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

Análise Computacional do Método de Restauração Inexata para Problemas de Otimização com Restrições de Igualdade e de Canalização

Diego Derivaldo dos Reis
Mestrado em Matemática Aplicada - Campinas - SP

Orientadora: Prfa. Dra. Marcia Aparecida Gomes Ruggiero

Análise Computacional do Método de Restauração Inexata para Problemas de Otimização com Restrições de Igualdade e de Canalização

Este exemplar corresponde à redação final da dissertação devidamente corrigida e defendida por **Diego Derivaldo dos Reis** e aprovada pela comissão julgadora.

Campinas, 14 de Maio de 2010

Prof. Dra. Márcia Aparecida Gomes Ruggiero
Orientadora

Banca Examinadora:

Prof. Dra. Márcia Aparecida Gomes Ruggiero
Prof. Dra. Sandra Augusta Santos
Prof. Dra. Luziane Ferreira de Mendonça

Dissertação apresentada ao Instituto de Matemática, Estatística e Computação Científica, UNICAMP, como requisito parcial para obtenção do Título de Mestre em Matemática Aplicada

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP
Bibliotecária: Maria Fabiana Bezerra Müller – CRB8 / 6162

Reis, Diego Derivaldo dos
R277a Análise computacional do método de restauração inexata para problemas de otimização com restrições de igualdade e de canalização / Diego Derivaldo dos Reis-- Campinas, [S.P. : s.n.], 2010.

Orientador : Márcia Aparecida Gomes Ruggiero
Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1.Programação não-linear. 2.Otimização matemática. 3. Sistemas não lineares. 4. Restauração inexata. I. Ruggiero, Márcia Aparecida Gomes. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.

Título em inglês: Computational analysis of inexat restoration methods for optimization with equality constraints and box.

Palavras-chave em inglês (Keywords): 1. Nonlinear programming. 2. Mathematical optimization. 3. Nonlinear systems. 4. Inexact restoration.

Área de concentração: Otimização

Titulação: Mestre em Matemática Aplicada

Banca examinadora: Prof. Dr. Márcia Aparecida Gomes Ruggiero (IMECC-UNICAMP)
Prof. Dr. Sandra Augusta Santos (IMECC-UNICAMP)
Prof. Dr. Luziane Ferreira de Mendonça (UFRJ)

Data da defesa: 14/05/2010

Programa de Pós-Graduação: Mestrado em Matemática Aplicada

Dissertação de Mestrado defendida em 14 de maio de 2010 e aprovada

Pela Banca Examinadora composta pelos Profs. Drs.

Prof.(a). Dr(a). MARCIA APARECIDA GOMES RUGGIERO

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

Prof. (a). Dr (a). LUZIANE FERREIRA DE MENDONÇA

Agradecimentos

Desde a graduação sempre pude contar com a minha família em cada um dos meus passos e não foi diferente quando decidi ingressar no mestrado. Não consegui uma bolsa, mas sempre pude contar com os meus pais e o apoio dos meus irmãos para fazer meu mestrado, por isso o meu agradecimento a Antônio, Maria Lúcia, Vanessa, João e Andréia, que sempre estarão comigo.

Durante o meu mestrado surgiu a oportunidade de emprego e eu deixei Campinas para aproveitar esta chance, mesmo porque não tinha nenhuma renda fixa, e não voltei a me comunicar com a minha orientadora, Marcia, durante um ano, o que é motivo mais do que suficiente para me dispensar da sua orientação, mas quando decidi voltar para Campinas, para terminar a minha dissertação, ela me deu todo apoio sem me cobrar pela minha falta e por isso consegui finalizar o meu mestrado. Por esta razão, e pelos 4 anos de orientação e dedicação, os meus mais sinceros agradecimentos a minha orientadora Marcia Aparecida Gomes Ruggiero.

Agradeço também aos meus colegas de graduação e mestrado, pois não se faz uma gradução e uma pós graduação em matemática sem contar com ajuda dos seus companheiros. Por isso agradeço a cada um de vocês pelas listas resolvidas e por todas os outros desafios que compartilhamos.

Agradecimentos também a todos os moradores da casa M7 da moradia da Unicamp com quem compartilhei um teto e enfrentei o desafio da convivência, desde a graduação até o final do mestrado.

E os meus agradecimentos finais àqueles que decidiram enfrentar a leitura deste texto e a minhas desculpas antecipadas pelos eventuais erros.

Boa leitura!

Resumo

Uma das estratégias empregadas para resolver um problema de programação não linear com restrições é usar métodos iterativos que geram uma seqüência de pontos viáveis. A razão é que frequentemente soluções viáveis são úteis em aplicações da engenharia, física ou química, ao contrário das aproximações não viáveis, até mesmo quando estas estão bem próximas do valor ótimo. Porém, quando lidamos com restrições não lineares não suaves, é difícil manter viabilidade e, simultaneamente, melhorar o valor da função objetivo.

