} = \sum_{i=1}^{N} X_{\sigma i}U_{ib}. \tag{4.9}$$

Defina

$$J(X) = \{Y \in \mathbb{R}^{K\times N} \mid Y = XU \text{ para alguma matriz unitária } U\}.$$

A seguir será provado que F é invariante em $J(X)$ para qualquer $X \in \mathbb{R}^{K\times N}$.

Proposição 4.7. *Seja $X \in \mathbb{R}^{K\times N}$ e considere $F(X)$ dada por (4.8). Tome $X' \in J(X)$, ou seja, $X' = XU$ para alguma matriz unitária U. Então, $F(X') = F(X)$.*

Prova. Pela definição de F,

$$\begin{aligned}F(X')_{\mu\nu} &= \langle g_\mu, hg_\nu\rangle + \sum_{b=1}^{N}\sum_{\sigma,\lambda=1}^{K} X'_{\lambda b}X'_{\sigma b}B^{\sigma\lambda}_{\mu\nu} \\ &= H_{\mu\nu} + \sum_{b=1}^{N}\sum_{\sigma,\lambda}(\sum_{i=1}^{N} X_{\sigma i}U_{ib})(\sum_{j=1}^{N} X_{\lambda j}U_{jb})B^{\sigma\lambda}_{\mu\nu} \\ &= H_{\mu\nu} + \sum_{i=1}^{N}\sum_{\sigma,\lambda}\sum_{j=1}^{N}(\sum_{b=1}^{N} U_{ib}U_{jb})X_{\sigma i}X_{\lambda j}B^{\sigma\lambda}_{\mu\nu}.\end{aligned} \tag{4.10}$$

Como $UU^T = I$, segue que $\sum_{b=1}^{N} U_{ib}U_{jb} = \delta_{ij}$. Então, pela Equação (4.10), segue que

$$\begin{aligned} F(X')_{\mu\nu} &= H_{\mu\nu} + \sum_{i=1}^{N}\sum_{\sigma,\lambda=1}^{K} X_{\lambda i}X_{\sigma i}B_{\mu\nu}^{\sigma\lambda} \\ &= F(X)_{\mu\nu}, \end{aligned}$$

o que prova o resultado. ■

Com essas considerações, o **funcional energia** $E : \mathbb{R}^{K\times N} \to \mathbb{R}$ pode ser definido:

$$E(X) = \sum_{i=1}^{N} X_i^T(F(X) + H)X_i, \tag{4.11}$$

sendo a matriz H definida em (4.5) e

$$X = \begin{bmatrix} | & | & & | \\ X_1 & X_2 & \cdots & X_N \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{K\times N}$$

a matriz formada pelos vetores coluna $X_i \in \mathbb{R}^K$.

A importância desse funcional E e do resultado que vem a seguir ficará evidente mais adiante. Esse resultado irá mostrar a invariância de E em $J(X)$.

Lema 4.8. *Seja $X \in \mathbb{R}^{K\times N}$ e considere $E(X)$ como em (4.11). Seja $X' \in J(X)$, ou seja, $X' = XU$ para alguma matriz unitária U. Então $E(X') = E(X)$.*

Prova. Pela definição (4.11),

$$E(X') = \sum_{i=1}^{N} {X'_i}^T(F(X') + H)X'_i.$$

Como $X' = XU$, segue que $X'_i = \sum_{b=1}^{N} X_b U_{bi}$ e assim, pela Proposição 4.7,

$$\begin{aligned} E(X') &= \sum_{i=1}^{N}(\sum_{b=1}^{N} X_b^T U_{bi})(F(X) + H)(\sum_{a=1}^{N} X_a U_{ai}) \\ &= \sum_{a,b} X_b^T(F(X) + H)X_a(\sum_{i=1}^{N} U_{ai}U_{bi}). \end{aligned}$$

Como $UU^T = I$, segue que

$$E(X') = \sum_{a=1}^{N} X_a^T (F(X) + H) X_a = E(X),$$

e o lema está provado. ■

Até o momento, os principais resultados estão relacionados à aplicação matricial F. Suas implicações, principalmente a da Proposição 4.7, serão indispensáveis para muitos resultados deste capítulo apresentado mais à frente.

A definição que será introduzida na seqüência está intimamente ligada ao objetivo principal deste capítulo.

Uma matriz $C \in \mathbb{R}^{K \times N}$ formada por vetores colunas C_i,

$$C = \begin{bmatrix} | & & | \\ C_1 & \cdots & C_N \\ | & & | \end{bmatrix},$$

é chamada de **ponto fixo Fock** se, para cada $C_i \in \mathbb{R}^K$, $i = 1, \ldots, N$, existe um escalar λ_i tal que

$$\begin{cases} F(C)C_i &= \lambda_i S C_i \\ C_i^T S C_j &= \delta_{ij}, \qquad i, j = 1, \ldots, N, \end{cases} \tag{4.12}$$

onde

$$\delta_{ij} = \begin{cases} 1, & \text{se } i = j \\ 0, & \text{se } i \neq j \end{cases},$$

F é a aplicação matricial (4.8) e S a matriz SDP definida em (4.7).

Os vetores C_i são chamados de autovetores, e os escalares λ_i, de autovalores. Isso porque, uma vez conhecida a matriz C que satisfaça (4.12), o problema pode ser reformulado como um problema de autovalores generalizado, a saber $AC_i = \lambda_i S C_i$, onde $A = F(C)$ (para mais detalhes de problemas de autovalores generalizados, ver [25]). A matriz diagonal $\Lambda \in \mathbb{R}^{N \times N}$ gerada pelos λ_i na diagonal é chamada de matriz de autovalores. É interessante notar que, pela Proposição 4.7, a matriz $F(C)$ é invariante para qualquer permutação das colunas de C. Assim, a disposição dos autovalores na diagonal de Λ pode e será assumida para ser em ordem crescente, ou seja, $\lambda_i \leq \lambda_j$ se $i < j$.

Um ponto fixo Fock $C \in \mathbb{R}^{K \times N}$ é dito para satisfazer o **princípio de "Aufbau"** [26, 58] se C é a matriz formada pelos autovetores associados aos N menores autovalores generalizados de $F(C)$.

Se todos os autovalores forem distintos, isto é, $\lambda_i \neq \lambda_j$ se $i \neq j$, o ponto fixo Fock C será chamado de **não-degenerado**. Se todos pontos que satisfazem (4.12) são não-

degenerados, o problema será chamado também de **não-degenerado**.

Em teorias de química quântica, freqüentemente uma matriz C com a propriedade (4.12) é chamada de **solução autoconsistente** [26, 53, 60].

Note que, pela natureza da matriz F, o problema (4.12) é não-linear, sendo, portanto, necessário um processo iterativo para resolvê-lo.

Existem atualmente vários procedimentos para encontrar pontos fixos Fock [3, 4, 9, 13, 16, 31, 48, 49], sendo o mais conhecido o procedimento de ponto fixo, algumas vezes chamado como método SCF ("Self-Consistent Field") [53]. Em resumo, o método consiste em introduzir uma aproximação inicial para a matriz C, atualizar a matriz F a partir desta matriz inicial, depois diagonalizar $F(C)$, obtendo uma nova matriz de coeficientes, atualizar novamente F e assim sucessivamente, até obter uma solução autoconsistente. Claramente, este método é uma típica iteração de ponto fixo [34, cap. 5]. Alguns dos métodos tradicionais para o cálculo de estruturas eletrônicas, inclusive o de ponto fixo, serão colocados com mais detalhes no Capítulo 5.

Daqui em diante, o principal objetivo deste trabalho é desenvolver uma teoria robusta que possibilite a criação de novos algoritmos eficientes baseados em técnicas de programação não-linear [20, 45] para obter matrizes C que satisfaçam (4.12). Com esse tipo de abordagem, uma grande variedade de métodos de otimização pode ser utilizada, em especial o método introduzido no Capítulo 3, podendo, inclusive, contornar o caso da possível não-convergência do algoritmo de ponto fixo. Uma outra motivação para usar esse tipo de abordagem se deve ao fato do elevado custo computacional na construção da Matriz de Fock F (ordem NK^4), o que permitiria lançar mão de teorias sofisticadas de otimização, de modo que esse custo fosse reduzido.

4.3 O problema de programação não-linear

Dada uma aplicação $\omega : \mathbb{R}^{K\times N} \to \mathbb{R}$, a notação $\frac{\partial \omega(X)}{\partial X_i} : \mathbb{R}^{K\times N} \to \mathbb{R}^K$ denotará a derivada vetorial de ω com relação à variável vetorial X_i, sendo $X_i \in \mathbb{R}^K$ a i-ésima coluna da matriz X. Assim,

$$\frac{\partial \omega(X)}{\partial X_i} = \begin{bmatrix} \frac{\partial \omega(X)}{\partial X_{1i}} \\ \vdots \\ \frac{\partial \omega(X)}{\partial X_{Ki}} \end{bmatrix}.$$

A seguir, é definido o principal problema de programação não-linear (PNL) deste capítulo. Algumas de suas principais propriedades estarão diretamente relacionadas a pontos fixos Fock.

Considere o problema de otimização

$$\begin{aligned} \min \quad & E(X) \\ s.a. \quad & X_i^T S X_j = \delta_{ij}, \quad i,j = 1,\ldots,N, \end{aligned}$$

onde

$$X = \begin{bmatrix} | & | & & | \\ X_1 & X_2 & \cdots & X_N \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{K\times N}$$

e $E : \mathbb{R}^{K\times N} \to \mathbb{R}$ é a função dada por (4.11), a saber,

$$E(X) = \sum_{i=1}^{N} X_i^T (F(X) + H) X_i.$$

O conjunto viável desse problema será denotado por Ω. Assim,

$$\Omega = \{X \in \mathbb{R}^{K\times N} \mid X_i^T S X_j = \delta_{ij},\ i,j = 1,\ldots,N\}, \tag{4.13}$$

também chamado de conjunto S**-ortonormal**. Então, o PNL pode ser reescrito como

$$\begin{aligned} \min \quad & E(X) \\ s.a. \quad & X \in \Omega. \end{aligned} \tag{4.14}$$

De imediato, segue que Ω é fechado e limitado, portanto compacto. Então, com o fato de o funcional E ser contínuo, decorre que o PNL (4.14) admite um minimizador global [55].

Note que o problema (4.14) apresenta N^2 restrições, mas como $X_i^T S X_j = X_j^T S X_i$, segue que algumas dessas restrições são redundantes. A quantidade de restrições pode, portanto, ser reduzida para $(N+1)N/2$. Para fins puramente teóricos, serão consideradas as N^2 restrições em todos os resultados deste capítulo, exceto no Teorema 4.10, em que são desprezadas as restrições redundantes.

Se $C^* \in \mathbb{R}^{K\times N}$ é solução do problema de otimização (4.14), $E(C^*)$ é chamada de **Energia Hartree-Fock**.

A seguir, é colocado um resultado que mostra a invariância de Ω por transformações unitárias pela direita.

Lema 4.9. *Seja $X \in \Omega$ e tome $X' \in J(X)$, ou seja, $X' = XU$ para alguma matriz unitária U. Então, $X' \in \Omega$.*

Prova. Por (4.9), segue que

$$\begin{aligned} X_i'^T S X_j' &= (\sum_{b=1}^{N} X_b^T U_{bi}) S (\sum_{a=1}^{N} X_a^T U_{aj}) \\ &= \sum_{b,a} U_{bi} U_{aj} X_b S X_a \\ &= \sum_{b,a} U_{bi} U_{aj} \delta_{ba} = \sum_{b} U_{bi} U_{bj} = \delta_{ij}. \end{aligned}$$

■

São apresentadas na seqüência algumas considerações fundamentais para os principais resultados deste capítulo, principalmente para o Teorema 4.10, que se refere a condições de regularidade de pontos de Ω (para mais detalhes sobre regularidade, ver [45]).

Para cada $i = 1, \ldots, N$ e cada $j = i, \ldots, N$ defina $\mathcal{R}_{ij} : \mathbb{R}^{K \times N} \to \mathbb{R}$ por

$$\mathcal{R}_{ij}(X) = X_i^T S X_j,$$

e note que Ω, o conjunto viável do problema (4.14), pode ser reescrito sem as restrições redundantes, ou seja,

$$\Omega = \{X \in \mathbb{R}^{K \times N} \mid \mathcal{R}_{ij}(X) = \delta_{ij},\ i = 1, \ldots, N \text{ e } j = i, \ldots, N\}.$$

Mais adiante, as condições clássicas de Lagrange do problema de programação não-linear serão usadas com freqüência. Para que essas condições sejam condições necessárias de otimalidade do problema (4.14), será necessário que se cumpra regularidade [45]. Nesse sentido, é colocado o teorema que se segue.

Teorema 4.10. *Para todo $Y \in \Omega$, os gradientes $\nabla \mathcal{R}_{ij}(Y)$, $i = 1, \ldots, N$ e $j = i, \ldots, N$, são linearmente independentes.*

Prova. Seja $Y \in \Omega$ a matriz formada pelos vetores coluna $\{Y_i\}_{i=1}^{N}$. Organizando as

variáveis na forma vetorial, ou seja, $(Y_1^T, \ldots, Y_N^T) \in \mathbb{R}^{KN}$, segue que, para $i < j$,

$$\nabla \mathcal{R}_{ij}(Y) = \begin{bmatrix} \frac{\partial \mathcal{R}_{ij}(Y)}{\partial Y_1} \\ \vdots \\ \frac{\partial \mathcal{R}_{ij}(Y)}{\partial Y_i} \\ \vdots \\ \frac{\partial \mathcal{R}_{ij}(Y)}{\partial Y_j} \\ \vdots \\ \frac{\partial \mathcal{R}_{ij}(Y)}{\partial Y_N} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ SY_j \\ 0 \\ \vdots \\ SY_i \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{array}{l} \\ \\ \leftarrow \text{ posição } (i-1)K+1 \text{ até } iK \\ \\ \\ \\ \leftarrow \text{ posição } (j-1)K+1 \text{ até } jK, \\ \\ \end{array}$$

se $i = j$,

$$\nabla \mathcal{R}_{ii}(Y) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 2SY_i \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow \text{ posição } (i-1)K+1 \text{ até } iK.$$

Note que $\nabla \mathcal{R}_{ij} \in \mathbb{R}^{NK}$.

Pela Proposição 4.3, a matriz S é simétrica e definida positiva, então o mesmo vale para S^{-1}. Defina a matriz diagonal por blocos $\widetilde{S} \in \mathbb{R}^{NK \times NK}$ por

$$\widetilde{S} = diag(\underbrace{S^{-1}, \ldots, S^{-1}}_{N \ vezes}) = \begin{bmatrix} S^{-1} & & \\ & \ddots & \\ & & S^{-1} \end{bmatrix},$$

que também é SDP (todos autovalores positivos). Considere o produto interno definido pela matriz $\widetilde{S}$. Assim, dados V e $W \in \mathbb{R}^{NK}$, definimos o produto interno $\langle V, W \rangle_{\widetilde{S}} = V^T \widetilde{S} W$.

Sejam i, j, p, q tais que $i \leq j$ e $p \leq q$. Note que $\nabla \mathcal{R}_{ij}(Y)$ e $\widetilde{S} \nabla \mathcal{R}_{pq}(Y)$ são dados,

respectivamente, por

$$\begin{array}{l} \text{posição } (i-1)K+1 \text{ a } iK \rightarrow \\ \\ \text{posição } (j-1)K+1 \text{ a } jK \rightarrow \end{array} \begin{bmatrix} 0 \\ \vdots \\ SY_j \\ \vdots \\ SY_i \\ \vdots \\ 0 \end{bmatrix} \quad \text{e} \quad \begin{bmatrix} 0 \\ \vdots \\ Y_q \\ \vdots \\ Y_p \\ \vdots \\ 0 \end{bmatrix} \begin{array}{l} \\ \leftarrow \text{ posição} \\ (p-1)K+1 \text{ a } pK \\ \\ \leftarrow \text{ posição} \\ (q-1)K+1 \text{ a } qK \end{array} .$$

Será provado que, se $i \neq p$ e/ou $j \neq q$, $\langle \nabla \mathcal{R}_{ij}(Y), \nabla \mathcal{R}_{pq}(Y) \rangle_{\widetilde{S}} = 0$, ou seja, os gradientes são ortogonais com esse produto interno.

Note que as possibilidades de $\langle \nabla \mathcal{R}_{ij}(Y), \nabla \mathcal{R}_{pq}(Y) \rangle_{\widetilde{S}}$ são

(i) 0 se $i \neq p$ e $j \neq q$;

(ii) $Y_j^T S Y_q$ se e somente se $i = p$ e $j \neq q$;

(iii) $Y_j^T S Y_p$ se e somente se $i = q$ e $j \neq p$;

(iv) $Y_i^T S Y_q$ se e somente se $j = p$ e $i \neq q$; e

(v) $Y_i^T S Y_p$ se e somente se $j = q$ e $i \neq p$.

Como as colunas de Y satisfazem $Y_i^T S Y_j = 0$ se $i \neq j$, segue que os itens (ii), (iii), (iv) e (v) são nulos, ou melhor, os gradientes são ortogonais com o produto interno $\langle \cdot, \cdot \rangle_{\widetilde{S}}$ e, por conseqüência, linearmente independentes. ■

De acordo com [45, p. 341], o resultado que segue pode ser demonstrado.

Teorema 4.11. *Suponha que $x^* \in \mathbb{R}^n$ é um minimizador local do problema*

$$\begin{array}{ll} \min & \omega(x) \\ s.a. & z_i(x) = 0 \qquad i = 1, \ldots, m, \end{array}$$

e que o conjunto $\{\nabla z_i(x^*)\}_{i=1}^m$ é linearmente independente. Então, existe um vetor de multiplicadores de Lagrange $\lambda^* \in \mathbb{R}^m$ tal que as seguintes condições, conhecidas como condições de Lagrange, são satisfeitas em (x^*, λ^*):

$$\left\{ \begin{array}{lcll} \nabla_x \ell(x^*, \lambda^*) & = & 0 & \\ z_i(x^*) & = & 0, & i = 1, \ldots, m, \end{array} \right.$$

onde $\ell(x, \lambda) = \omega(x) + \sum_{i=1}^m \lambda_i^T z_i(x)$ é a função lagrangeana.

Um ponto que satisfaz as condições de Lagrange é também chamado de **ponto estacionário**.

Dado $X \in \mathbb{R}^{K\times N}$, defina

$$\langle X_i, \frac{\partial F(X)}{\partial X_j} X_i\rangle = \begin{bmatrix} X_i^T \frac{\partial F(X)}{\partial X_{1j}} X_i \\ X_i^T \frac{\partial F(X)}{\partial X_{2j}} X_i \\ \vdots \\ X_i^T \frac{\partial F(X)}{\partial X_{Kj}} X_i \end{bmatrix}, \tag{4.15}$$

onde

$$\frac{\partial F(X)}{\partial X_{pj}} = \begin{bmatrix} \frac{\partial F(X)_{11}}{\partial X_{pj}} & \cdots & \frac{\partial F(X)_{1K}}{\partial X_{pj}} \\ \vdots & \ddots & \vdots \\ \frac{\partial F(X)_{K1}}{\partial X_{pj}} & \cdots & \frac{\partial F(X)_{KK}}{\partial X_{pj}} \end{bmatrix}.$$

Assim, para $j = 1, \ldots, N$, segue que

$$\frac{\partial E(X)}{\partial X_j} = 2(F(X) + H)X_j + \sum_{i=1}^{N} \langle X_i, \frac{\partial F(X)}{\partial X_j} X_i\rangle. \tag{4.16}$$

A função lagrangeana para o problema (4.14) (ver [45]) será definida por

$$\ell(X, \theta) = E(X) - \sum_{i=1}^{N}\sum_{j=1}^{N} \theta_{ji}(X_i^T S X_j - \delta_{ij}),$$

onde θ é a matriz formada pelos multiplicadores de Lagrange θ_{ij}, que é simétrica pelo fato de S ser simétrica e $\ell(X, \theta)$ ser real. Assim, pelo Teorema 4.11, as condições de Lagrange para o problema (4.14) são

$$\begin{cases} \dfrac{\partial E(X)}{\partial X_j} - \displaystyle\sum_{i=1}^{N} \theta_{ij} S X_i \;\; = 0 \;\; \text{para} \;\; j = 1 \ldots, N, \\ X \in \Omega, \end{cases} \tag{4.17}$$

sendo $\frac{\partial E(X)}{\partial X_j}$ dada por (4.16). As condições de Lagrange são também conhecidas como **condições de Karush-Kuhn-Tucker** ou simplesmente condições **KKT**.

Daqui em diante as notações $B_{*\nu}^{\sigma\lambda}$ e $B_{**}^{\sigma\lambda}$ denotam, respectivamente, o vetor $[B_{i\nu}^{\sigma\lambda}]_i \in \mathbb{R}^K$ e a matriz $[B_{ij}^{\sigma\lambda}]_{ij} \in \mathbb{R}^{K\times K}$.

A seguir são colocados alguns resultados teóricos que simplificarão as condições de Lagrange anteriores, começando por um dos principais lemas deste capítulo, que apresenta as derivadas de primeira ordem do funcional energia.

Lema 4.12. *Tome $X \in \mathbb{R}^{K\times N}$ e sejam $F(X)$ como em (4.8) e $E(X)$ como em (4.11). Então,*

$$2HX_j + \sum_{i=1}^{N} \langle X_i, \frac{\partial F(X)}{\partial X_j} X_i \rangle = 2F(X)X_j$$

e

$$\frac{\partial E(X)}{\partial X_j} = 4F(X)X_j.$$

Prova. Note que

$$\langle X_j, \frac{\partial F(X)}{\partial X_j} X_j \rangle = \begin{bmatrix} X_j^T \frac{\partial F(X)}{\partial X_{1j}} \\ \vdots \\ X_j^T \frac{\partial F(X)}{\partial X_{Kj}} \end{bmatrix} X_j. \tag{4.18}$$

Usando o Lema 4.6, segue que a Equação (4.18) é:

$$\begin{bmatrix} X_j^T \sum_{\sigma=1}^{K} X_{\sigma j}(B_{*1}^{1\sigma} + B_{*1}^{\sigma 1}) & \dots & X_j^T \sum_{\sigma=1}^{K} X_{\sigma j}(B_{*K}^{1\sigma} + B_{*K}^{\sigma 1}) \\ \vdots & \ddots & \vdots \\ X_j^T \sum_{\sigma=1}^{K} X_{\sigma j}(B_{*1}^{K\sigma} + B_{*1}^{\sigma K}) & \dots & X_j^T \sum_{\sigma=1}^{K} X_{\sigma j}(B_{*K}^{K\sigma} + B_{*K}^{\sigma K}) \end{bmatrix} X_j =$$

$$\begin{bmatrix} \sum_{\sigma,\lambda=1}^{K} X_{\sigma j}X_{\lambda j}(B_{\lambda 1}^{1\sigma} + B_{\lambda 1}^{\sigma 1}) & \dots & \sum_{\sigma,\lambda=1}^{K} X_{\sigma j}X_{\lambda j}(B_{\lambda K}^{1\sigma} + B_{\lambda K}^{\sigma 1}) \\ \vdots & \ddots & \vdots \\ \sum_{\sigma,\lambda=1}^{K} X_{\sigma j}X_{\lambda j}(B_{\lambda 1}^{K\sigma} + B_{\lambda 1}^{\sigma K}) & \dots & \sum_{\sigma,\lambda=1}^{K} X_{\sigma j}X_{\lambda j}(B_{\lambda K}^{K\sigma} + B_{\lambda K}^{\sigma K}) \end{bmatrix} X_j =$$

$$\begin{bmatrix} \sum_{\lambda,\sigma,p} X_{\lambda j}X_{\sigma j}X_{pj}(B_{\lambda p}^{1\sigma} + B_{\lambda p}^{\sigma 1}) \\ \vdots \\ \sum_{\lambda,\sigma,p} X_{\lambda j}X_{\sigma j}X_{pj}(B_{\lambda p}^{N\sigma} + B_{\lambda p}^{\sigma N}) \end{bmatrix}. \tag{4.19}$$

Note que, pela Proposição 4.2, para todo $i = 1, \dots, K$, segue que

$$\sum_{\lambda,\sigma,p} X_{\lambda j}X_{\sigma j}X_{pj}(B_{\lambda p}^{i\sigma} + B_{\lambda p}^{\sigma i}) = \sum_{\lambda,\sigma,p} X_{\lambda j}X_{\sigma j}X_{pj}(B_{ip}^{\sigma\lambda} + B_{pi}^{\sigma\lambda}).$$

Assim, a Equação (4.19) torna-se

$$\begin{bmatrix} \sum_{\sigma,\lambda=1}^{K} X_{\sigma j}X_{\lambda j}(B_{11}^{\sigma\lambda}+B_{11}^{\sigma\lambda}) & \dots & \sum_{\sigma,\lambda=1}^{K} X_{\sigma j}X_{\lambda j}(B_{1K}^{\sigma\lambda}+B_{K1}^{\sigma\lambda}) \\ \vdots & \ddots & \vdots \\ \sum_{\sigma,\lambda=1}^{K} X_{\sigma j}X_{\lambda j}(B_{K1}^{\sigma\lambda}+B_{1K}^{\sigma\lambda}) & \dots & \sum_{\sigma,\lambda=1}^{K} X_{\sigma j}X_{\lambda j}(B_{KK}^{\sigma\lambda}+B_{KK}^{\sigma\lambda}) \end{bmatrix} X_j =$$

$$(G(X)+G(X)^T)X_j -$$

$$\begin{bmatrix} \sum_{\substack{b=1\\b\neq j}}^{N}\sum_{\sigma,\lambda=1}^{K} X_{\sigma b}X_{\lambda b}(B_{11}^{\sigma\lambda}+B_{11}^{\sigma\lambda}) & \dots & \sum_{\substack{b=1\\b\neq j}}^{N}\sum_{\sigma,\lambda=1}^{K} X_{\sigma b}X_{\lambda b}(B_{1K}^{\sigma\lambda}+B_{K1}^{\sigma\lambda}) \\ \vdots & \ddots & \vdots \\ \sum_{\substack{b=1\\b\neq j}}^{N}\sum_{\sigma,\lambda=1}^{K} X_{\sigma b}X_{\lambda b}(B_{K1}^{\sigma\lambda}+B_{1K}^{\sigma\lambda}) & \dots & \sum_{\substack{b=1\\b\neq j}}^{N}\sum_{\sigma,\lambda=1}^{K} X_{\sigma b}X_{\lambda b}(B_{KK}^{\sigma\lambda}+B_{KK}^{\sigma\lambda}) \end{bmatrix} X_j =$$

$$(G(X)+G(X)^T)X_j - \begin{bmatrix} \sum_{\substack{b=1\\b\neq j}}^{N}\sum_{\sigma,\lambda,a} X_{\sigma b}X_{\lambda b}X_{aj}(B_{1a}^{\sigma\lambda}+B_{a1}^{\sigma\lambda}) \\ \vdots \\ \sum_{\substack{b=1\\b\neq j}}^{N}\sum_{\sigma,\lambda,a} X_{\sigma b}X_{\lambda b}X_{aj}(B_{Ka}^{\sigma\lambda}+B_{aK}^{\sigma\lambda}) \end{bmatrix}.$$

Chamando a última parcela desta equação de $T(X)$ e usando o fato de que $G(X)$ é simétrica, segue que

$$\langle X_j, \frac{\partial F(X)}{\partial X_j} X_j\rangle = 2G(X)X_j - T(X).$$

Por outro lado,

$$\sum_{\substack{i=1\\i\neq j}}^{N}\langle X_i, \frac{\partial F(X)}{\partial X_j} X_i\rangle = \begin{bmatrix} \sum_{\substack{i=1\\i\neq j}}^{N} X_i^T \frac{\partial F(X)}{\partial X_{1j}} X_i \\ \vdots \\ \sum_{\substack{i=1\\i\neq j}}^{N} X_i^T \frac{\partial F(X)}{\partial X_{Kj}} X_i \end{bmatrix} =$$

$$\begin{bmatrix} \sum_{i\neq j} X_i^T \sum_{\sigma} X_{\sigma j}(B_{**}^{\sigma 1} + B_{**}^{1\sigma})X_i \\ \vdots \\ \sum_{i\neq j} X_i^T \sum_{\sigma} X_{\sigma j}(B_{**}^{\sigma K} + B_{**}^{K\sigma})X_i \end{bmatrix} = \begin{bmatrix} \sum_{i\neq j}\sum_{\sigma,\lambda,a} X_{\sigma j}X_{\lambda i}X_{ai}(B_{\lambda a}^{\sigma 1} + B_{\lambda a}^{1\sigma}) \\ \vdots \\ \sum_{i\neq j}\sum_{\sigma,\lambda,a} X_{\sigma j}X_{\lambda i}X_{ai}(B_{\lambda a}^{\sigma K} + B_{\lambda a}^{K\sigma}) \end{bmatrix}.$$

Usando a Proposição 4.2 e renomeando os índices $i \leftrightarrow b$ e $a \leftrightarrow \sigma$, a equação anterior fica

$$\begin{bmatrix} \sum_{b\neq j}\sum_{\sigma,\lambda,a} X_{aj}X_{\lambda b}X_{\sigma b}(B_{\lambda\sigma}^{a1} + B_{\lambda\sigma}^{1a}) \\ \vdots \\ \sum_{b\neq j}\sum_{\sigma,\lambda,a} X_{aj}X_{\lambda b}X_{\sigma b}(B_{\lambda\sigma}^{aK} + B_{\lambda\sigma}^{Ka}) \end{bmatrix} =$$

$$\begin{bmatrix} \sum_{b\neq j}\sum_{\sigma,\lambda,a} X_{aj}X_{\lambda b}X_{\sigma b}(B_{a1}^{\lambda\sigma} + B_{1a}^{\lambda\sigma}) \\ \vdots \\ \sum_{b\neq j}\sum_{\sigma,\lambda,a} X_{aj}X_{\lambda b}X_{\sigma b}(B_{aK}^{\lambda\sigma} + B_{Ka}^{\lambda\sigma}) \end{bmatrix} = T(X).$$

Então,

$$\begin{aligned} 2HX_j + \sum_{i=1}^{N}\langle X_i, \frac{\partial F(X)}{\partial X_j}X_i\rangle &= 2HX_j + 2G(X)X_j - T(X) + T(X) \\ &= 2(H + G(X))X_j = 2F(X)X_j. \end{aligned}$$

Portanto, por (4.16), segue que

$$\frac{\partial E(X)}{\partial X_j} = 4F(X)X_j,$$

o que prova o resultado. ■

Teorema 4.13. *O sistema KKT do PNL (4.14) é dado por*

$$\begin{cases} F(X)X_j - \sum_{i=1}^{N}\varepsilon_{ij}SX_i = 0 \qquad j = 1,\ldots,N, \\ X \in \Omega, \end{cases}$$

ou na forma matricial

$$\begin{cases} F(X)X - SX\varepsilon = 0 \\ X \in \Omega, \end{cases} \tag{4.20}$$

sendo ε a matriz formada pelos multiplicadores de Lagrange ε_{ij}.

Prova. Por (4.16) e pelo Lema 4.12, segue que as condições KKT de (4.14) são $X \in \Omega$ e

$$4F(X)X_j - \sum_{i=1}^{N} \theta_{ij} S X_i \;\; = \;\; 0$$

para $j = 1, \ldots, N$. Portanto, o resultado segue com $\varepsilon_{ij} = \frac{1}{4}\theta_{ij}$. ■

Corolário 4.14. *Todo ponto fixo Fock satisfaz as condições KKT do PNL (4.14).*

Prova. Seja $C \in \mathbb{R}^{K \times N}$ um ponto fixo Fock. Então, por (4.12),

$$\begin{cases} F(C)C_i = \lambda_i S C_i, & i = 1 \ldots, N, \\ C \in \Omega. \end{cases}$$

Pelo Teorema 4.13, segue que C satisfaz as condições KKT do PNL (4.12) com multiplicadores de Lagrange

$$\varepsilon_{ij} = \begin{cases} \lambda_i & \text{se } i = j \\ 0 & \text{se } i \neq j, \end{cases}$$

ou seja, os multiplicadores de Lagrange associados às restrições de ortogonalidade são nulos. ■

Um fato que deve ser observado é que, se um ponto estacionário do PNL (4.14) possui os multiplicadores de Lagrange associados a restrições de ortogonalidade nulos, ou seja, a matriz ε em (4.20) é diagonal, então pelo Teorema 4.13 este ponto é um ponto fixo Fock.

O objetivo agora é desenvolver uma metodologia para, a partir de um ponto estacionário em que os multiplicadores associados às restrições de ortogonalidade não são nulos, obter um outro ponto estacionário com mesmo valor da função objetivo E que seja um ponto fixo Fock. Para isso será preciso mostrar mais algumas invariâncias por transformações unitárias.

Dada uma matriz $X \in \mathbb{R}^{K \times N}$, será provado que o conjunto de pontos estacionários do problema (4.14) é invariante em $J(X)$. Assim, se X é um ponto estacionário e $X' \in J(X)$, será mostrado que X' também é um ponto estacionário.

Teorema 4.15. *Seja C um ponto estacionário de (4.14) com matriz de multiplicadores de Lagrange $\varepsilon \in \mathbb{R}^{N \times N}$ e tome $C' \in J(C)$, ou seja, $C' = CU$ para alguma matriz unitária U. Então, C' é estacionário do problema (4.14) como matriz de multiplicadores de Lagrange $\varepsilon' = U^T \varepsilon U$ e, ainda mais, $E(C) = E(C')$.*

Prova. Por hipótese,

$$F(C)C - SC\varepsilon = 0.$$

Pelo Lema 4.9, C' é viável. Considerando $\varepsilon' = U^T \varepsilon U$, pela Proposição 4.7, segue que

$$\begin{aligned} F(C')C' - SC'\varepsilon' &= F(C)CU - SCUU^T\varepsilon U \\ &= (F(C)C - SC\varepsilon)U = 0. \end{aligned}$$

Assim, C' é um ponto estacionário. Note que nem ε nem ε' precisam ser matrizes diagonais. Pelo Lema 4.8, segue que $E(C') = E(C)$. ■

Até o momento vários resultados foram introduzidos, estando a maioria deles direta ou indiretamente relacionados ao resultado que vem a seguir. Este resultado pode ser considerado um dos principais deste capítulo. Nele é colocado de maneira implícita um procedimento em que, a partir de um ponto estacionário do PNL, digamos C, é obtido um ponto fixo Fock $C' \in J(C)$ com mesmo valor do funcional energia E. Portanto, esse resultado introduz um procedimento em que o ponto fixo Fock C' pode ser construído a partir do ponto estacionário C. Assim, não se trata somente de um resultado de existência, mas também de um resultado construtivo.

Teorema 4.16. *Seja C um ponto estacionário do problema (4.14) com a matriz de multiplicadores de Lagrange Λ, não necessariamente diagonal. Então, existe um ponto fixo Fock $C' \in J(C)$ com $E(C) = E(C')$ para alguma matriz diagonal Λ'.*

Prova. Primeiramente, será analisado o comportamento dos multiplicadores de Lagrange quando o ponto estacionário C é submetido a transformações unitárias pela direita.

Pela hipótese de C e Λ, segue, pelo Teorema 4.13, que

$$\begin{cases} F(C)C_i - \sum_{j=1}^{N} \lambda_{ji} SC_j = 0, & i = 1, \ldots, N, \\ C \in \Omega. \end{cases}$$

Note que

$$\lambda_{ji} = C_j^T F(C) C_i.$$

Assim, se $\widetilde{C} = CU$ para alguma matriz unitária U, $\widetilde{\lambda}_{ji} = \widetilde{C}_j^T F(\widetilde{C})\widetilde{C}_i$, onde $\widetilde{C}_i = \sum_{b=1}^{N} C_b U_{bi}$, como em (4.9). Usando a Proposição 4.7, segue que

$$\begin{aligned} \widetilde{\lambda}_{ji} &= (\sum_{b=1}^{N} C_b U_{bj})^T F(C) (\sum_{a=1}^{N} C_a U_{ai}) \\ &= \sum_{b,a} U_{bj} U_{ai} C_b^T F(C) C_a = \sum_{b,a} U_{bj} U_{ai} \lambda_{ba} \\ &= U_j^T \Lambda U_i. \end{aligned}$$

Então, chamando $\widetilde{\Lambda}$ a matriz com coeficientes $\widetilde{\lambda}_{ji}$, segue que $\widetilde{\Lambda} = U^T \Lambda U$.

Note que Λ é real simétrica. Assim, pela decomposição espectral, existe uma matriz unitária $V \in \mathbb{R}^{N\times N}$ e uma matriz diagonal Λ' (com autovalores dispostos em ordem crescente) tal que $\Lambda' = V^T \Lambda V$. Tome $C' = CV$, ou seja, $C'_i = \sum_{j=1}^{N} C_j V_{ji}$, então, pelo Teorema 4.15, segue que

$$\begin{cases} F(C')C'_i - \lambda'_i S C'_i &= 0 \qquad i = 1, \ldots, N \\ C' \in \Omega, \end{cases}$$

com $E(C') = E(C)$, ou na forma matricial

$$\begin{cases} F(C')C' - SC'\Lambda' &= 0 \\ C' \in \Omega, \end{cases}$$

ou seja, C' é um ponto fixo Fock. ■

Portanto, fica estabelecido um procedimento para obter uma solução autoconsistente a partir de um ponto estacionário do problema (4.14). Em resumo, esse procedimento consiste em diagonalizar a matriz de multiplicadores de Lagrange Λ associada ao ponto estacionário C, obtendo-se, então, uma matriz unitária V contendo os autovetores de Λ, e assim, como foi provado, $C' = CV$ é um ponto fixo Fock.

Uma vez conhecido um ponto fixo Fock, será possível construir uma aproximação para a função de onda [60], como será apresentado na seção a seguir. Com essa função em mãos, muitas propriedades físicas e químicas do sistema eletrônico podem ser obtidas.

4.4 Invariância da função de onda por matrizes unitárias

Nesta seção é mostrado que dados C', um ponto fixo Fock, e $C \in J(C')$, então, a menos de um sinal, a aproximação para a função de onda obtida por C' é a mesma que a obtida por C. Portanto, pelo Teorema 4.16, um ponto estacionário do PNL (4.14) gera a mesma aproximação para a função de onda que o ponto fixo Fock construído a partir desse ponto estacionário, ou seja, o procedimento colocado no Teorema 4.16 para obter pontos fixos Fock a partir de pontos estacionários pode ser dispensado se o objetivo é obter uma aproximação para a função de onda.