Uma alternativa é empregar métodos de Restauração Inexata. Em linhas gerais, nestes métodos, a cada iteração dois novos pontos são gerados, um que visa melhorar a viabilidade e outro que diminui o valor da função objetivo. Um terceiro ponto é obtido de modo a atingir um decréscimo mínimo de uma função de mérito composta pelos dois primeiros pontos e que busca o equilíbrio entre viabilidade e otimalidade.

Ao processo de encontrar o ponto que melhora a viabilidade, damos o nome de restauração e o objetivo central deste trabalho é analisar esta fase. Analisamos problemas de otimização onde as restrições são não lineares acrescidas por restrições de canalização (limitantes inferior e superior para as variáveis). Para realizar a restauração usamos o método proposto por J. B. Francisco, N. Krejić e J. M. Martínez [11], no qual são considerados sistemas não lineares com restrições de canalização e que faz uso de uma estratégia de região de confiança com escalamento.

O método de Restauração Inexata que usamos é baseado no algoritmo proposto por M. A. Gomes Ruggiero, J. M. Martínez e S. A. Santos [13] que emprega a direção do gradiente espectral projetado [4] para resolver o problema de *hard-spheres*, onde a restauração pode ser sempre feita de maneira exata. Neste trabalho resolvemos problemas nos quais a fase de restauração não é necessariamente feita de maneira exata.

Os testes computacionais, realizados com problemas acadêmicos, atestam a eficiência do esquema proposto. Usando o algoritmo proposto em [11] para realizar a fase de restauração, implementamos no software MatLab 7.7 o algoritmo do método de Restauração Inexata, encontrado em [13], utilizando o mesmo conjunto de problemas teste usados em [11] além de outros encontrados em [14], obtendo bons resultados.

Palavras-chave: Otimização Não Linear, Restauração Inexata, Sistemas Não Lineares, Viabilidade, Parâmetro Espectral.

Abstract

One of the strategies employed to solve a nonlinear programming problem with constraints is to use iterative methods that generate a sequence of points feasible. The reason is that viable solutions are often useful in applications engineering, physics or chemistry, unlike the approaches are not viable, even when they are very close to the optimum value. But when dealing with soft constraints nonlinear, it is difficult to maintain viability and, simultaneously, improve the value of the objective function.

An alternative is to employ methods of Inexact Restoration. In general, these methods, each iteration two new points are generated, one that aims to improve the viability and another that decreases the value of the objective function. A third point is obtained in order to achieve a decrease of at least a merit function consisting of the first two points and that seeks a balance between feasibility and optimality.

The process of finding the point that improves the viability, we give the name of restoration and purpose of this paper is to analyze this phase. We analyze optimization problems where the constraints are nonlinear constraints added by channeling (lower and upper bounds for variables). To accomplish the restoration we use the method proposed by Mr B. Francis, N. Kreji'c and J. M. Martinez [11], which are considered non-linear systems with restricted channel that uses a trust region strategy with scaling.

The Inexact restoration method we use is based on the algorithm proposed by M. A. Gomes Ruggiero, J. M. Martínez and S. A. Santos [13] that employs the spectral projected gradient direction [4] to solve the problem of hard-spheres, where the restoration can be done in exactly. Present paper, problems in which phase of restoration is not necessarily done exactly.

The computational tests carried out with academic problems, proving the efficiency of the proposed scheme. Using the algorithm proposed in [11] to accomplish the restoration phase, implemented in MatLab 7.7 the algorithm of the method of Inexact Restoration, found in [13], using the same set of test problems used in [11] and other found in [14], obtaining good results.

Keywords: Nonlinear Optimization, Inexact Restoration, Nonlinear Systems, Feasibility, Spectral Parameter.

Sumário

Capítulo 1

Introdução

Otimização é uma importante área da matemática aplicada cujos resultados são largamente utilizados na ciência de tomadas de decisão e na análise de sistemas físicos. Para usá-la, devemos primeiro identificar alguma função que represente o desempenho sob estudo. Tal função pode ser o lucro, tempo, energia potencial, e a meta é encontrar valores das variáveis que otimizam o valor desta *função objetivo*. Frequentemente as variáveis devem satisfazer algumas condições, denominadas *restrições*. Por exemplo, quantidades tais como densidade de elétrons de uma molécula ou a taxa de juros de um empréstimo não podem ser negativas.

O processo de identificar o objetivo, variáveis e restrições para um dado problema é conhecido como modelagem. Construir um modelo apropriado é o primeiro passo - algumas vezes o mais importante - no processo de otimização. Se o modelo for muito simples, não representará o problema original, mas se for muito complexo, pode se tornar muito difícil, ou mesmo impossível numericamente, de resolver. Neste trabalho, usamos o seguinte modelo onde as variáveis podem ser expressas como um vetor $x \in \mathbb{R}^n$ e o objetivo como uma função $f : \mathbb{R}^n \to \mathbb{R}$:

$$\begin{aligned} \min \quad & f(x) \\ \text{s. a} \quad & C(x) = 0 \\ & l \leq x \leq u, \end{aligned} \tag{1.1}$$