Seja $C \in \mathbb{R}^{K\times N}$. Para cada $a = 1, \ldots, N$, considere as funções espaciais $\phi_a : \mathbb{R}^3 \to \mathbb{R}$

dadas pela combinação linear

$$\phi_a = \sum_{i=1}^{K} C_{ia} g_i. \tag{4.21}$$

Quando C for um ponto estacionário de (4.14), as funções $\{\phi_i\}_{i=1}^{N}$ serão chamadas de **orbitais espaciais**, e, a partir dessas funções, uma aproximação da função de onda, a saber ψ_t, é construída, como será visto à frente.

A seguir, será colocado um resultado que descreve o comportamento das funções espaciais (4.21) quando C é submetido a transformações unitárias pela direita.

Teorema 4.17. *Seja $C \in \mathbb{R}^{K\times N}$ e considere $C' \in J(C)$, ou seja, $C' = CU$ para alguma matriz unitária U. Se $\phi'_a = \sum_{i=1}^{K} C'_{ia} g_i$, então $\phi'_a = \sum_{b=1}^{N} \phi_b U_{ba}$, onde $\phi_b = \sum_{i=1}^{K} C_{ib} g_i$.*

Prova.

$$\begin{aligned} \phi'_a &= \sum_{i=1}^{K} C'_{ia} g_i = \sum_{i=1}^{K} (\sum_{b=1}^{N} C_{ib} U_{ba}) g_i \\ &= \sum_{b=1}^{N} (\sum_{i=1}^{K} C_{ib} g_i) U_{ba} = \sum_{b=1}^{N} \phi_b U_{ba}. \end{aligned}$$

■

Uma vez conhecidos os orbitais espaciais $\{\phi_i\}_{i=1}^{N}$, os orbitais *spins* $\{\chi_i\}_{i=1}^{2N}$ poderão ser calculados por $\chi_a(x_j, w) = \phi_i(x_j)\alpha(w)$, onde $\alpha(w)$ é a função *spin* associada ao elétron i. Neste trabalho é considerada a existência de duas funções *spins*, a saber, α e β. O orbital *spin* χ_a como função das coordenadas do elétron j será denotado por $\chi_a(j)$. Note que para cada função ϕ_i estão associadas duas funções χ_a, uma associada ao *spin* α e a outra ao *spin* β, pelo fato de ser considerado um número par de elétrons. Por isso são calculadas N funções ϕ_i e depois construídas $2N$ funções χ_a.

Tendo em mãos a funções $\{\chi_i\}_{i=1}^{2N}$, a aproximação para a função de onda $\psi_t : \mathbb{R}^{8N} \to \mathbb{R}$ pode ser construída por (ver [26, 53, 60] para mais detalhes):

$$\psi_t = (2N!)^{-1/2} det(A),$$

onde

$$A = \begin{bmatrix} \chi_1(1) & \chi_2(1) & \cdots & \chi_{2N}(1) \\ \chi_1(2) & \chi_2(2) & \cdots & \chi_{2N}(2) \\ \vdots & \vdots & \ddots & \vdots \\ \chi_1(2N) & \chi_2(2N) & \cdots & \chi_{2N}(2N) \end{bmatrix}.$$

Vale a pena observar que o domínio de ψ_t é $\mathbb{R}^{8N}$ pelo fato de serem considerados $2N$

elétrons, apresentando cada elétron três coordenadas espaciais e uma coordenada *spin*, o que resulta em $8N$ variáveis.

Se na Equação (4.21) as funções $\{\phi_i\}_{i=1}^N$ são geradas por um minimizador global do PNL (4.14), digamos, $C^* \in \mathbb{R}^{K\times N}$, então ψ_t busca aproximar a função de onda que descreve o estado fundamental do sistema.

Seguindo a idéia em [60, p. 120], será investigado o comportamento da função de onda ψ_t quando os orbitais *spins* $\{\chi_i\}_{i=1}^{2N}$ são submetidos a transformações unitárias pela direita.

Para cada $a = 1, \ldots, 2N$, seja $\chi'_a = \sum_{b=1}^{2N} \chi_b V_{ba}$, onde $V \in \mathbb{R}^{2N\times 2N}$ satisfaz $VV^T = V^TV = I$. Seja $A' = AV$, então

$$\begin{aligned} A' &= \begin{bmatrix} \chi_1(1) & \ldots & \chi_{2N}(1) \\ \vdots & \ddots & \vdots \\ \chi_1(2N) & \ldots & \chi_{2N}(2N) \end{bmatrix} \begin{bmatrix} V_{11} & \ldots & V_{12N} \\ \vdots & \ddots & \vdots \\ V_{2N1} & \ldots & V_{2N2N} \end{bmatrix} \\ &= \begin{bmatrix} \chi'_1(1) & \ldots & \chi'_{2N}(1) \\ \vdots & \ddots & \vdots \\ \chi'_1(2N) & \ldots & \chi'_{2N}(2N) \end{bmatrix}. \end{aligned}$$

Assim, segue que

$$\psi'_t = (2N!)^{-1/2} det(A') = (2N!)^{-1/2} det(V) det(A) = (\pm)\psi_t. \tag{4.22}$$

Dos pontos de vista físico e químico, ψ_t e ψ'_t são as mesmas, já que para obter a grande maioria das propriedades físicas e químicas é necessário o quadrado da função de onda.

A seguir, será estabelecido um teorema que mostra que a função ψ_t permanece inalterada, a menos de um sinal, quando a matriz de coeficientes C, que gera as funções espaciais $\{\phi_i\}_{i=1}^N$, é submetida a transformações unitárias pela direita. Com esse resultado segue que o ponto estacionário do PNL e o ponto fixo Fock obtido através do procedimento discutido no Teorema 4.16 geram, a menos de um sinal, a mesma aproximação para a função de onda.

Teorema 4.18. *Tome $C \in \mathbb{R}^{K\times N}$ e seja $C' \in J(C)$, ou seja, $C' = CU$ para alguma matriz unitária $U \in \mathbb{R}^{N\times N}$. Para $a = 1, \ldots, N$, considere $\phi_a = \sum_{i=1}^K C_{ia} g_i$ e $\phi'_a = \sum_{i=1}^K C'_{ia} g_i$ e sejam os orbitais $\{\chi_i\}_{i=1}^{2N}$ e $\{\chi'_i\}_{i=1}^{2N}$ gerados pelas funções $\{\phi_i\}_{i=1}^N$ e $\{\phi'_i\}_{i=1}^N$, respectivamente. Então, a menos de um sinal, as funções de onda ψ_t e ψ'_t geradas, respectivamente, por $\{\chi_i\}_{i=1}^{2N}$ e $\{\chi'_i\}_{i=1}^{2N}$ são as mesmas.*

Prova. Pelo Teorema 4.17, segue que $\phi'_a = \sum_{b=1}^N \phi_b U_{ba}$. Como consideramos o caso restrito,

note que

$$A = \begin{bmatrix} \phi_1(1)\alpha & \cdots & \phi_N(1)\alpha & \phi_1(1)\beta & \cdots & \phi_N(1)\beta \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \phi_1(2N)\alpha & \cdots & \phi_N(2N)\alpha & \phi_1(2N)\beta & \cdots & \phi_N(2N)\beta \end{bmatrix}$$

e, assim,

$$A' = \begin{bmatrix} \phi'_1(1)\alpha & \cdots & \phi'_N(1)\alpha & \phi'_1(1)\beta & \cdots & \phi'_N(1)\beta \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \phi'_1(2N)\alpha & \cdots & \phi'_N(2N)\alpha & \phi'_1(2N)\beta & \cdots & \phi'_N(2N)\beta \end{bmatrix} =$$

$$A \begin{bmatrix} U & 0 \\ 0 & U \end{bmatrix} = AV,$$

onde $V \in \mathbb{R}^{2N \times 2N}$ satisfaz $V^T V = VV^T = I$. Portanto, como observado em (4.22), o resultado segue. ■

Uma técnica que busca simplificar as equações vistas anteriormente, utilizada em alguns métodos para calcular estruturas eletrônicas [10, 16], consiste em usar a matriz densidade nas formulações, que nada mais é que uma mudança de variáveis. Uma introdução dessa abordagem é colocada na seção que se segue.

4.5 O cálculo de estruturas eletrônicas em função da matriz densidade

Dada uma matriz $C \in \mathbb{R}^{K \times N}$, será definida de **matriz densidade** a matriz

$$D \equiv D(C) = CC^T \in \mathbb{R}^{K \times K}.$$

Assim,

$$D_{\sigma\lambda} = \sum_{b=1}^{N} C_{\sigma b} C_{\lambda b}$$

e, portanto, por (4.8), a Matriz de Fock F pode ser reescrita em função da matriz densidade D,

$$F(D)_{\mu\nu} = H_{\mu\nu} + \sum_{\sigma,\lambda=1}^{K} D_{\sigma\lambda} B_{\mu\nu}^{\sigma\lambda}. \tag{4.23}$$

Note que, para evitar a introdução de outra notação, está sendo usada F para representar a mesma Matriz de Fock quando escrita em função da matriz densidade $D \in \mathbb{R}^{K \times K}$ e

também da matriz $X \in \mathbb{R}^{K\times N}$.

Com essas considerações, o seguinte resultado pode ser colocado.

Proposição 4.19. *Para cada $i \in \{1,\ldots,q\}$, considere a matriz densidade $D^i \in \mathbb{R}^{K\times K}$ e um escalar $\alpha^i \in \mathbb{R}$ tal que*

$$\sum_{i=1}^{q} \alpha^i = 1.$$

Então,

$$F(\sum_{i=1}^{q} \alpha^i D^i) = \sum_{i=1}^{q} \alpha^i F(D^i).$$

Prova. Por (4.23), tem-se que

$$\begin{aligned} F(\sum_{i=1}^{q} \alpha^i D^i)_{\mu\nu} &= H_{\mu\nu} + \sum_{\sigma,\lambda=1}^{K}\sum_{i=1}^{q} \alpha^i [D^i]_{\sigma\lambda} B_{\mu\nu}^{\sigma\lambda} \\ &= (\sum_{i=1}^{q} \alpha^i) H_{\mu\nu} + \sum_{i=1}^{q} \alpha^i (\sum_{\sigma,\lambda=1}^{K} [D^i]_{\sigma\lambda} B_{\mu\nu}^{\sigma\lambda}) \\ &= \sum_{i=1}^{q} \alpha^i (H_{\mu\nu} + \sum_{\sigma,\lambda=1}^{K} [D^i]_{\sigma\lambda} B_{\mu\nu}^{\sigma\lambda}) \\ &= \sum_{i=1}^{q} \alpha^i F(D^i), \end{aligned}$$

provando o resultado. ■

O teorema que se segue coloca uma condição necessária e suficiente para que uma dada matriz $C \in \mathbb{R}^{K\times N}$, com $C^T S C = I$ e $D = CC^T$, seja um ponto fixo Fock. Um resultado semelhante pode ser encontrado em [42, p. 129].

Teorema 4.20. *Seja $C \in \mathbb{R}^{K\times N}$ tal que $C^T S C = I$ e considere $D = CC^T$. Então,*

$$F(D)DS - SDF(D) = 0$$

se e somente se

$$F(C')C' = SC'\Lambda$$

para alguma matriz diagonal Λ e alguma matriz $C' \in J(C)$.

Prova. Suponha primeiro que $C' \in J(C)$ é um ponto fixo Fock, ou seja,

$$F(C')C' = SC'\Lambda.$$

Seja $U \in \mathbb{R}^{N\times N}$ unitária tal que $C' = CU$. Note que $F(D) \equiv F(C)$ e, pela Proposição 4.7, que diz que F é invariante em $J(C)$, segue que $F(C') = F(C)$. Então, pelo fato de $F(C')$, Λ e S serem matrizes simétricas, segue que

$$\begin{aligned} F(D)DS - SDF(D) &= F(C)CC^TS - SCC^TF(C) \\ &= F(C)CUU^TC^TS - SCUU^TC^TF(C) \\ &= F(C')C'C'^TS - SC'C'^TF(C') \\ &= SC'\Lambda C'^TS - SC'\Lambda C'^TS = 0. \end{aligned}$$

Suponha agora que $C^TSC = I$ e também que $D = CC^T$ satisfaz

$$F(D)DS - SDF(D) = 0.$$

Então,

$$\begin{aligned} F(D)CC^TS - SCC^TF(D) = 0 \quad &\Rightarrow \\ F(D)C - SC\underbrace{C^TF(D)C}_{\widetilde{\Lambda}} = 0 \quad &\Rightarrow \\ F(D)C - SC\widetilde{\Lambda} = 0. \qquad &(4.24) \end{aligned}$$

Seja $U \in \mathbb{R}^{N\times N}$ unitária tal que

$$U^T\widetilde{\Lambda}U = \Lambda$$

é a decomposição espectral de $\widetilde{\Lambda}$. Assim, de (4.24) e pela Proposição 4.7, segue que

$$\begin{aligned} F(D)C = SCU\Lambda U^T \quad &\Rightarrow \\ F(C)CU = SCU\Lambda \quad &\Rightarrow \\ F(CU)CU = SCU\Lambda. & \end{aligned}$$

Então, chamando $C' = CU$, segue que

$$F(C')C' = SC'\Lambda,$$

e pelo Lema 4.9 tem-se que $C'^TSC' = I$. Portanto, C' é um ponto fixo Fock, o que prova o resultado. ■

Na tentativa de simplificar o problema do cálculo de estruturas eletrônicas, alguns métodos consideram o problema em função da matriz densidade [10, 16]. A vantagem dessa abordagem é que o funcional energia (4.11) é quadrático quando reescrito em função dessa matriz, como é mostrado abaixo:

$$\begin{aligned} E(D) &= 2\sum_{i,j=1}^{K} H_{ij}D_{ij} + \sum_{i,j=1}^{K}\sum_{\sigma,\lambda=1}^{K} D_{ij}D_{\sigma\lambda}B_{ij}^{\sigma\lambda} \\ &= \mathrm{tr}(2HD + G(D)D) \\ &= \mathrm{tr}(HD + F(D)D), \end{aligned} \tag{4.25}$$

onde $\mathrm{tr}(A)$ denota o traço da matriz A.

Com isso em mente, o problema do cálculo de estruturas eletrônicas pode ser reformulado de uma maneira diferente, como mostra o teorema que se segue, mas antes considere o seguinte subconjunto de matrizes:

$$\mathcal{M}_K = \{D \in \mathbb{R}^{K\times K} \mid DSD = D, \quad D^T = D \ \text{ e } \ \mathrm{tr}(SD) = N\}. \tag{4.26}$$

Teorema 4.21. *Uma matriz $D \in \mathcal{M}_K$ se e somente se existe uma matriz $C \in \mathbb{R}^{K\times N}$ tal que $D = CC^T$ e $C^TSC = I$.*

Prova. Suponha primeiro que existe $C \in \mathbb{R}^{K\times N}$ tal que $D = CC^T$ e $C^TSC = I$. Então, D é simétrica e

$$DSD = CC^TSCC^T = CC^T = D.$$

Além disso,

$$\begin{aligned} \mathrm{tr}(SD) &= \sum_{i,j=1}^{K} S_{ij}D_{ij} = \sum_{i,j=1}^{K} S_{ij}\sum_{b=1}^{N} C_{ib}C_{jb} \\ &= \sum_{b=1}^{N}\sum_{i,j=1}^{K} C_{ib}S_{ij}C_{jb} \\ &= \sum_{b=1}^{N} \underbrace{C_b^TSC_b}_{1} = N. \end{aligned}$$

Reciprocamente, considere $D \in \mathcal{M}_K$. Do fato de que $D = DSD$, tem-se que $SD = (SD)^2$, ou seja, SD é uma projeção. Portanto, os autovalores de SD são iguais a um (1) ou zero (0). Mas como $\mathrm{tr}(SD) = N$, existem exatamente N autovalores iguais a um (1) e

$(K-N)$ iguais a zero (0). Assim, existem $\{v_1, \ldots, v_N, v_{N+1}, \ldots, v_K\}$ tais que

$$\begin{aligned} SDv_i &= v_i \text{ para } i = 1, \ldots, N, \\ &\text{e} \\ SDv_i &= 0 \text{ para } i = N+1, \ldots, K. \end{aligned}$$

Então, como S é simétrica definida positiva,

$$\begin{aligned} S^{1/2}DS^{1/2}S^{-1/2}v_i &= S^{-1/2}v_i \text{ para } i = 1, \ldots, N, \\ &\text{e} \\ S^{1/2}DS^{1/2}S^{-1/2}v_i &= 0 \text{ para } i = N+1, \ldots, K. \end{aligned}$$

Portanto, chamando $p_i = S^{-1/2}v_i$, segue que

$$\begin{aligned} S^{1/2}DS^{1/2}p_i &= p_i \text{ para } i = 1, \ldots, N, \\ &\text{e} \\ S^{1/2}DS^{1/2}p_i &= 0 \text{ para } i = N+1, \ldots, K. \end{aligned}$$

Mas $S^{1/2}DS^{1/2}$ é uma matriz simétrica e, portanto, $p_i^T p_j = \delta_{ij}$ para $i, j = 1, \ldots, K$. Seja $P \in \mathbb{R}^{K \times N}$ a matriz dada por

$$\begin{bmatrix} | & & | \\ p_1 & \cdots & p_N \\ | & & | \end{bmatrix},$$

então $P^T P = I \in \mathbb{R}^{N \times N}$ e, pela decomposição espectral de $S^{1/2}DS^{1/2}$,

$$D = S^{-1/2}PP^TS^{-1/2} = (S^{-1/2}P)(S^{-1/2}P)^T.$$

Assim, chamando $C = S^{-1/2}P \in \mathbb{R}^{K \times N}$, segue que

$$D = CC^T$$

e

$$C^TSC = P^TS^{-1/2}SS^{-1/2}P = P^TP = I,$$

provando o teorema. ■

Portanto, por esse teorema, o problema de programação não-linear (4.14) para calcular estruturas eletrônicas pode ser reescrito em função da matriz densidade, resultando no

seguinte PNL:

$$\begin{array}{ll} \min & \mathrm{tr}(HD + F(D)D) \\ s.a. & D \in \mathcal{M}_K. \end{array} \tag{4.27}$$

Uma outra conseqüência do Teorema 4.21 é que o problema de encontrar os N autovetores associados aos N menores autovalores generalizados de uma dada matriz $A \in \mathbb{R}^{K\times K}$ simétrica pode ser encarado mediante o seguinte problema de programação não-linear:

$$\begin{array}{ll} \min & \mathrm{tr}(AD) \\ s.a. & D \in \mathcal{M}_K, \end{array} \tag{4.28}$$

sendo a função $\mathrm{tr}(AD)$ linear com relação a D.

O capítulo que se segue apresenta alguns dos mais reconhecidos métodos para calcular estruturas eletrônicas de átomos e moléculas. Esses métodos estão, de alguma forma, baseados na teoria que foi desenvolvida neste capítulo.

Capítulo 5

Alguns Métodos Tradicionais para Calcular Estruturas Eletrônicas

O objetivo deste capítulo é fornecer um breve histórico dos tradicionais métodos para o cálculo de estruturas eletrônicas. Esses métodos estão comumente implementados nos principais pacotes de química computacional disponíveis.

Para começar o capítulo, é abordado o método de ponto fixo SCF ("Self Consistent Field") [53], talvez o mais conhecido. Na seqüência é colocado o método QC-SCF [3, 4], o qual lança mão de técnicas de parametrização exponencial do conjunto das matrizes unitárias, o que resulta, como conseqüência, em um problema de otimização irrestrita. Este é o mais conhecido entre os métodos de convergência localmente quadrática para resolver estruturas eletrônicas. Outro método considerado é o recente ODA ("Optimal Damping Algorithm") [10], que usa a abordagem de matrizes densidades e técnicas de relaxação do conjunto viável para conseguir, sob certas hipóteses, convergência global a pontos estacionários. Como será usada mais à frente, a bem conhecida e eficiente técnica de extrapolação DIIS ("Direct Inversion on the Iterative Subspace") é colocada, a qual, em muitos casos, melhora significativamente a taxa de convergência do cálculo eletrônico. Para finalizar, o recente método TRRH [52] é abordado. Este método lança mão de técnicas de região de confiança para conseguir um método estável no cálculo de estruturas eletrônicas.

5.1 O método de ponto fixo

Um dos métodos mais tradicionais e ainda bastante usado no cálculo de estruturas eletrônicas é o método de ponto fixo de campo autoconsistente, também conhecido como algoritmo de ponto fixo SCF ou método de Roothaan [53].

Alguns resultados teóricos deste método foram recentemente relatados [11], porém, por

se tratar de um algoritmo de ponto fixo [34], nenhuma propriedade de convergência pode ser garantida, sendo bastante freqüentes os casos em que este método diverge. Considerando esse fato, é introduzido no Capítulo 6 um novo algoritmo globalmente convergente para calcular estruturas eletrônicas, o qual pode ser interpretado como uma globalização do método de ponto fixo.

A idéia é avaliar a matriz de Fock no iterando atual e construir o próximo iterado a partir dos autovetores associados aos menores autovalores desta matriz.

O algoritmo de ponto fixo é colocado a seguir.

Algoritmo 5.1 (Ponto fixo). *Tome* $C^0 \in \mathbb{R}^{K\times N}$ *tal que* ${C^0}^T S C^0 = I$. *Faça* $D^0 = C^0 {C^0}^T$ *e* $i \leftarrow 0$.

Passo 1. *Calcule* $F(D^i)$ *como em (4.23).*

Passo 2. *Encontre* $C \in \mathbb{R}^{K\times N}$, *a matriz cujas colunas são os* N *autovetores associados aos* N *menores autovalores generalizados de* $F(D^i)$, *ou seja,*

$$F(D^i)C = SC\Lambda. \tag{5.1}$$

Passo 2. *Faça* $D^{i+1} = CC^T$, $i \leftarrow i+1$, *e volte para o Passo 1.*

Uma característica interessante do método de ponto fixo é que ele pode ser interpretado como uma seqüência de problemas de programação não-linear, dados por (4.28). Assim, a i-ésima iteração é equivalente a resolver

$$\begin{aligned} \min \quad & \mathrm{tr}(F(D^i)D) \\ s.a. \quad & D \in \mathcal{M}_K, \end{aligned}$$

em que $\mathcal{M}_K$ é o conjunto dado por (4.26). Isso se deve ao fato de que resolver esse problema é equivalente a encontrar os N autovetores associados aos N menores autovalores generalizados de $F(D^i)$, como observado anteriormente, no Capítulo 4.

Há um pouco mais de três décadas, foi proposta a abordagem **level-shift SCF** [27], a qual procura contornar as dificuldades de convergência do método de ponto fixo. Esse método introduz um parâmetro μ suficientemente grande no problema de autovalores (5.1) de modo que, na i-ésima iteração, a matriz densidade D^{i+1} não esteja muito longe de D^i. Assim, os subproblemas do método level-shift são dados por problemas de autovalores do tipo

$$(F(D^i) - \mu S D^i S)C = SC\Lambda. \tag{5.2}$$

Note que, quanto maior for o parâmetro μ, mais o problema (5.2) se parece com o problema de autovalores

$$(SD^iS)C = SC\Lambda,$$

cuja solução é C^i, sendo $D^i = C^iC^{iT}$. Entretanto, no algoritmo level-shift, o parâmetro μ é mantido constante em todas as iterações.

No Capítulo 6, é visto que um passo do algoritmo de ponto fixo SCF corresponde a minimizar certa aproximação quadrática do funcional energia sujeito às restrições de S-ortonormalidade.

5.2 O método QC-SCF

O QC-SCF ("Quadratically Convergent SCF"), introduzido por Bacskay [3, 4], é um método para calcular estruturas eletrônicas com taxa de convergência localmente quadrática. Neste método não é necessário calcular autovalores. Entretanto, o método apresenta algumas desvantagens, entre elas o elevado custo computacional para calcular o gradiente e, principalmente, a Hessiana. Mais ainda, trata-se de um método de convergência local geralmente com pequeno raio de convergência. Para atingir convergência quadrática, é necessária a aproximação inicial estar suficientemente próxima da solução.

Usando um resultado conhecido [21, 26], a parametrização exponencial do conjunto das matrizes unitárias, juntamente com o método de Newton, o método QC-SCF aborda o problema do cálculo de estruturas eletrônicas como um problema irrestrito.

O resultado colocado a seguir, cuja prova está baseada em [26, p. 81], apresenta o principal resultado em que este método está fundamentado.

Teorema 5.1. *Considere o subconjunto de matrizes unitárias*

$$\mathcal{U} = \{U \in \mathbb{R}^{K\times K} \mid U^TU = UU^T = I\}.$$

Então, uma matriz $U \in \mathcal{U}$ se e somente se

$$U = \exp(A)$$

para alguma matriz anti-simétrica A.

Prova. Seja $U \in \mathcal{U}$. Então,

$$U^TU = UU^T = I.$$

Assim, pela decomposição espectral, existe $V \in \mathbb{C}^{K\times K}$ com $V^HV = VV^H = I$, $\Sigma \in \mathbb{R}^{K\times K}$ uma matriz diagonal, onde, para cada $j \in \{1, \dots K\}$, $\Sigma_{jj} = \exp(\hat{\imath}\theta_j)$ com $\theta_j \in \mathbb{R}$,

tal que

$$U = V\Sigma V^H = V\exp(\hat{\imath}\Theta)V^H,$$

onde Θ é a matriz diagonal formada pelos θ_j na diagonal, e $\hat{\imath}$ é o parâmetro imaginário tal que $(\hat{\imath})^2 = -1$. A notação V^H significa a hermitiana de V.

Então, usando propriedades da função exponencial para matrizes [25], segue que

$$U = V\exp(\hat{\imath}\Theta)V^H = \exp(\hat{\imath}V\Theta V^H),$$

onde $A \equiv (\hat{\imath}V\Theta V^H)$ satisfaz $A^H = -A$, ou seja, uma matriz anti-simétrica.

Reciprocamente, suponha que existe uma matriz real anti-simétrica A tal que

$$U = \exp(A) \in \mathbb{R}^{K\times K}.$$

Assim, como $A^TA = -AA = AA^T$,

$$\begin{aligned} U^TU &= \exp(A)^T\exp(A) \\ &= \exp(A^T)\exp(A) \\ &= \exp(A^T + A) = \exp(0) = I, \end{aligned}$$

de onde o resultado segue. ■

Por esse teorema, note que o funcional energia E pode ser reescrito em função da parametrização exponencial e, portando, o cálculo de estruturas eletrônicas se torna um problema irrestrito.

Baseado nessa idéia é que Bacskay introduziu o QC-SCF, que é o algoritmo colocado a seguir. Antes é preciso fazer algumas considerações.

Dada uma matriz $Z \in \mathbb{R}^{N\times(K-N)}$, considere a matriz anti-simétrica

$$A(Z) = \begin{bmatrix} 0 & Z \\ -Z^T & 0 \end{bmatrix} \in \mathbb{R}^{K\times K}. \tag{5.3}$$

Portanto, o funcional energia pode ser considerado em função da matriz Z, ou seja,

$$\widehat{E}(Z) = E(C^0\exp(A(Z))),$$

onde E é dado por (4.11) e $C^0 \in \mathbb{R}^{K\times K}$ é tal que ${C^0}^TSC^0 = I$.

Como observado por Bacskay [4] e também por Douady *et al.* [21], a escolha da matriz anti-simétrica $A \equiv A(Z)$ dada por (5.3) remove singularidades na matriz Hessiana de $\widehat{E}(Z)$.

Algoritmo 5.2 (QC-SCF). *Seja* $C^0 \in \mathbb{R}^{K\times K}$ *tal que* ${C^0}^T S C^0 = I$ *e* $Z^0 \in \mathbb{R}^{N\times(K-N)}$. *Faça* $i \leftarrow 0$.

Passo 1. *Calcule* g_i *e* B_i, *que são, respectivamente, a Hessiana e o gradiente de* $\widehat{E}$ *avaliado em* Z^i.

Passo 2. *Resolva o sistema linear*

$$B_i \Delta^i = -g_i.$$

Passo 3. *Faça* $Z^{i+1} = Z^i + \Delta^i$,

$$A^{i+1} = \begin{bmatrix} 0 & Z^{i+1} \\ -(Z^{i+1})^T & 0 \end{bmatrix},$$

$$C^{i+1} = C^i \exp(A^{i+1}),$$

$i \leftarrow i+1$, *e volte para o Passo 1.*

Como mencionado anteriormente, o esforço computacional para construir B_i é extremamente alto. Uma outra desvantagem é o cálculo da função exponencial no Passo 3, que só pode ser feito aproximadamente. Na maioria dos casos, é feita uma aproximação de primeira ordem, ou seja,

$$\exp(A^i) \approx I + A^i,$$

seguida de uma ortogonalização das colunas de $(I + A^i)$, usando, por exemplo, Gram-Schmidt [25].

Na tentativa de contornar a dificuldade de construir a matriz Hessiana, alguns trabalhos têm-se direcionado para os métodos do tipo quasi-Newton, como é observado em [2, 13], sendo usada a correção secante BFGS [45] para atualizar B_{i+1}. Esse tipo de abordagem diminui significativamente o custo computacional, porém perde a propriedade de convergência quadrática.

5.3 O método ODA

O recente algoritmo ODA ("Optimal Damping Algorithm") [10] é o mais simples representante dos algoritmos de restrições relaxadas, também chamados de RCA ("Relaxed Constraints Algorithms"), que nada mais são do que algoritmos que buscam encontrar o estado de mínima energia de uma estrutura eletrônica resolvendo o PNL (4.27), porém com as restrições de idempotência relaxadas, o que deixa o conjunto viável convexo. Então,

definindo,

$$\widetilde{\mathcal{M}}_K = \{D \in \mathbb{R}^{K\times K} \mid DSD \leq D, \text{traço}(SD) = N \text{ e } D = D^T\},$$

os algoritmos RCA buscam resolver o PNL

$$\begin{array}{ll} \min & \text{traço}(HD + F(D)D) \\ s.a. & D \in \widetilde{\mathcal{M}}_K. \end{array} \tag{5.4}$$

A única diferença entre o conjunto $\widetilde{\mathcal{M}}_K$ e o $\mathcal{M}_K$, definido em (4.26), é que a restrição $DSD = D$ é substituída por $DSD \leq D$.

Um resultado que justifica usar o PNL relaxado (5.4), cuja prova pode ser encontrada em [19], diz que a restrição $DSD = D$ é automaticamente satisfeita para todos os pontos estacionários de (5.4).

Para começar a descrição do ODA, considere $\widetilde{D} \in \widetilde{\mathcal{M}}_K$ e $D' \in \mathcal{M}_K$. Então, usando a proposição (4.19), que diz que F é linear em D, tem-se que

$$\begin{aligned} E(\widetilde{D} + \lambda(D' - \widetilde{D})) &= \text{tr}(H(\widetilde{D} - \lambda(D' - \widetilde{D})) + F(\widetilde{D} + \lambda(D' - \widetilde{D}))(\widetilde{D} + \lambda(D' - \widetilde{D}))) \\ &= E(\widetilde{D}) + \lambda(\text{tr}(HD' + F(\widetilde{D})(D' - \widetilde{D}) + F(D')\widetilde{D}) - E(\widetilde{D})) + \\ &\quad +\lambda^2\text{tr}((F(D') - F(\widetilde{D}))(D' - \widetilde{D})) \\ &= E(\widetilde{D}) + 2\lambda(\text{tr}(F(D)(D' - \widetilde{D}))) + \\ &\quad +\lambda^2\text{tr}((F(D') - F(\widetilde{D}))(D' - \widetilde{D})) \end{aligned}$$

e, então, a derivada do funcional energia $E(D)$ na direção D' pode ser calculada:

$$\begin{aligned} S_{\widetilde{D}\to D'} &= \frac{d}{d\lambda}E(\widetilde{D} + \lambda(D' - \widetilde{D}))\Big|_{\lambda=0} \\ &= 2\text{tr}(F(D)(D' - \widetilde{D})). \end{aligned}$$

Assim, a direção de máxima descida, ou seja, a matriz densidade D para a qual $S_{\widetilde{D}\to D'}$ atinge o mínimo, é dada pela solução do problema de otimização

$$\begin{array}{ll} \min & \text{tr}(F(\widetilde{D})D') \\ s.a. & D' \in \mathcal{M}_K. \end{array} \tag{5.5}$$

Portanto, como visto em (4.28), a solução do PNL (5.5) é $D = CC^T$, onde as colunas de $C \in \mathbb{R}^{K\times N}$ são os N autovetores associados aos N menores autovalores generalizados de $F(\widetilde{D})$.

Com essas considerações, o algoritmo ODA pode ser colocado.

Algoritmo 5.3 (ODA). *Tome* $\widetilde{D}^0 \in \widetilde{\mathcal{M}}_K$. *Faça* $i \leftarrow 0$.

Passo 1. *Calcule* $F(\widetilde{D}^i)$.

Passo 2. *Encontre* $C \in \mathbb{R}^{K\times N}$, *a matriz cujas colunas são os* N *autovetores associados aos* N *menores autovalores generalizados* $\{\lambda_j^i\}_{j=1}^N$ *de* $F(\widetilde{D}^i)$.

Passo 3. *Faça* $D^{i+1} = CC^T \in \mathcal{M}_K$.

Passo 4. *Encontre* $\widetilde{D}^{i+1} \in \widetilde{\mathcal{M}}_K$ *solução de*

$$\begin{array}{ll} \min & E(\widetilde{D}^i + t(D^{i+1} - \widetilde{D}^i)) \\ s.a. & t \in [0,1]. \end{array}$$

Passo 5. *Faça* $i \leftarrow i+1$ *e volte para o Passo 1.*

Note que como o funcional energia $E(D)$ é quadrático com relação à matriz densidade D, então, no Passo 4 do algoritmo, $\widetilde{D}^{i+1}$ pode ser calculado analiticamente.

Uma importante propriedade teórica do ODA é que a seqüência $\{E(\widetilde{D}^i)\}_i$ é monótona decrescente e, portanto, sob a hipótese de "**Uniform Well-Posedness**" [10] ou UWP, que supõe $\lambda_{N+1}^i \geq \lambda_N^i + \gamma$ para alguma constante $\gamma > 0$, é possível provar que, para qualquer aproximação inicial $\widetilde{D}^0$, o método converge para um ponto estacionário de (5.4) (ver [19, p. 35] para mais detalhes). Entretanto, a hipótese UWP é uma condição sobre a seqüência gerada pelo algoritmo, o que não é aconselhável quando se pretende obter um rigoroso resultado de convergência.

Uma técnica de extrapolação que busca melhorar o desempenho do método ODA, chamada de EDIIS, é introduzida em [9]. Essa técnica, a qual está baseada na extrapolação DIIS, consiste em, a partir de certa iteração, resolver subproblemas do tipo

$$\begin{array}{ll} \min & E(\widetilde{D}) \\ s.a. & \widetilde{D} = \displaystyle\sum_{p=0}^{t-1} \alpha_p D^p, \quad \alpha_p \geq 0 \ \text{ e } \ \sum_{p=0}^{t-1} \alpha_p = 1, \end{array}$$

onde as matrizes $\{D^p\}_{p=0}^{t-1}$ são os t iterados mais recentes do algoritmo. Porém, esses subproblemas são mais difíceis de resolver do que os subproblemas que aparecem na aceleração DIIS, a qual é colocada na seqüência.

5.4 A extrapolação DIIS

Como visto anteriormente, o método de ponto fixo SCF pode não convergir. Pensando nisso, Pulay [48, 49] introduziu um método, o qual chamou de DIIS ("Direct Inversion in the Iterative Subspace"), com o objetivo de melhorar a taxa de convergência do método e, inclusive, conseguir convergência para os casos onde o ponto fixo falha.

Por se tratar de uma técnica de extrapolação, nenhuma propriedade teórica de convergência pode ser garantida. Mesmo assim, esse método é talvez o mais usado hoje em dia no cálculo de estruturas eletrônicas, mostrado-se bastante eficiente, sendo que comparações indicam que, na maioria dos casos, o método apresenta desempenho superior ao QC-SCF. Uma abordagem híbrida entre o método QC-SCF e o DIIS é considerada por Rendell [51].

Para começar a descrever o procedimento, considere $\{D^i\}_{i=1}^q$ q iterandos do algoritmo de ponto fixo (Algoritmo 5.1).

Pelo Teorema 4.20, uma condição necessária e suficiente para uma matriz $C \in \mathbb{R}^{K\times N}$ ser um ponto fixo Fock é que

$$F(D)DS - SDF(D) = 0,$$

onde $D = CC^T$. Com base nisso, para cada $i = \{1, \ldots, q\}$, defina as matrizes resíduos

$$R^i = F(D^i)D^iS - SD^iF(D^i) \in \mathbb{R}^{K\times K}. \tag{5.6}$$

Matrizes resíduos balanceadas podem ser obtidas transformado cada R^i para a base ortonormal. Neste caso, é equivalente a trocar R^i por $\widetilde{R}^i \equiv S^{-1/2}R^iS^{-1/2}$ e, então,

$$\widetilde{R}^i = \widetilde{F}^i\widetilde{D}^i - \widetilde{D}^i\widetilde{F}^i,$$

onde $\widetilde{F}^i = S^{-1/2}F(D^i)S^{-1/2}$ e $\widetilde{D}^i = S^{1/2}D^iS^{1/2}$.

Uma iteração do método DIIS consiste, portanto, em resolver, após as q iterações do método de ponto fixo, o PNL

$$\begin{aligned} \min \quad & \|\sum_{i=1}^{q} \alpha_i R^i\|_F^2 \\ s.a. \quad & \sum_{i=1}^{q} \alpha_i = 1, \end{aligned}$$

onde $\|\cdot\|_F$ denota a norma de Frobenius [25]. Facilmente, as condições de primeira ordem

deste problema podem ser obtidas, as quais são dadas pelo sistema linear

$$\begin{bmatrix} L_{11} & L_{12} & \cdots & L_{1q} & -1 \\ L_{21} & L_{22} & \cdots & L_{2q} & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ L_{q1} & L_{q2} & \cdots & L_{qq} & -1 \\ -1 & -1 & \cdots & -1 & 0 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_q \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -1 \end{bmatrix}, \tag{5.7}$$

em que $\lambda \in \mathbb{R}$ é o multiplicador de Lagrange e $L_{ij} \equiv \langle R^i, R^j \rangle = \mathrm{tr}(R^i R^{j^T})$. Sendo assim, é considerada a seguinte matriz densidade:

$$D' = \sum_{i=1}^{q} \alpha_i D^i.$$

Note que a restrição

$$\sum_{i=1}^{q} \alpha_i = 1$$

é colocada para que se possa usar a Proposição 4.19 e, então,

$$F' = F(D') = \sum_{i=1}^{q} \alpha_i F(D^i),$$

não havendo, portanto, a necessidade de avaliar F na nova matriz densidade, desde que se conheça $F(D^i)$ para $i = 1, \ldots q$.

Após esses comentários, o algoritmo DIIS pode ser colocado.

Algoritmo 5.4 (DIIS). *Considere $\{D^i\}_{i=1}^q$ q iterados do ponto fixo SCF e $\{F(D^i)\}_{i=1}^q$ suas respectivas matrizes de Fock.*

Passo 1. *Construa, para cada $i = 1, ..., q$, as matrizes resíduos R^i dadas por (5.6).*

Passo 2. *Calcule os elementos L_{ij} e resolva o sistema linear (5.7) obtendo $\{\alpha_i\}_{i=1}^q$.*

Passo 3. *Atualize a Matriz de Fock por*

$$F' = \sum_{i=1}^{q} \alpha_i F(D^i).$$

Passo 4. *Encontre $C \in \mathbb{R}^{K \times N}$, a matriz cujas colunas são os N autovetores associados*

aos N menores autovalores generalizados de F', ou seja,

$$F'C = SC\Lambda.$$

Passo 5. *Faça $D^{q+1} = CC^T$ e calcule $F(D^{q+1})$. Faça $q \leftarrow q+1$ e volte para o Passo 1.*

Note que neste algoritmo é necessário conhecer q iterados do algoritmo de ponto fixo SCF, sendo que, na prática, o algoritmo DIIS começa a ser desempenhado quando, no algoritmo de ponto fixo, a matriz resíduo (5.6) apresente o módulo de seu maior elemento ($\|R^i\|_\infty$) menor que dada tolerância. Esse mesmo valor, $\|R^i\|_\infty$, pode ser usado para declarar que o algoritmo atingiu a precisão desejada. Nas implementações, o parâmetro q é geralmente mantido fixo, assim são consideradas as últimas q matrizes D^i calculadas.

5.5 O algoritmo TRRH

Recentemente, Thørgensen *et al.* [52] introduziram um método baseado em teorias de região de confiança para resolver o problema central deste capítulo. Esse método, por se apoiar também nas teorias e no algoritmo de Roothaan, foi chamado de TRRH ("Trust-Region Roothaan-Hall").

Para começar, considere $\bar{X} \in \mathbb{R}^{K\times N}$ conhecida e $X \in \mathbb{R}^{K\times N}$ qualquer. Assim, considere as matrizes densidades $\bar{D} = \bar{X}\bar{X}^T$ e $D = XX^T$, e defina $E^{RH} : \mathbb{R}^{K\times N} \to \mathbb{R}$ por

$$E^{RH}(D) = 2\mathrm{tr}(F(\bar{D})D) = 2\sum_{i=1}^{K} X_i^T F(\bar{D}) X_i.$$

Então, como $F(\bar{D})$ é simétrica,

$$\nabla_X E^{RH}(D) = 4F(\bar{D})X$$

e, portanto, em $X \equiv \bar{X}$, tem-se que o gradiente de E^{RH} é igual ao da função energia E dada em (4.25),

$$\begin{aligned} E(X) &= \sum_{i=1}^{N} X_i^T (F(X) + H) X_i \\ &= \mathrm{tr}(2HD + G(D)D), \end{aligned}$$

onde $D = XX^T$.

Dessa forma, a função E^{RH} em função de $X \in \mathbb{R}^{K\times N}$ pode ser interpretada como um modelo quadrático para a verdadeira função energia E, sendo a matriz Hessiana de E^{RH} diferente da Hessiana de E em $X \equiv \bar{X}$. Assim, como observado anteriormente, uma iteração do algoritmo de ponto fixo SCF consiste na resolução do PNL,

$$\begin{aligned} \min \quad & E^{RH}(D) \\ s.a. \quad & D \in \mathcal{M}_K, \end{aligned} \tag{5.8}$$

podendo, então, fornecer uma solução que não diminua suficientemente a função energia. Pensando nisso é que o algoritmo TRRH foi introduzido, sendo adaptada a estratégia de região de confiança para o PNL (5.8), buscando atingir o decréscimo suficiente na função energia. Vale a pena lembrar que resolver o problema (5.8) é equivalente a resolvê-lo em função de $X \in \mathbb{R}^{K\times N}$, onde $D = XX^T$, considerando o conjunto viável as restrições de S-ortonormalidade.

Em vez de considerar a restrição usual de região de confiança no problema (5.8), a idéia do algoritmo TRRH é incorporar a restrição linear que se segue, definida pelo produto interno dado pela matriz S no espaço da matriz $\mathbb{R}^{K\times K}$:

$$\langle D, \bar{D}\rangle_S = \mathrm{tr}(DS\bar{D}S) = a\sqrt{N\mathrm{tr}(\bar{D}S\bar{D}S)}, \tag{5.9}$$

onde $a \in [0, 1]$ é uma constante a qual deve ser ajustada em cada iteração e $\mathrm{tr}(\bar{D}S\bar{D}S) = N$, desde que $\bar{D}$ seja idempotente. Note que, se $D = \bar{D}$ (idempotente), então $a = 1$. Também, para $a = 1$, existe uma única matriz D que satisfaz a restrição (5.9), a saber, $\bar{D}$.

No método TRRH, o parâmetro a busca imitar o raio de confiança, e quando próximo da unidade este procura fazer o papel do raio de confiança próximo de zero. Sendo assim, para uma matriz densidade D ser considerada uma candidata no método TRRH, é necessário que

$$a = \frac{\mathrm{tr}(DS\bar{D}S)}{\sqrt{N\mathrm{tr}(\bar{D}S\bar{D}S)}} \tag{5.10}$$

seja maior ou igual que um determinado valor a_{min}. No artigo original é considerado $a_{min} = 0.975$.

Note que o parâmetro a pode ser interpretado como o ajuste para a igualdade na

desigualdade de Cauchy-Schwartz, pois, por esta mesma desigualdade,

$$\begin{aligned}\langle D, \bar{D}\rangle_S &= \mathrm{tr}(DS\bar{D}S) \\ &\leq \sqrt{\langle D, D\rangle_S \langle \bar{D}, \bar{D}\rangle_S} \\ &= \sqrt{\mathrm{tr}(DSDS)\mathrm{tr}(\bar{D}S\bar{D}S)} \\ &= \sqrt{\mathrm{tr}(DS)\mathrm{tr}(\bar{D}S\bar{D}S)} = \sqrt{N\mathrm{tr}(\bar{D}S\bar{D}S)}.\end{aligned}$$

Considerando a restrição (5.9) no PNL (5.8) e introduzindo o multiplicador de Lagrange μ associado a essa nova restrição, a nova função lagrangeana pode ser obtida:

$$\mathcal{L}(X, \widetilde{\Lambda}, \mu) = E^{RH}(D) - 2\mu(\mathrm{tr}(DS\bar{D}S) - a\sqrt{N\mathrm{tr}(\bar{D}S\bar{D}S)}) - 2\mathrm{tr}(\widetilde{\Lambda}(X^TSX - I)),$$

onde $D = XX^T$. Então, impondo as condições de primeira ordem, ou seja, diferenciando $\mathcal{L}$ com relação a X e igualando a zero, é obtido o seguinte problema de autovalores:

$$(F(\bar{D}) - \mu S\bar{D}S)X = SX\widetilde{\Lambda},$$

onde $\widetilde{\Lambda}$ não necessariamente é diagonal. Assim, usando as invariâncias por matrizes unitárias pela direita, como no Capítulo 4, esse problema pode ser reescrito como

$$(F(\bar{D}) - \mu S\bar{D}S)X(\mu) = SX(\mu)\Lambda(\mu), \tag{5.11}$$

sendo $\Lambda(\mu)$ diagonal. Note que esse problema de autovalores é o mesmo que o (5.2), considerado no algoritmo level-shift SCF.

Entretanto, a diferença entre o TRRH e o level-shift está na escolha do parâmetro μ.

Dado um iterando $\bar{D} \in \mathbb{R}^{K\times K}$ do algoritmo TRRH, será definido

$$\mathrm{Ared}(D) = E(\bar{D}) - E(D) \quad \text{e} \quad \mathrm{Pred}(D) = E^{RH}(\bar{D}) - E^{RH}(D).$$

Para cada valor de μ é gerada uma matriz $D \equiv D(\mu) = C(\mu)C(\mu)^T$ e, conseqüentemente, $a \equiv a(\mu)$, conforme a Equação (5.10).

Como sempre, será considerado que os autovalores estão dispostos em ordem crescente na diagonal de $\Lambda(\mu)$. Assim, será definido

$$\lambda^L(\mu) = \Lambda(\mu)_{N+1,N+1}, \quad \lambda^H(\mu) = \Lambda(\mu)_{N,N}$$

e a função $\Delta\lambda : \mathbb{R}^+ \to \mathbb{R}$ por

$$\Delta\lambda(\mu) = \lambda^L(\mu) - \lambda^H(\mu).$$

Para μ suficientemente grande, $\Delta\lambda(\mu)$ é aproximadamente linear em μ. Isso segue da Equação (5.11), pois, para μ suficientemente grande, a dependência de $\Lambda(\mu)$ é praticamente linear em μ [52]. Sendo assim, o parâmetro μ para o problema (5.11) é escolhido no domínio em que $\Delta\lambda$ tem um comportamento linear e, portanto, $\mu \in [\mu_{min}, \infty)$.

Para obter μ_{min}, considere os parâmetros $0 < \mu_1 < \mu_2$ suficientemente grandes e encontre $\Delta\lambda(\mu_1)$ e $\Delta\lambda(\mu_2)$. Portanto, o intervalo em que $\Delta\lambda$ tem um comportamento linear pode ser obtido interpolando esses dois valores. Com isso, o parâmetro μ_{min} é obtido calculando a raiz dessa interpolação linear.

Para determinar o parâmetro μ ideal para (5.11), testa-se primeiro se $a(\mu_{min}) > a_{min}$. Se isso for constatado, a matriz $D(\mu_{min})$ será aceita se satisfaz a condição

$$\text{Ared}(D(\mu_{min})) > 0.$$

Se isso não valer, o parâmetro μ é aumentado gradativamente até que as condições

$$a(\mu) > a_{min} \quad \text{e} \quad \text{Ared}(D(\mu)) > 0 \tag{5.12}$$

sejam satisfeitas.

Para o algoritmo estar bem definido, é preciso mostrar que existe μ suficientemente grande, de modo que as condições (5.12) sejam cumpridas.

Definindo

$$\bar{\Delta} = D(\mu) - \bar{D}$$

e usando a identidade $\text{tr}(AG(B)) = \text{tr}(BG(A))$, válida para matrizes simétricas A e B, segue que

$$\begin{aligned}
\text{Ared}(D(\mu)) &= \text{tr}(2H(\bar{D} - D(\mu)) + G(\bar{D})\bar{D} - G(\bar{\Delta} + \bar{D})(\bar{\Delta} + \bar{D})) \\
&= \text{tr}(2H(\bar{D} - D(\mu)) + 2G(\bar{D})(\bar{D} - D(\mu))) - \text{tr}(G(\bar{\Delta})\bar{\Delta}) \\
&= \text{Pred}(D(\mu)) - \text{tr}(G(\bar{\Delta})\bar{\Delta}).
\end{aligned}$$

Portanto, como $\text{Pred}(D(\mu)) = O(\bar{\Delta})$, $\text{tr}(G(\bar{\Delta})\bar{\Delta}) = O(\bar{\Delta}^2)$, $\bar{\Delta} \to 0$ quando $\mu \to \infty$ e ainda $\text{Pred}(D(\mu)) > 0$ para todo μ suficientemente grande, resulta que as condições (5.12) valem para μ suficientemente grande.

Uma alternativa para evitar o cálculo de $\text{Ared}(D(\mu))$ é sugerida por Thørgensen *et*

al. [52]. Com essa sugestão, uma aproximação para Ared pode ser obtida em função dos iterandos anteriores, evitando, portanto, novas avaliações da função energia. Entretanto, essa aproximação apenas é uma boa representação quando um considerável número de iterações já está disponível.

Embora se trate de um método de região de confiança, nenhuma propriedade de convergência é garantida pelo método TRRH.

Buscando melhorar a convergência, é sugerida por Thørgensen *et al.* [52] uma estratégia para acelerar a convergência do TRRH, a qual foi chamada de TRDSM ("Trust-Region Density-Subspace Minimization"). Essa abordagem está também baseada em região de confiança e na aceleração EDIIS [9].

Capítulo 6

Um Algoritmo Globalmente Convergente para o Cálculo Eletrônico

Neste capítulo é introduzido um novo algoritmo globalizado numericamente viável para o cálculo de estruturas eletrônicas de átomos e moléculas [23]. Como comentado anteriormente, este método apresentará uma surpreendente ligação com o bem conhecido método SCF, podendo ser visto como uma globalização deste.

O algoritmo é um caso particular do Algoritmo 3.1, introduzido no Capítulo 3, o que garantirá boa definição e convergência global a pontos estacionários. Uma característica fundamental para a viabilidade da implementação é que, pelo fato de usar o parâmetro de regularização para penalizar a região de confiança, a resolução dos subproblemas serão equivalentes a encontrar uma decomposição espectral, deixando o algoritmo atrativo do ponto de vista numérico.

6.1 Considerações gerais

Algoritmos globalmente convergentes eficientes para calcular estruturas eletrônicas não são muito freqüentes, sendo do conhecimento do autor desta tese somente a existência do método ODA [10], o qual supõe condições sobre a sequência de iterados para conseguir a prova de convergência, como apresentado no Capítulo 5.

Para facilitar o entendimento, será introduzido antes um algoritmo intermediário, o qual será um caso particular do Algoritmo 3.1. Para isso, considere o problema (3.1),

$$\begin{array}{ll} \min & f(x) \\ s.a. & x \in \Gamma, \end{array} \tag{6.1}$$

onde $f : \mathbb{R}^n \to \mathbb{R}$ é uma função diferenciável sobre algum conjunto aberto contendo $\Gamma \subset \mathbb{R}^n$, sendo este um conjunto fechado arbitrário.

Daqui em diante, o gradiente de f será denotado por g. Assim, $\nabla f(x) = g(x)$.

Com as consiedrações acima, o algoritmo que segue pode ser estabelecido. Sua semelhança com o Algoritmo Modelo (Algoritmo 3.1) pode facilmente ser constatada, em que $B_\rho^k \equiv H_k$ e $A_\rho^k \equiv 0$ para $\rho = 0$ e, $B_\rho^k \equiv 0$ e $A_\rho^k \equiv \sigma^k A$ para os demais valores de ρ, sendo $A \in \mathbb{R}^{n\times n}$ uma matriz simétrica e definida positiva arbitrária.

Algoritmo 6.1 (Algoritmo intermediário). *Tome $x^0 \in \Gamma$. Escolha $\rho_b > 0 \in \mathbb{R}$, ζ_1, $\zeta_2 \in \mathbb{R}$ com $1 < \zeta_1 < \zeta_2 < +\infty$, $\sigma_{min}, \sigma_{max} \in \mathbb{R}$ com $0 < \sigma_{min} < \sigma_{max} < +\infty$, $\beta_1 \in (0, \frac{1}{2}]$ e $A \in \mathbb{R}^{n\times n}$ simétrica e definida positiva.*
Faça $k \leftarrow 0$.

Passo 1. *Calcule f_k, $g_k \equiv g(x^k)$ e escolha $H_k \in \mathbb{R}^{n\times n}$ simétrica.*

Passo 2. *Defina*

$$R^k(x) = g_k^T(x - x^k) + \frac{1}{2}(x - x^k)^T H_k(x - x^k)$$

e encontre x_R^k solução global de

$$\begin{array}{ll} \min & R^k(x) \\ s.a. & x \in \Gamma. \end{array} \tag{6.2}$$

Se $R^k(x_R^k) = 0$, pare! x_R^k é um ponto estacionário de (3.1).

Passo 3. *Defina $ared_R = f(x^k) - f(x_R^k)$ e $pred_R = -R^k(x_R^k)$.*

Se

$$\frac{ared_R}{pred_R} \geq \beta_1,$$

escolha $x^{k+1} \in \Gamma$ tal que $f(x^{k+1}) \leq f(x_R^k)$, $\rho^k = 0$, $k \leftarrow k + 1$ e volte para o Passo 1.

Senão, escolha $\rho \in (0, \rho_b]$, $\sigma^k \in [\sigma_{min}, \sigma_{max}]$ e vá para o Passo 4.

Passo 4. *Defina*

$$Q^k(x) = g_k^T(x - x^k) + \frac{1}{2}\rho\sigma^k(x - x^k)^T A(x - x^k)$$

e encontre x_Q^k solução global de

$$\begin{array}{ll} \min & Q^k(x) \\ s.a. & x \in \Gamma. \end{array} \tag{6.3}$$

Se $Q^k(x_Q^k) = 0$, pare! x_Q^k é um ponto estacionário de (3.1).

Passo 5. *Defina $ared_Q = f(x^k) - f(x_Q^k)$ e*

$$pred_Q = -g_k^T(x_Q^k - x^k) - \frac{1}{2}\sigma^k(x_Q^k - x^k)^T A(x_Q^k - x^k).$$

Se

$$\frac{ared_Q}{pred_Q} \geq \beta_1,$$

escolha $x^{k+1} \in \Gamma$ tal que $f(x^{k+1}) \leq f(x_Q^k)$, $\rho^k = \rho$, $k \leftarrow k+1$ e volte para o Passo 1.

Senão, escolha $\rho_{novo} \in [\zeta_1\rho, \zeta_2\rho]$, faça $\rho = \rho_{novo}$ e volte para o Passo 4.

Claramente, esse algoritmo é um caso particular do Algoritmo 3.1. Com efeito, sob as mesmas hipóteses, ele está bem definido e todo ponto de acumulação é um ponto estacionário de (6.1).

Nos Passos 3 e 5 do Algoritmo 6.1, a escolha de x^{k+1} poderia ser x_R^k ou x_Q^k, dependendo de qual satisfaz a condição de decréscimo suficiente, no entanto escolher x^{k+1}, de modo que o valor da função objetivo f seja menor do que no ponto que satisfaz a condição de decréscimo suficiente, prepara o algoritmo para incorporar esquemas de aceleração em implementações práticas, como, por exemplo, o método DIIS.

Na seqüência, começará a ser introduzido o algoritmo para o cálculo de estruturas eletrônicas. Para isso, será necessário fazer algumas observações.

De agora em diante, neste capítulo, toda matriz será denotada em letras maiúsculas, como, por exemplo, $X \in \mathbb{R}^{K\times N}$. Quando esta for vista como um vetor de $\mathbb{R}^{NK}$, será denotada por letra minúscula, por exemplo x, sendo que $X_i \in \mathbb{R}^K$, algumas vezes também denotada por $[X]_i$, denotará o i-ésimo componente vetorial de $x \in \mathbb{R}^{NK}$ e a i-ésima coluna de X. Assim, a matriz

$$X = \begin{bmatrix} | & & | \\ X_1 & \cdots & X_N \\ | & & | \end{bmatrix} \in \mathbb{R}^{K\times N},$$

quando vista como um vetor, será denotada por

$$x = \begin{bmatrix} X_1 \\ \vdots \\ X_N \end{bmatrix} \in \mathbb{R}^{NK}.$$

Note que, com esta consideração, os resultados teóricos obtidos nas seções anteriores, principalmente os referentes ao Capítulo 4, permanecem inalterados, já que todos resultados estão associados às componentes vetoriais X_i.

As notações $x^k \in \mathbb{R}^{NK}$ e X^k serão reservadas para denotar o k-ésimo termo de uma seqüência, respectivamente, para vetor e matriz.

De agora em diante, a função $E : \mathbb{R}^{KN} \to \mathbb{R}$ denotará o funcional energia como em (4.11), no Capítulo 4. Então,

$$E(x) = \sum_{i=1}^{N} X_i^T (F(X) + H) X_i,$$

onde as matrizes H e $F(X)$, esta última a Matriz de Fock, são definidas, respectivamente, em (4.5) e (4.8).

Como conseqüência, para cada $i = 1 \ldots, N$, segue pelo Lema 4.12 que

$$\frac{\partial E(X)}{\partial X_i} = 4F(X)X_i,$$

ou seja, o gradiente da E, denotado daqui em diante por g, é dado por

$$g(x) \equiv \nabla E(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial X_1} \\ \vdots \\ \frac{\partial f(x)}{\partial X_N} \end{bmatrix} = 4 \begin{bmatrix} F(X)X_1 \\ \vdots \\ F(X)X_N \end{bmatrix}.$$

Então, definido-se $\mathcal{H}_1 : \mathbb{R}^{NK} \to \mathbb{R}^{NK \times NK}$ por

$$\mathcal{H}_1(X) = diag(\underbrace{F(X), \ldots, F(X)}_{N \ vezes}) = \begin{bmatrix} F(X) & & \\ & \ddots & \\ & & F(X) \end{bmatrix}, \tag{6.4}$$

segue que

$$g(x) = 4\mathcal{H}_1(X)x. \tag{6.5}$$

Como visto no Capítulo 4, o cálculo de estruturas eletrônicas consiste em resolver o problema (4.14),

$$\begin{array}{ll} \min & E(x) \\ s.a. & x \in \Omega, \end{array} \tag{6.6}$$

onde, como sempre, Ω é o conjunto compacto dado por

$$\Omega = \{x \in \mathbb{R}^{KN} \mid X_i^T S X_j = \delta_{ij},\ i, j = 1, \ldots, N\},$$

ou, na forma matricial,

$$\Omega = \{X \in \mathbb{R}^{K\times N} \mid X^T S X = I\},$$

sendo S definida em (4.7) e $I \in \mathbb{R}^{N\times N}$ a matriz identidade.

O objetivo deste capítulo é usar o Algoritmo 6.1 para resolver o PNL (6.6) e, então, daqui em diante, $\Gamma \equiv \Omega$ e $f \equiv E$.

A seguir é mostrado que o algoritmo de ponto fixo SCF pode ser interpretado como uma seqüência de subproblemas do tipo (6.2) para uma matriz H_k devidamente escolhida.

6.2 Um passo do ponto fixo SCF como um problema de programação não-linear

Dado um iterando $x^k \in \mathbb{R}^{NK}$, o método proposto consiste em encontrar x^{k+1} por meio do Algoritmo 6.1. Portanto, é necessário resolver o subproblema (6.2),

$$\begin{aligned} \min \quad & R^k(x) \\ s.a. \quad & x \in \Omega, \end{aligned}$$

onde

$$R^k(x) = (x - x^k)^T g(x^k) + \frac{1}{2}(x - x^k)^T H_k (x - x^k)$$

é assumida para ser uma representação quadrática de E em torno de x^k, e a matriz $H_k \in \mathbb{R}^{NK\times NK}$, simétrica, representando uma aproximação para a matriz Hessiana de E avaliada em x^k.

Para cada $k \in \mathbb{N}$ no Passo 1 do Algoritmo 6.1, escolha $H_k = 4\mathcal{H}_1(X^k)$, onde $\mathcal{H}_1(X)$ é definida em (6.4).

Sendo assim, tem-se, pela Equação (6.5) e definição de $\mathcal{H}_1$ (Equação (6.4)), que

$$\begin{aligned} R^k(x) &= (x - x^k)^T g(x^k) + \frac{1}{2}(x - x^k)^T H_k (x - x^k) \\ &= 4(x - x^k)^T \mathcal{H}_1(X^k) x^k + 2(x - x^k)^T \mathcal{H}_1(X^k)(x - x^k) \\ &= 4x^T \mathcal{H}_1(X^k) x^k - 4{x^k}^T \mathcal{H}_1(X^k) x^k + \\ &\quad 2[x^T \mathcal{H}_1(X^k) x - 2x^T \mathcal{H}_1(X^k) x^k + {x^k}^T \mathcal{H}_1(X^k) x^k] \\ &= 2x^T \mathcal{H}_1(X^k) x - 2{x^k}^T \mathcal{H}_1(X^k) x^k \\ &= 2\sum_{i=1}^{N} X_i^T F(X^k) X_i - c_k, \end{aligned}$$

onde $c_k = 2{x^k}^T \mathcal{H}_1(X^k) x^k$ é uma constante. Portanto, resolver o subproblema (6.2) com

$H_k = 4\mathcal{H}_1(x^k)$ é equivalente a resolver

$$\begin{aligned} \min \quad & \frac{1}{2}\sum_{i=1}^{N} X_i^T F(X^k) X_i \\ s.a. \quad & X \in \Omega, \end{aligned}$$

cujo minimizador global deste problema, como é conhecido da Álgebra Linear [56], são os N autovetores associados aos N menores autovalores de $F(X^k)$ (para uma demonstração detalhada deste fato, ver o Apêndice A).

Dessa maneira, resolver (6.2) com $H_k = 4\mathcal{H}_1(X^k)$ consiste em desempenhar uma iteração de ponto fixo SCF a partir de um iterado x^k.

É interessante observar que, ao definir-se

$$\langle \frac{\partial F(X)}{\partial X_a}, X_b \rangle = \sum_{i=1}^{K}\sum_{j=1}^{K} X_{ja} X_{ib} (B_{i*}^{*j} + B_{i*}^{j*}),$$

sendo B_{i*}^{*j} a matriz $[B_{ip}^{qj}]_{pq} \in \mathbb{R}^{K\times K}$, tem-se que a matriz Hessiana de E, denotada por $\mathcal{H}$, é dada por

$$\mathcal{H}(X) = 4\mathcal{H}_1(X) + \mathcal{H}_2(X),$$

onde $\mathcal{H}_1(X)$ é definida em (6.4) e

$$\mathcal{H}_2(X) = \begin{bmatrix} \langle \frac{\partial F(X)}{\partial X_1}, X_1 \rangle & \cdots & \langle \frac{\partial F(X)}{\partial X_N}, X_1 \rangle \\ \langle \frac{\partial F(X)}{\partial X_1}, X_2 \rangle & \cdots & \langle \frac{\partial F(X)}{\partial X_N}, X_2 \rangle \\ \vdots & \ddots & \vdots \\ \langle \frac{\partial F(X)}{\partial X_1}, X_N \rangle & \cdots & \langle \frac{\partial F(X)}{\partial X_N}, X_N \rangle \end{bmatrix} \in \mathbb{R}^{NK\times NK},$$

ou seja, a matriz H_k é uma parcela da verdadeira Hessiana de E.

Com essas considerações, para implementar o algoritmo, é necessário resolver o subproblema (6.3), chamado aqui de **subproblema fácil**. A seguir é mostrado que, para uma matriz A simétrica definida positiva (SDP) devidamente escolhida, esse subproblema se resume em encontrar a decomposição em valores singulares de uma determinada matriz, que, por sua vez, pode ser encarado novamente como um problema de autovalores. Portanto, o método que será proposto, tema central desta seção, consistirá de uma seqüência de decomposição espectral.

6.3 O subproblema fácil como um problema de valores singulares

O objetivo desta seção é mostrar que, para uma determinada matriz A simétrica e definida positiva, resolver o subproblema (6.3) é equivalente a encontrar a decomposição em valores singulares de uma dada matriz.

Para isso, considere

$$A = diag(\underbrace{S, \ldots, S}_{N \ vezes}) = \begin{bmatrix} S & & \\ & \ddots & \\ & & S \end{bmatrix} \in \mathbb{R}^{NK \times NK}, \tag{6.7}$$

onde S é a matriz definida em (4.7).

Dado um iterando x^k, note que o subproblema (6.3) pode ser reescrito como

$$\begin{array}{ll} \min & \dfrac{2}{\sigma^k \rho} g(x^k)^T (x - x^k) + (x - x^k)^T A (x - x^k) \\ s.a. & x \in \Omega, \end{array} \tag{6.8}$$

onde

$$g_k = g(x^k) = 4 \begin{bmatrix} F(X^k)[X^k]_i \\ \vdots \\ F(X^k)[X^k]_N \end{bmatrix}.$$

Considere a seguinte troca de variável em $\mathbb{R}^{NK}$:

$$y = A^{1/2} x.$$

Conseqüentemente,

$$x = A^{-1/2} y, \quad y^k = A^{1/2} x^k \quad \text{e} \quad x^k = A^{-1/2} y^k.$$

Além disso, escrevendo

$$Y_i = S^{1/2} X_i \quad \text{e } [Y^k]_i = S^{1/2} [X^k]_i \text{ para todo } i = 1, \ldots, N,$$

tem-se que

$$y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_N \end{bmatrix}, y^k = \begin{bmatrix} [Y^k]_1 \\ \vdots \\ [Y^k]_N \end{bmatrix} \in \mathbb{R}^{NK}$$

ou, na forma matricial,

$$Y = \begin{bmatrix} | & & | \\ Y_1 & \cdots & Y_N \\ | & & | \end{bmatrix}, \; Y^k = \begin{bmatrix} | & & | \\ [Y^k]_1 & \cdots & [Y^k]_N \\ | & & | \end{bmatrix} \in \mathbb{R}^{K\times N}.$$

Com essas mudanças de variáveis, o problema (6.8) torna-se

$$\begin{array}{ll} \min & \dfrac{2}{\sigma^k\rho} g_k^T A^{-1/2}(y - y^k) + \|y - y^k\|^2 \\ s.a. & y_i^T y_j = \delta_{ij}, \quad i,j = 1,\ldots,N. \end{array}$$

Chamando

$$\bar{g}_k = \frac{1}{\sigma^k\rho} A^{-1/2} g_k,$$

este subproblema é equivalente a

$$\begin{array}{ll} \min & 2\bar{g}_k^T(y - y^k) + \|y - y^k\|^2 \\ s.a. & y_i^T y_j = \delta_{ij}, \quad i,j = 1,\ldots,N, \end{array}$$

que, por sua vez, é equivalente a

$$\begin{array}{ll} \min & \|y - (y^k - \bar{g}_k)\|^2 \\ s.a. & y_i^T y_j = \delta_{ij}, \quad i,j = 1,\ldots,N. \end{array}$$

Portanto, denotando

$$z^k = y^k - \bar{g}_k = \begin{bmatrix} [Z^k]_1 \\ \vdots \\ [Z^k]_N \end{bmatrix} \in \mathbb{R}^{NK}$$

ou, na forma matricial,

$$Z^k = \begin{bmatrix} | & & | \\ [Z^k]_1 & \cdots & [Z^k]_N \\ | & & | \end{bmatrix} \in \mathbb{R}^{K\times N},$$

o problema (6.3) se resume em resolver,

$$\begin{array}{ll} \min & \|Y - Z^k\|_F^2 \\ s.a. & Y^T Y = I, \end{array} \tag{6.9}$$

em que, como sempre, $\|\cdot\|_F$ denota a norma de Frobenius [25].

Considere a decomposição SVD de Z^k,

$$Z^k = U^k \Sigma^k {V^k}^T,$$

onde $U^k \in \mathbb{R}^{K\times K}$ e $V^k \in \mathbb{R}^{N\times N}$ são matrizes unitárias e $\Sigma^k \in \mathbb{R}^{K\times N}$ é diagonal. Um fato conhecido da álgebra linear [25] é que $\|QC\|_F = \|CQ\|_F = \|C\|_F$ sempre que Q é uma matriz unitária, assim o problema (6.9) é equivalente a

$$\begin{array}{ll} \min & \|{U^k}^T Y V^k - \Sigma^k\|_F^2 \\ s.a. & Y^T Y = I. \end{array} \tag{6.10}$$

Defina $W = {U^k}^T Y V^k$. Então, através de um cálculo rápido, mostra-se que $W^T W = I$ se e somente se $Y^T Y = I$, e portanto a solução do problema (6.10) é

$$Y = U^k W {V^k}^T,$$

onde W resolve

$$\begin{array}{ll} \min & \|W - \Sigma^k\|_F^2 \\ s.a. & W^T W = I. \end{array}$$

Claramente, a solução deste problema é a matriz diagonal em $\mathbb{R}^{K\times N}$, que tem os elementos da diagonal todos iguais a um (1). De agora em diante, esta matriz será chamada de $I_{K\times N}$. Então, a solução Y de (6.9) é dada por

$$Y = U^k I_{K\times N} {V^k}^T.$$

Portanto, escrevendo

$$U^k = \begin{bmatrix} | & & | & & | \\ [u^k]_1 & \cdots & [u^k]_N & \cdots & [u^k]_K \\ | & & | & & | \end{bmatrix} \text{ e } V^k = \begin{bmatrix} | & & | \\ [v^k]_1 & \cdots & [v^k]_N \\ | & & | \end{bmatrix},$$

segue que

$$Y = [u^k]_1 [v^k]_1^T + \cdots + [u^k]_N [v^k]_N^T.$$

Finalmente, a solução do subproblema (6.3) é

$$X = S^{-1/2} Y.$$

Note que os vetores singulares à direita $[v^k]_1, \ldots, [v^k]_N$ formam uma base dos autovetores unitários de ${Z^k}^T Z^k$, onde, como sempre, é assumido que os autovalores estão em

ordem decrescente. Para cada autovalor não-nulo desta matriz com autovetor $[v^k]_i$, o correspondente vetor singular à esquerda $[u^k]_i$ pode ser calculado como

$$\begin{aligned} [u^k]_i &= \frac{Z^k[v^k]_i}{\|Z^k[v^k]_i\|} \\ &= \frac{Z^k[v^k]_i}{\sqrt{\mu_i}}, \end{aligned} \tag{6.11}$$

onde μ_i é o autovalor de ${Z^k}^T Z^k$ associado ao autovetor $[v^k]_i$.

Se necessário, o conjunto $\{[u^k]_1, \ldots, [u^k]_N\}$ deve ser completado por algum processo de ortogonalização.

6.4 Um algoritmo globalmente convergente para calcular estruturas eletrônicas

Com todas as considerações feitas nas seções anteriores, o algoritmo globalmente convergente para calcular estruturas eletrônicas de átomos e moléculas, chamado por este trabalho de **TR**, pode ser apresentado.

O parâmetro σ^k é o parâmetro espectral adotado em [8, 50], a saber,

$$\sigma^k = \max\left\{\sigma_{min}, \min\left\{\sigma_{max}, \frac{(x^k - x^{k-1})^T(g_k - g_{k-1})}{(x^k - x^{k-1})^T A(x^k - x^{k-1})}\right\}\right\},$$

o qual busca representar a Hessiana na direção $(x^k - x^{k-1})$. O objetivo desta escolha é deixar a iteração com aspecto newtoniano.

Na sequência, o Algoritmo TR é estabelecido.

Algoritmo 6.2 (Algoritmo TR). *Tome $x^0 \in \Omega$. Escolha $\rho_b > 0$, ζ_1, $\zeta_2 \in \mathbb{R}$ com $1 < \zeta_1 < \zeta_2 < +\infty$, $\sigma_{min}, \sigma_{max} \in \mathbb{R}$ com $0 < \sigma_{min} < \sigma_{max} < +\infty$, $\beta_1 \in (0, \frac{1}{2}]$ e considere $A \in \mathbb{R}^{NK \times NK}$ dada por (6.7).*
Faça $k \leftarrow 0$ e $\sigma^0 = 1$.
Calcule, para $\mu, \nu, a, b = \{1, \ldots, K\}$,

$$B^{ab}_{\mu\nu} = 2(\mu\nu|ab) - (\mu b|a, \nu).$$

Calcule $H, S \in \mathbb{R}^{K \times K}$ definidos, respectivamente, em (4.5) e (4.7).
Calcule a decomposição espectral de S,

$$S = Q_S D_S Q_S^T,$$

onde $Q_S \in \mathbb{R}^{K\times K}$ é unitária e $D_S \in \mathbb{R}^{K\times K}$ diagonal.
Defina

$$S^{1/2} = Q_S D_S^{1/2} Q_S^T \quad e \quad S^{-1/2} = Q_S D_S^{-1/2} Q_S^T.$$

Passo 1. *Ponto fixo SCF: encontre $X_c^k \in \mathbb{R}^{K\times N}$ (ou $x_c^k \in \mathbb{R}^{NK}$), a matriz (vetor) cujas colunas são os N autovetores associados aos N menores autovalores generalizados de $F(X^k)$ com relação à matriz S, ou seja, encontre $X_c^k \in \mathbb{R}^{K\times N}$ e $\Lambda^k \in \mathbb{R}^{N\times N}$ diagonal tal que*

$$F(X^k)X_c^k = SX_c^k\Lambda^k.$$

Se $X_c^k = X^k$, então X_c^k é um ponto fixo Fock.

Defina

$$g_k = \nabla f(x^k) = 4\mathcal{H}_1(X^k)x^k \text{ e } H_k = \mathcal{H}_1(X^k),$$

onde $\mathcal{H}_1$ é dada por (6.4).

Se $k \geq 1$, defina

$$\sigma^k = \max\left\{\sigma_{min}, \min\left\{\sigma_{max}, \frac{(x^k - x^{k-1})^T(g_k - g_{k-1})}{(x^k - x^{k-1})^T A(x^k - x^{k-1})}\right\}\right\}.$$

Escolha $\rho \in (0, \rho_b]$.

Passo 2. *Defina $ared = f(x^k) - f(x_c^k)$ e*

$$pred = -g_k^T(x_c^k - x^k) - \frac{1}{2}(x_c^k - x^k)^T H_k(x_c^k - x^k).$$

Se $ared \geq \beta_1 pred$, escolha $x^{k+1} \in \Gamma$ tal que $f(x^{k+1}) \leq f(x_c^k)$, $k \leftarrow k+1$ e volte para o Passo 1.

Senão, escolha $\rho_{novo} \in [\zeta_1\rho, \zeta_2\rho]$, faça $\rho = \rho_{novo}$ e vá para o Passo 3.

Passo 3. *Resolver o problema fácil.*

Calcule

$$Z^k = S^{1/2}X^k - \frac{4}{\sigma^k\rho}S^{-1/2}F(X^k)X^k \in \mathbb{R}^{K\times N}$$

e encontre $[v^k]_1, \ldots, [v^k]_N \in \mathbb{R}^N$ uma base de autovetores ortonormais de ${Z^k}^T Z^k \in \mathbb{R}^{N\times N}$.

Para todo $[v^k]_i$ *correspondendo a um autovalor não nulo de* ${Z^k}^T Z^k$*, calcule* $[u^k]_i$ *como em (6.11),*

$$[u^k]_i = \frac{Z[v^k]_i}{\|Z[v^k]_i\|}.$$

Se necessário, complete o conjunto $[u^k]_1, \ldots, [u^k]_N$ *de modo que todos estes vetores sejam unitários e ortogonais entre si.*

Passo 4. *Defina*

$$U^k = \begin{bmatrix} | & & | \\ [u^k]_1 & \cdots & [u^k]_N \\ | & & | \end{bmatrix} \in \mathbb{R}^{K\times N} \text{ e } V^k = \begin{bmatrix} | & & | \\ [v^k]_1 & \cdots & [v^k]_N \\ | & & | \end{bmatrix} \in \mathbb{R}^{K\times N}.$$

Calcule $X_c^k = S^{-1/2} U^k {V^k}^T \in \mathbb{R}^{K\times N}$ *e seja* $x_c^k \in \mathbb{R}^{NK}$ *a matriz* X_c^k *vista como um vetor de* $\mathbb{R}^{NK}$*.*

Passo 5. *Defina*

$$H_k = \sigma^k A \in \mathbb{R}^{NK\times NK}$$

e volte para o Passo 2.

Portanto, como este algoritmo é um caso particular do Algoritmo 6.1, segue como conseqüência imediata que está bem definido, ou seja, após um número finito de aumentos de ρ, é encontrado X_c^k de modo que $ared \geq \beta_1 pred$. Também, todo ponto de acumulação é um ponto estacionário do problema

$$\begin{aligned} \min \quad & E(X) \\ s.a. \quad & X \in \Omega, \end{aligned}$$

ou seja, é um **ponto fixo Fock**.

Como no Algoritmo 6.1, a escolha de x^{k+1} no Passo 2 do algoritmo acima poderia ser x_c^k; no entanto, escolher x^{k+1} de modo que $f(x^{k+1}) \leq f(x_c^k)$ prepara o Algoritmo TR para incorporar esquemas de aceleração em implementações práticas. Um destes esquemas considerados neste trabalho, talvez o mais natural para este algoritmo, é a bem conhecida extrapolação DIIS [48, 49], colocada com detalhes no Capítulo 5. O Algoritmo TR incorporando a aceleração DIIS será chamado de TR+DIIS.

A diferença essencial entre o TR e o TR+DIIS é que, neste último, em vez de usar somente iterações SCF no Passo 1, usa-se o esquema DIIS como colocado no Algoritmo 5.4. Portanto, se o funcional energia (E), quando avaliado no ponto candidato fornecido pelo

esquema DIIS (x^k_{DIIS}), for menor do que quando avaliado em x^k_c e, além disso, satisfizer a condição de decréscimo suficiente,

$$ared \geq \beta_1 pred,$$

o novo iterado x^{k+1} é escolhido para ser x^k_{DIIS}, caso contrário o algoritmo segue normalmente como no TR. Com isso o método TR+DIIS apresenta as mesmas propriedades teóricas do TR.

Vale a pena observar que, em ambos os algoritmos, pontos estacionários não necessariamente são pontos "Aufbau", embora a tendência é que os métodos convirjam a pontos com essa característica, já que o primeiro passo em cada iteração é o passo dado pelo clássico ponto fixo SCF.

Alguns experimentos computacionais são colocados a seguir.

6.5 Resultados numéricos

Para mostrar a eficiência dos algoritmos introduzidos neste capítulo, a saber, TR e TR+DIIS, são colocados nesta seção alguns resultados numéricos. Ambos os métodos serão confrontados com o algoritmo de ponto fixo SCF (FP) e o DIIS (respectivamente, Algoritmos 5.1 e 5.4) em moléculas que apresentam significativas dificuldades de resolução.

Os resultados colocados nesta seção foram obtidos em Fortran, em um computador Pentium 1.5 Ghz, com 256 Mbytes de memória RAM, sendo que a maioria das rotinas envolvidas foram fornecidas por alguns bolsistas do programa de pós-graduação do Instituto de Química da Unicamp.

Para a implementação do Algoritmo TR, no Passo 2 foi escolhido $x^{k+1} = x^k_c$. Na versão TR+DIIS, foi aproveitada a vantagem de liberdade da escolha de x^{k+1}, que foi escolhido como um passo acelerado do método DIIS (x^k_{DIIS}), usando os iterados anteriores para melhorar x^k_c, sempre que este apresentar o decréscimo suficiente na função energia.

No TR+DIIS a aceleração é usada a partir da segunda iteração. Portanto, a extrapolação usa dois resíduos. Nas iterações subseqüentes, o número de resíduos usados na interpolação é aumentado até alcançar o máximo de dez. Daí em diante, os dez mais recentes resíduos, gerados pelas dez mais recentes iterações bem-sucedidas, são usados. Os resíduos correspondentes a pontos onde a energia aumenta são descartados por motivos de extrapolação.

Para os experimentos,

- foram tomados $\beta_1 = 10^{-4}$, $\sigma_{min} = 10^{-2}$ e $\sigma_{max} = 10^2$;

- foi assumida a convergência para qualquer método testado quando

$$\frac{|E(x^k) - E(x^{k+1})|}{|E(x^k)|} < 10^{-9};$$

- o número máximo de iterações permitidas foi de 5001; e
- foram usados diferentes tipos de pontos iniciais: os autovetores da matriz H definida em (4.5); o ponto inicial de Huckel fornecido pelo pacote GAMESS [57], o qual é destinado para o mesmo problema; e, para alguns casos, foi tomada a aproximação inicial induzida pela matriz identidade. A escolha destas aproximações se deve ao fato de elas poderem ser facilmente obtidas.

Para os testes, foram usadas moléculas com geometrias conforme a Tabela 6.1. As moléculas CrC e Cr_2 são conhecidas por apresentarem propriedades de convergência instáveis [9, 10]. Duas geometrias para a molécula de CO foram escolhidas como exemplos, já que geometrias distorcidas causam dificuldades de convergência [26]. Finalmente, moléculas de água (H_2O) e amônia (NH_3) foram usadas nos exemplos para ilustrar como os algoritmos TR e TR+DIIS se comportam em situações onde os algoritmos clássicos são bem-sucedidos.

Tabela 6.1: Parâmetros geométricos das moléculas usadas nos experimentos

	Geometria		
Molécula	"Bond length"/ Å	Ângulo	Dihedral
CrC	2.00		
Cr_2	2.00		
CO	1.40		
CO(Dist)	2.80		
H_2O	0.95(OH)	109°(HOH)	
NH_3	1.008(NH)	109°(HNH)	120°(HNHH)

A Tabela 6.2 mostra o número de iterações necessárias pelos métodos FP, DIIS, TR e TR+DIIS para atingir a convergência para dois tipos de bases, a saber, STO-3G e 6-31G. Para as moléculas de água e amônia, os experimentos mostraram que o número de iterações tomadas pelos métodos FP e TR, por um lado, e DIIS e TR+DIIS, por outro, é o mesmo. Isso se deve ao fato de que o ponto fixo SCF e o algoritmo DIIS são sempre bem-sucedidos em fornecer um novo ponto candidato com uma redução suficiente na função energia. Neste caso, nunca é necessário aumentar o parâmetro de regularização e, portanto, o TR e o TR+DIIS se reduzem, respectivamente, ao métodos FP e DIIS.

Para a molécula de CO, usando a base STO-3G, a convergência do clássico método FP sempre falha. A energia oscila até atingir o número máximo de iterações (5001). Para este exemplo, o DIIS é bastante eficiente, convergindo para qualquer ponto inicial em, no máximo, 11 iterações. O método TR sempre converge em todos os casos, como era esperado, porém quase dobrou o número de iterações em relação ao DIIS e converge para uma solução que viola o princípio de "Aufbau" quando a aproximação inicial foi derivada da matriz identidade. Finalmente, o TR+DIIS apresenta convergência rápida, com um pouco menos iterações do que o DIIS, para um ponto que satisfaz o princípio de "Aufbau". Na molécula CO com geometria distorcida, a eficiência e a confiabilidade dos algoritmos TR e TR+DIIS se tornam aparentes: o método FP falha em todos os casos. O DIIS converge em 117 iterações para um ponto com energia maior do que a solução encontrada em 12 e 10 iterações, respectivamente, pelos métodos TR e TR+DIIS quando a base STO-3G é usada, tomando a aproximação inicial dada pela matriz H. Usando a aproximação de Huckel, o DIIS converge para uma solução com energia menor, entretanto são necessárias 85 iterações contra 13 e 15 iterações, respectivamente, dos métodos TR e TR+DIIS. Finalmente, quando é usada a base 6-31G, o DIIS converge mais rápido, mas para um ponto com energia maior do que as obtidas pelos algoritmos TR e TR+DIIS. Foi observado que o TR convergiu em 384 iterações a partir da aproximação inicial Huckel porque seu primeiro ponto candidato é dado pelo algoritmo de ponto fixo, o qual falha sistematicamente para este problema.

Para a molécula Cr_2, o método DIIS teve um melhor desempenho do que os métodos propostos por este trabalho. Foi obtida convergência para todos os métodos, porém o método FP convergiu a um ponto com 9.3 u.a. maior do que a solução encontrada pelo método DIIS. A diferença em energia para as outras soluções é na ordem de 5×10^{-6} $u.a.$ Nestes casos, o DIIS convergiu em, no máximo, 37 iterações, sendo 384 e 134 iterações necessárias para alcançar a convergência para os métodos TR e TR+DIIS, respectivamente, a partir da aproximação Huckel.

Finalmente, um experimento bastante interessante foi fornecido pela molécula CrC. Para a base 6-31G, todos, exceto o FP, convergiram. No DIIS foi necessário menos iterações quando foi tomada a aproximação inicial fornecida pela matriz H, porém mais iterações do que o TR+DIIS, quando começando da aproximação inicial Huckel. O método TR realizou mais iterações do que ambos os métodos em todos os casos e convergiu para uma solução com energia levemente maior.

Quando é usada a base STO-3G, os testes foram mais interessantes, como é ilustrado na Figura 6.1. Começando com a aproximação inicial dada pela matriz H, os métodos FP e DIIS não conseguem convergência em 5001 iterações, como pode ser visto na Figura 6.1(a). O método TR convergiu em 71 iterações para uma solução com energia maior, que

não satisfaz o princípio de "Aufbau", enquanto o TR+DIIS convergiu em 29 iterações para uma solução "Aufbau" com menor energia. A partir da aproximação inicial Huckel, o DIIS convergiu, porém com mais iterações do que o TR+DIIS, e o TR convergiu em um número significativamente maior, como mostrado na Figura 6.1(b). Finalmente, da aproximação fornecida pela identidade, o DIIS oscilou no começo e parou de oscilar provavelmente graças a erros de arredondamentos numéricos, convergindo finalmente em 180 iterações, como pode ser visto na Figura 6.1(c). O método TR converge em 40 iterações para uma solução com energia maior, e o TR+DIIS converge para um ponto com a menor energia em 36 iterações. Este é um interessante exemplo em que o método DIIS falha para um ponto inicial, enquanto o TR é bem-sucedido.

A Tabela 6.2 mostra que o método FP falha na convergência em 12 dos 19 testes realizados, tendo o método TR convergido em todos os casos, apesar do fato de o primeiro ponto candidato ser calculado como no método de ponto fixo (FP).

Tabela 6.2: Número de iterações desempenhadas para cada algoritmo em cada problema. FP: Algoritmo ponto fixo SCF; DIIS: A extrapolação DIIS; TR: A versão simples do algoritmo proposto neste trabalho; TR+DIIS: A versão acelerada do algoritmo TR

			Algoritmo			
Molécula	Base	Ponto inicial	FP	DIIS	TR	TR+DIIS
H_2O	STO-3G	H^{core}	7	5	7	5
	6-31G	H^{core}	18	8	18	8
NH_3	STO-3G	H^{core}	8	7	8	7
	6-31G	H^{core}	14	7	14	7
CO	STO-3G	H^{core}	X[a]	11	22	10
		Huckel	X[a]	7	16	7
		Identity	X[a]	11	17[b,c]	9
CO(Dist)	STO-3G	H^{core}	X[a]	117[c]	12	10
		Huckel	X[a]	85	13	15
	6-31G	H^{core}	X[a]	27[c]	158	115
		Huckel	X[a]	36[c]	384	59
Cr_2	STO-3G	H^{core}	52[c]	13	56	38
		Huckel	12[c]	33[c]	398	134
		Identity	7[c]	37	50[c]	26[c]
CrC	STO-3G	H^{core}	X[a]	X[a]	71[b,c]	29
		Huckel	X[a]	49	129	23
		Identity	X[a]	180	40[c]	36
	6-31G	H^{core}	X[a]	19	102[c]	29[c]
		Huckel	X[a]	52[c]	113[c]	37

[a]Não convergiu em 5001 iterações.

[b]Convergiu a um ponto que viola o princípio de "Aufbau".

[c]Convergiu para uma energia maior do que algum dos outros algoritmos.

Vale a pena observar que, para cada iteração dos métodos introduzidos neste trabalho, mais do que uma avaliação é necessária quando for preciso aumentar o parâmetro de regularização. Por essa razão, em casos críticos, um pequeno número de iterações dos métodos TR e TR+DIIS não necessariamente reflete um pequeno tempo computacional. Entretanto, como eficientes procedimentos de escalamentos lineares estão sendo desenvolvidos para construir matrizes de Fock, métodos confiáveis tornam-se mais e mais importantes. Com os resultados obtidos, pode-se observar que os métodos introduzidos por este trabalho são bastantes confiáveis, podendo, então, ser usados como uma alternativa automática para fornecer convergência para problemas difíceis de resolver quando divergência ou comportamentos oscilatórios são detectados em outros algoritmos.

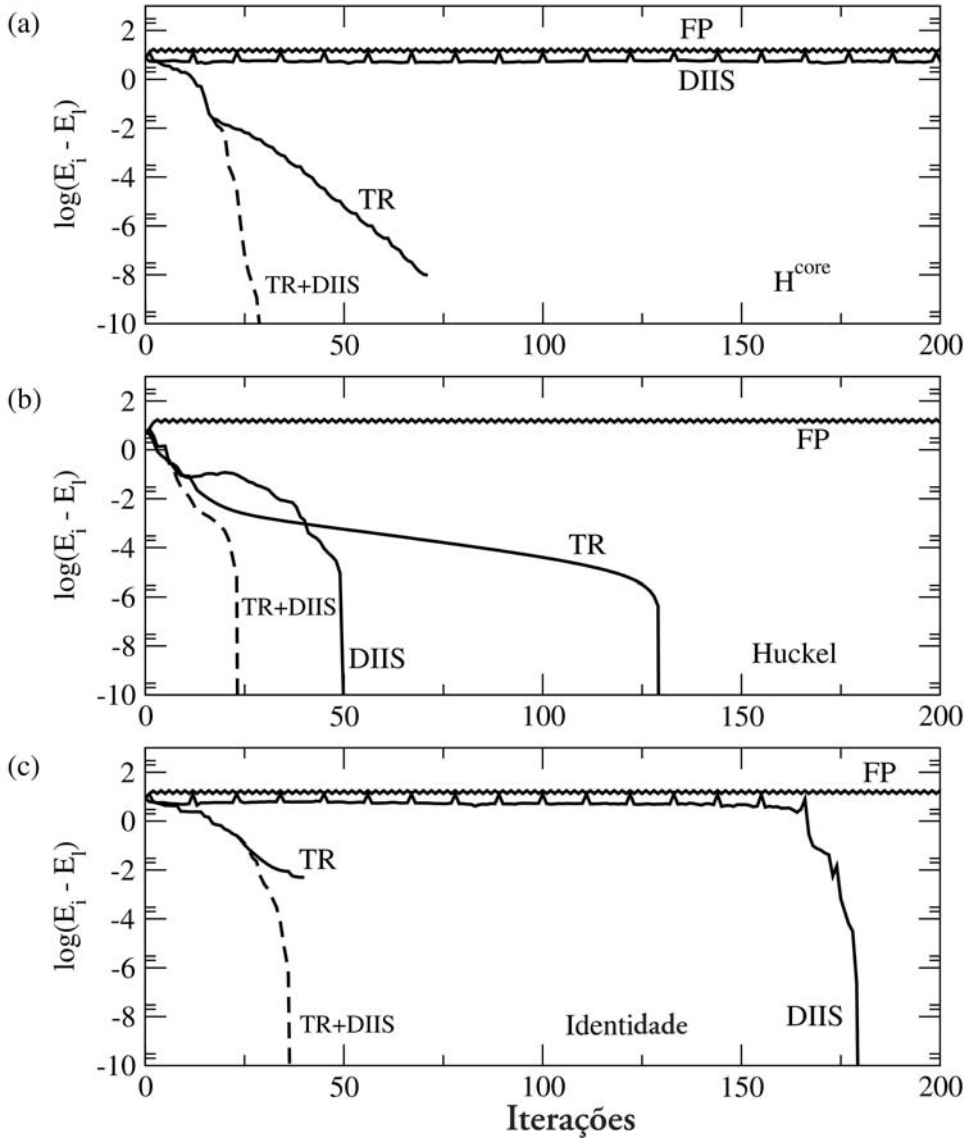

Figura 6.1: Comportamento da convergência dos quatros métodos para a molécula CrC usando a base STO-3G

Conclusões

Na primeira parte deste trabalho foi introduzido um novo algoritmo para resolver sistemas não-lineares de equações indeterminados sujeitos à restrição de caixa. O principal objetivo é a fase de viabilidade dos algoritmos de programação não-linear baseados em restauração periódica. Neste caso, muitas soluções são pontos interiores. Com isso em mente, este método pode ser interpretado como uma globalização do método de norma mínima de Newton para resolver sistemas indeterminados. A estratégia de globalização é essencialmente a mesma introduzida por Coleman e Li [15] e adaptada por Bellavia *et al.* [6] para sistemas quadrados de equações não-lineares.

Como foi visto, um número limitado de experimentos numéricos mostraram que o algoritmo se comporta como na teoria. Em muitas iterações ele se reduz ao método de norma mínima de Newton, e, quando isso não acontece, a estratégia de convergência global conduz os iterados para uma solução.

Foi observada convergência quadrática na vizinhança de soluções interiores, exceto no Problema 12, onde o Jacobiano é bastante mal condicionado.

O Algoritmo KYF [32] (Algoritmo 2.2), o qual foi usado para comparar os resultados numéricos, pode também ser considerado como uma globalização do método de norma mínima de Newton, pelo menos quando o parâmetro de regularização é bastante pequeno, o que corresponde a situações em que os melhores resultados foram observados. Em geral, a estratégia de globalização do KYF, baseada em gradiente projetado, não é tão eficiente quanto a estratégia de globalização do algoritmo proposto (Algoritmo 2.1).

Como muitos problemas de programação não-linear apresentam significativo número de restrições e de variáveis, trabalhos futuros poderão considerar a extensão desse tipo de algoritmos a fim de considerar a esparsidade do Jacobiano e usar procedimentos iterativos lineares para encontrar a direção newtoniana. Além disso, as dificuldades de avaliar o Jacobiano deixam uma oportunidade de estender os métodos quasi-Newton para sistemas indeterminados [36, 62] para os casos com variáveis canalizadas.

Na segunda parte deste trabalho foi introduzido um novo algoritmo para resolver problemas de programação não-linear, sendo o conjunto viável um conjunto fechado arbitrário. Este algoritmo foi baseado em técnicas de regiões de confiança [40] e nas idéias do método

de Levenberg-Marquardt [45], sendo no algoritmo proposto os subproblemas mais simples do que os considerados em [40], fato que pode ser observado na aplicação do método para um determinado tipo de problema, a saber, o cálculo eletrônico, em que foi comprovada a eficiência do método por meio de experimentos numéricos.

Sob fracas hipóteses, foi provada convergência global na seqüência de iterados, mostrando que qualquer ponto de acumulação da seqüência gerada pelo método é um ponto estacionário do problema de otimização.

Uma importante aplicação do método está no cálculo de estruturas eletrônicas, resultando em um algoritmo globalmente convergente, o qual foi chamado neste trabalho de TR. Neste caso, os subproblemas envolvidos são resolvidos por uma simples decomposição espectral, o que torna esse algoritmo confiável e viável do ponto de vista numérico para esse tipo de problema. Os experimentos numéricos mostraram que o algoritmo TR é robusto e se comporta como esperado. A convergência é obtida em todos os exemplos, apesar do fato de ser tomada uma simples iteração do método de ponto fixo como primeiro passo do método. Este novo algoritmo deve ser bastante útil quando falhas de outros métodos são detectadas, fornecendo, então, confiabilidade para rotinas usadas no cálculo eletrônico. Qualquer heurística ou método conhecido para ser eficiente pode ser usado na fase de aceleração da estrutura algorítmica do método TR sem afetar as propriedades de convergência global, já que o algoritmo foi preparado para incorporar em sua estrutura estratégias de aceleração. Nesse sentido, o algoritmo TR deve ser visto não como um competidor dos outros algoritmos, mas sim como um procedimento que garanta que a convergência seja obtida. Nos resultados numéricos foi incorporada a bem conhecida aceleração DIIS no algoritmo TR, resultando em um procedimento chamado aqui por TR+DIIS, para calcular estruturas eletrônicas. Neste caso, os experimentos mostraram que essa abordagem é bastante promissora, obtendo um desempenho superior ao dos outros métodos testados para a maioria dos problemas. A hibridização com outros métodos deve ser também eficiente, abrindo, portanto, um amplo e importante caminho para pesquisas futuras.

Uma importante característica do método é que sua implementação não depende da forma da função objetivo, mas sim da natureza das restrições. Portanto, outras aplicações para qualquer problema que envolva restrições de ortonormalidade certamente fornecerão resultados também confiáveis.

Referências Bibliográficas

[1] J. ABADIE and J. CARPENTIER. Generalization of the Wolfe reduced-gradient method to the case of nonlinerar constraints. In *in Optimization*, pages 37–47, New York, 1968. Academic Press.

[2] J. E. ALMLÖF and T. H. FISHER. General methods for geometry and wave function optimization. *Journal of Physical Chemistry*, 96:9768–9774, 1992.

[3] G. B. BACSKAY. A quadratically convergent Hartree-Fock (QC-SCF) method. Application to closed shell systems. *Chemical Physics*, 61:385–404, 1981.

[4] G. B. BACSKAY. A quadratically convergent Hartree-Fock (QC-SCF) method. application to open-shell orbital optimization and coupled pertubed Hartree-Fock calculations. *Chemical Physics*, 65:383–396, 1982.

[5] V. K. BASAPUR, A. MIELE, and E. M. SIMS. Sequential gradient-restoration algorithm for mathematical programming problems with inequality constraints, Part 1, Theory. Aero-Astronautics Report 168, Rice University, 1983.

[6] S. BELLAVIA, M. MACCONI, and B. MORINI. An affine scaling trust-region approach to bound-constrained nonlinear systems. *Applied Numerical Mathematics*, 44:257–280, 2003.

[7] S. BELLAVIA, M. MACCONI, and B. MORINI. STRSCNE: A scaled trust-region solver for constrained nonlinear equations. *Computational Optimization and Applications*, 28:31–50, 2004.

[8] E. G. BIRGIN, J. M. MARTÍNEZ, and M. RAYDAN. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, 10:1196–1211, 2000.

[9] E. CANCÈS, K. K. KUDIN, and G. E. SCUSERIA. A black-box self-consistent field convergence algorithm: One step closer. *Journal of Chemical Physics*, 116(19):8255–8261, 2002.

[10] E. CANCÈS and C. LE BRIS. Can we outperform the DIIS approach for eletronic structure calculations? *International Journal of Quantum Chemistry*, 79:82–90, 2000.

[11] E. CANCÈS and C. LE BRIS. On the convergence of SCF algorithms for the Hartree-Fock equations. *Mathematical Modeling and Numerical Analysis*, 34(4):749–774, 2000.

[12] M. R. CELIS, J. E. DENNIS, and R. A. TAPIA. *A trust region strategy for nonlinear equality constrained optimization*, pages 71–82. in: P. T. Boggs, R. Byrd and R. Schnabel, eds. Numerical Optimization. SIAM, Philadelphia, 1984.

[13] G. CHABAN, M. S. GORDON, and M. W. SCHIMDT. Approximate second order method for orbital optimization of SCF and MCSCF wavefunctions. *Theoretical Chemistry Accounts*, 97:88–95, 1993.

[14] T. F. COLEMAN and Y. LI. On the convergence of interior-reflective newton methods for nonlinear minimization subject to bounds. *Mathematical Programming*, 67:189–224, 1994.

[15] T. F. COLEMAN and Y. LI. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal of Optimization*, 6:418–445, 1996.

[16] L. L. COMBS and C. A. WAGGONER. Optimization techniques in energy calculations involving the Hartree-Fock density matrix. *Journal of Optimization Theory and Applications*, 76:225–240, 1993.

[17] E. E. CRAGG, A. V. LEVY, and A. MIELE. Modifications and extensions of the conjugate-gradient restoration algorithm for mathematical programming problems. *Journal of Optimization Theory and Applications*, 7:450–472, 1971.

[18] N. DAN, M. FUKUSHIMA, and N. YAMASHITA. Convergence properties of inexact Levenberg-Marquardt method under local error bound conditions. *Optimization Methods and Software*, 17:605–626, 2002.

[19] M. DEFRANCESCHI and C. Le BRIS. *Mathematical models and methods for ab initio quantum chemistry*, volume 74 of *Lectures Notes in Chemistry*, chapter 2. Springer, Berlin, 2000.

[20] J. E. DENNIS. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, Philadelphia, 1996.

[21] J. DOUADY, Y. ELLINGER, R. SUBRA, and B. LEVY. Exponential transformation of molecular orbitals: A quadratically convergent SCF procedure. I. General formulation and application to closed-shell ground states. *Journal of Chemical Physics*, 72:1452–1462, 1980.

[22] T. H. DUNNING. Gaussian-basis sets for use in correlated molecular calculation. *Journal of Chemical Physics*, 90(2):1007–1023, 1989.

[23] J. B. FRANCISCO, J. M. MARTÍNEZ, and L. MARTÍNEZ. Globally convergent trust-region methods for self-consistent field electronic structure calculations. *Journal of Chemical Physics*, 121:10863–10878, 2004.

[24] D. M. GAY. Computing optimal locally constrained step. *SIAM Journal on Scientific Computing*, 2(2):186–197, 1981.

[25] G. A. GOLUB and C. F. VAN LOAN. *Matrix Computations*. The John Hopkins University Press Ltda, London, 3 edition, 1996.

[26] T. HELGAKER, J. JØRGENSEN, and J. OLSEN. *Molecular electronic - Structure theory*. Wiley, Chichester, 2000.

[27] I. H. HILLIER and V. R. SAUNDERS. A level-shift method for converging closed-shell Hartree-Fock wave functions. *International Journal of Quantum Chemistry*, 7:699–705, 1973.

[28] W. HOCK and J. SCHITTKOWSKI. *Test Examples for Nonlinear Programming Codes*. Series Lectures Notes in Economics Mathematical Systems. Springer Verlag, 1981.

[29] S. HUZINAGA. Gaussian-type functions for polyatomic systems. *Journal of Chemical Physics*, 42(4):1293–1302, 1965.

[30] S. HUZINAGA and Y. SAKAI. Gaussian-type functions for polyatomic systems. *Journal of Chemical Physics*, 50(3):1371–1381, 1969.

[31] J. I'HAYA and T. SANO. Constrained newton approach adequate to direct self-consistent field calculation in closed-and open-shell configuration. *Journal of Chemical Physics*, 95:6607–6614, 1991.

[32] C. KANZOW, N. YAMASHITA, and M. FUKUSHIMA. Levenberg-Marquardt methods for constrained nonlinear equations with strong local convergence properties. Technical Report (to appear in *Journal of Computational and Applied Mathematics*)

2002-007, Department of Applied Mathematics and Physics, Kyoto University, April 2002.

[33] D. N. KOZAKEVICH, J. M. MARTÍNEZ, and S. A. SANTOS. Solving nonlinear systems of equations with simple bounds. *Computational and Applied Mathematics*, 16:215–235, 1997.

[34] E. KREYSZIG. *Introductory functional analysis with applications*. Wiley classics library. J. Wiley, 1989.

[35] P. L. LIONS. Solutions of Hartree-Fock equations for coulomb systems. *Communications in Mathematical Physics*, 109:33–97, 1987.

[36] J. M. MARTÍNEZ. Quasi-newton methods for solving underdetermined nonlinear simultaneous equations. *Journal of Computational and Applied Mathematics*, 34:171–190, 1990.

[37] J. M. MARTÍNEZ. Quasi-inexact-newton methods with global convergence for solving constrained nonlinear systems. *Nonlinear Analysis*, 30(1):1–7, 1997.

[38] J. M. MARTÍNEZ. Inexact restoration method with lagrangian tangent decrease and new merit function for nonlinear programming. *Journal of Optimization Theory and Applications*, 111:39–58, 2001.

[39] J. M. MARTÍNEZ and E. A. PILOTTA. Inexact restoration algorithms for constrained optimization. *Journal of Optimization Theory and Applications*, 104:135–163, 2000.

[40] J. M. MARTÍNEZ and S. A. SANTOS. A trust region strategy for minimization on arbitrary domains. *Mathematical Programming*, 68:267–302, 1995.

[41] J. M. MARTÍNEZ and S. A. SANTOS. Convergence results on an algorithm for norm constrained regularization and related problems. *RAIRO Operations Research*, 31:269–294, 1997.

[42] R. McWEENY and B. T. SUTCLIFFE. *Methods of Molecular Quantum Mechanics*. Academic Press, New York, 1969.

[43] J. M. MORE and D. C. SORENSEN. Computing a trust region step. *SIAM Journal on Scientific Computing*, 4(3):553–572, 1983.

[44] A. W. NAYLOR and G. R. SELL. *Linear Operator Theory in Engineering and Science*. Springer Series in Applied Mathematical Science. Springer Verlag, New York, 1982.

[45] J. NOCEDAL and S. J. WRIGHT. *Numerical Optimization*. Springer Series in Operations Research. Springer Verlag, New York, 1999.

[46] E. PÄRT-ENANDER and A. SJÖBERG. *The Matlab Handbook 5*. Addison Wesley, Harlow, UK, 1999.

[47] M. J. D. POWELL and Y. YUAN. A trust region algorithm for equality constrained optimization. *Mathematical Programming*, 49:189–211, 1990.

[48] P. PULAY. Convergence acceleration iterative sequences. The case of SCF iteration. *Chemical Physics Letters*, 73(2):393–398, 1980.

[49] P. PULAY. Improved SCF convergence acceleration. *Journal of Computational Chemistry*, 3(4):556–560, 1982.

[50] M. RAYDAN. The Barzilai and Borwein gradient method for large scale unconstrained minimization problem. *SIAM Journal on Optimization*, 7:26–33, 1997.

[51] A. P. RENDELL. Diagonalization-free SCF. *Chemical Physics Letters*, 229:204–210, 1994.

[52] L. THØRGENSEN, J. OLSEN, D. YEAGER, P. JØRGENSEN, P. SAŁEK, and T. HELGAKER. A trust-region self-consistent field method: Towards a black-box optimization in Hartree-Fock and Kohn-Sham theories. *Journal of Chemical Physics*, 121(1):16–27, 2004.

[53] C. C. J. ROOTHAN. New developments in molecular orbital theory. *Review of Modern Physics*, 23(2):69–89, 1951.

[54] J. B. ROSEN. The gradient projection method for nonlinear programming, Part 2, nonlinear constraints. *SIAM Journal on Applied Mathematics*, 9:514–532, 1961.

[55] W. RUDIN. *Principles of Mathematical Analysis*. McGraw-Hill, New York, 3 edition, 1976.

[56] A. H. SAMEH and J. A. WISNIEWSKI. A trace minimization algorithm for the generalized eigenvalue problem. *SIAM Journal on Numerical Analysis*, 19:1243–1259, 1982.

[57] M. W. SCHIMDT and et al. General atomic and molecular electronic structure system. *Journal of Computational Chemistry*, 14:1347–1363, 1993.

[58] J. SIMONS. An experimental chemist's guide to ab-initio quantum chemistry. *Journal of Physical Chemistry*, 95:1017–1029, 1991.

[59] D. C. SORENSEN. Newton's method with a model trust region modification. *SIAM Journal on Numerical Analysis*, 19(2):409–426, 1982.

[60] A. SZABO and S. N. OSTLUND. *Modern Quantum Chemistry: Introduction to Advanced Eletronic Structure Theory*. Dover Pub., New York, 1989.

[61] A. VARDI. A trust region algorithm for equality constrained minimization: Convergence properties and implementation. *SIAM Journal on Control and Optimization*, 28:34–49, 1985.

[62] H. F. WALKER and L. T. WATSON. Least-change secant update methods for underdetermined systems. *SIAM Journal on Numerical Analysis*, 27(5):1227–1262, 1990.

Apêndice A

O Cálculo de Autovalores como um Problema de Programação Não-Linear

Neste apêndice é mostrado que os autovalores de uma certa matriz simétrica A formam a solução global de um problema de programação não-linear, introduzido mais à frente. Esse resultado, embora seja um fato conhecido de álgebra linear, merece atenção especial, já que alguns resultados importantes deste trabalho não seriam possíveis caso sua veracidade não fosse confirmada.

Dada uma matriz simétrica $A \in \mathbb{R}^{K\times N}$, defina $\upsilon : \mathbb{R}^{K\times N} \to \mathbb{R}$ $(K \leq N)$ por

$$\upsilon(X) = \frac{1}{2}\sum_{i=1}^{N} X_i^T A X_i, \tag{A.1}$$

onde

$$X = \begin{bmatrix} | & & | \\ X_1 & \cdots & X_N \\ | & & | \end{bmatrix}.$$

A proposição a seguir mostra que υ é invariante por transformações unitárias pela direita. Para tanto, dado $X \in \mathbb{R}^{K\times N}$, considere, como sempre,

$$J(X) = \{Y \in \mathbb{R}^{K\times N} \mid Y = XU \text{ para alguma matriz unitária } U \in \mathbb{R}^{N\times N}\}.$$

Proposição A.1. *Considere a aplicação υ como em (A.1) e tome $Y \in J(X)$ para algum $X \in \mathbb{R}^{K\times N}$. Então, $\upsilon(X) = \upsilon(Y)$.*

Prova. Por hipótese, $Y_i = \sum_{j=1}^{N} X_j U_{ji}$ para alguma matriz unitária U. Assim,

$$\begin{aligned} \upsilon(Y) &= \frac{1}{2}\sum_{i=1}^{N} Y_i^T A Y_i = 2\sum_{i=1}^{N}(\sum_{j=1}^{N} X_j^T U_{ji}) A (\sum_{a=1}^{N} X_a U_{ai}) \\ &= \frac{1}{2}\sum_{j,a=1}^{N} X_j^T A X_a \sum_{i=1}^{N} U_{ji} U_{ai}. \end{aligned}$$

Como U é uma matriz unitária, segue que $\sum_{i=1}^{N} U_{ji} U_{ai} = \delta_{ja}$ e, portanto,

$$\upsilon(Y) = \frac{1}{2}\sum_{j=1}^{N} X_j^T A X_j = \upsilon(X),$$

e a proposição está provada. ■

Considere agora o problema de programação não-linear

$$\begin{array}{ll} \min & \upsilon(X) \\ s.a. & X \in \Omega, \end{array} \tag{A.2}$$

onde

$$\Omega = \{X \in \mathbb{R}^{K\times N} \mid X_i^T S X_j = \delta_{ij},\ i,j = 1,\ldots,N\} \tag{A.3}$$

para alguma matriz $S \in \mathbb{R}^{K\times K}$ simétrica definida positiva.

Como A é simétrica,

$$\frac{\partial \upsilon(X)}{\partial X_i} = A X_i \tag{A.4}$$

e, portanto, as condições KKTs do PNL (A.2) são

$$\left\{ \begin{array}{lll} A X_i - \sum_{j=1}^{N} \theta_{ji} S X_j &=& 0 \qquad i = 1,\ldots,N \\ X \in \Omega, & & \end{array} \right. \tag{A.5}$$

onde θ_{ij} são os multiplicadores de Lagrange. Considerando $\Theta \in \mathbb{R}^{N\times N}$ a matriz formada pelos multiplicadores θ_{ij}, o sistema KKT anterior pode ser escrito na forma matricial

$$\left\{ \begin{array}{lll} AX - SX\Theta &=& 0 \\ X \in \Omega. & & \end{array} \right. \tag{A.6}$$

O teorema a seguir mostra que encontrar o minimizador global de (A.2) é equivalente a encontrar os N autovetores associados aos N menores autovalores generalizados da matriz A, ou seja, resolver o problema (A.2) consiste em encontrar os N autovetores associados

aos N menores autovalores do problema

$$AC = SC\mu, \tag{A.7}$$

onde C é a matriz formada pelos autovetores na coluna, μ a matriz diagonal formada pelos autovalores e S é a matriz definida em (A.3).

Teorema A.2. *Seja $C \in \mathbb{R}^{K\times N}$ a matriz cujas colunas são os N autovetores associados aos N menores autovalores do problema de autovalores generalizado (A.7). Então, C é um minimizador global do problema de programação não-linear (A.2).*

Prova. Por hipótese, como A é simétrica e S é positiva definida, tem-se que as colunas de C são S-ortonormais e, então,

$$\begin{cases} AC = SC\mu \\ C \in \Omega, \end{cases}$$

onde $\mu \in \mathbb{R}^{K\times N}$ é a matriz diagonal formada pelos autovalores μ_i. Pela Equação (A.6), segue que C satisfaz as condições KKT de (A.2). Note que os multiplicadores de Lagrange associados às restrições de ortogonalidade são nulos.

Suponha que C não é minimizador global, então existe $V \in \mathbb{R}^{K\times N}$ satisfazendo (A.6) tal que $g(V) < g(C)$. Por hipótese tem-se que existe uma matriz $\Theta \in \mathbb{R}^{N\times N}$, formada pelos multiplicadores de Lagrange $\theta_{ij} \in \mathbb{R}$, tal que

$$\begin{cases} AV = SV\Theta \\ V \in \Omega. \end{cases} \tag{A.8}$$

Note que a matriz $\Theta \in \mathbb{R}^{N\times N}$ é simétrica e, então, diagonalizável. Assim, existe uma matriz unitária $U \in \mathbb{R}^{N\times N}$ e uma matriz diagonal $\Lambda \in \mathbb{R}^{N\times N}$ tal que

$$U^T\Theta U = \Lambda.$$

Considere $V' = VU$. Então, pela Proposição A.1, $g(V') = g(V)$, e, como Ω é invariante por matrizes unitárias (ver Lema 4.9 no Capítulo 4), segue que $V' \in \Omega$. Multiplicando a Equação (A.8) por U e usando o fato que $UU^T = I$, segue que

$$\begin{aligned} AVU &= SVUU^T\Theta U \\ &= SV'\Lambda. \end{aligned}$$

Assim,

$$\begin{cases} AV' = SV'\Lambda \\ V' \in \Omega, \end{cases}$$

ou seja, V' é um ponto estacionário do PNL (A.2) com os multiplicadores associados às restrições de ortogonalidade nulos, já que Λ é uma matriz diagonal. Note que as colunas da matriz V' são N autovetores generalizados de A e que, para todo $i = 1, \ldots, N$, têm-se $C_i^T A C_i = \mu_i$ e ${V'_i}^T A V'_i = \lambda_i$, onde μ_i e λ_i são autovalores generalizados de A, sendo os $\{\mu_i\}_{i=1}^N$ os menores. Portanto,

$$g(C) = \frac{1}{2}\sum_{i=1}^{N} \mu_i \leq \frac{1}{2}\sum_{i=1}^{N} \lambda_i = g(V') = g(V),$$

o que é uma contradição. Logo, C é um minimizador global. ■