onde $C : \mathbb{R}^n \to \mathbb{R}^m$ é uma função cujas componentes $C_i(x),\ i = 1 \ldots m$ são funções continuamente diferenciáveis e as componentes dos vetores l e u fornecem os limitantes inferiores e superiores, respectivamente, às componentes de x. A solução x^* de um problema de otimização como o (1.1), também chamada de minimizador, é local se $f(x^*) \leq f(x)$ para qualquer x pertencente a uma vizinhança de x contida no domínio de f e global se $f(x^*) \leq f(x)$ para qualquer x pertencente ao domínio de f. Para

este tipo de problema podemos empregar métodos que trabalham exclusivamente com pontos que pertencem a $\mathcal{D} = \{x \in \mathbb{R}^n : C(x) = 0 \text{ e } l \leq x \leq u\}$, aos quais chamaremos de pontos viáveis. A preferência por métodos de pontos viáveis vem da necessidade de satisfazer restrições que se referem, por exemplo, a alguma lei física que não pode ser violada, pois uma solução inviável não faria sentido. Mas, dependendo da curvatura de C, fica impraticável usar estratégias de pontos viáveis, pois a busca por um novo ponto viável pode resultar em um passo pequeno, o que pode deixar o método muito lento. Temos como exemplo de métodos viáveis o método de pontos interiores e o método Simplex ([22], p. 374), usados para problemas de otimização linear e o método de restrições ativas para problemas com restrições lineares ([22], p. 430).

Existem métodos que trabalham com pontos não necessariamente viáveis, como os de Penalização, Lagrangiano Aumentado e Programação Quadrática Seqüencial (PQS)([22], p. 529).

Por exemplo, a cada iteração de um método de penalização aplicado ao problema (1.1), o subproblema

$$\begin{aligned} \min_{x} \quad & P(x, \mu_k) = f(x) + \mu_k p(C(x)) \\ \text{s. a} \quad & l \leq x \leq u \end{aligned} \tag{1.2}$$

é resolvido, onde μ_k é um escalar positivo e $p(C(x)) \geq 0$, sendo que $p(C(x^k)) = 0$ se e somente se $C(x^k) = 0$, por exemplo, $p(C(x)) = \|C(x)\|_2^2$. Aplica-se um algoritmo de minimização para problemas com restrições de canalização em (1.2), escolhendo x^k como ponto inicial ([22]), para determinar o próximo iterando x^{k+1} como solução deste problema. Fazendo $\mu_k \to \infty$ devemos encontrar o minimizador x^* do problema (1.1) que deve satisfazer $C(x) = 0$ e ao mesmo tempo dar o menor valor possível para $f(x)$. A dificuldade deste método está na escolha de μ_k a cada iteração. Se escolhermos μ_k muito grande daremos maior peso as restrições e o valor de f não terá uma redução satisfatória, mas se o valor de μ_k for pequeno f terá uma redução adequada, porém o valor de $p(C(x^k))$ não será reduzido suficientemente, o que mostra um ponto x^k longe da viabilidade.

Uma alternativa ao método de penalização é o método do Lagrangiano Aumentado:

$$\min_{x} \mathbb{L}(x, \lambda_k, \mu_k) = f(x) + \lambda_k^T C(x) + \mu_k p(C(x)). \tag{1.3}$$

O processo de resolução é semelhante ao do método de penalização, mas inclui o vetor de multiplicadores de Lagrange $\lambda_k \in \mathbb{R}^m$ que deve ser atualizado junto com μ_k.

Já o método de Programação Quadrática sequencial (PQS) aproxima o problema original (1.1) por um problema quadrático da seguinte maneira:

$$\min \quad Q(p)$$

$$\begin{aligned} \text{s. a} \quad & L(p) = 0 \\ & l \leq x^k + p \leq u \end{aligned} \tag{1.4}$$

onde $Q(p)$ é uma aproximação quadrática em torno do ponto x^k para f e $L(p)$ é uma aproximação linear para C numa vizinhança de x^k. O próximo iterando será definido por $x^{k+1} = x^k + p^*$, onde p^* é solução de (1.4).

Uma alternativa a estes algoritmos é o método de Restauração Inexata, que decompõe a iteração em duas fases, uma relacionada à otimalidade no subespaço tangente e a outra relacionada à viabilidade.

Em resumo, o método de Restauração Inexata consiste em, a partir de um ponto x, obter um ponto candidato y satisfazendo uma aproximação linear das restrições, tal que a medida da inviabilidade em y seja uma fração da inviabilidade em x. Uma vez determinado y, calcula-se z, quase sempre inviável, pela minimização da função objetivo em um aproximação tangente das restrições. Se z satisfaz determinada condição de decréscimo suficiente, então o próximo iterando é definido com z. Caso contrário a distância entre y e z deve ser diminuída até que a condição seja satisfeita.

Para determinar o ponto y, como estamos tratando especificamente de um problema com restrições de canalização, usaremos um algoritmo específico para resolver sistemas não lineares com restrições de canalização proposto em [11].

Neste trabalho a proposta central é analisar um algoritmo de Restauração Inexata para problemas com restrições de igualdade, no qual a fase de Restauração seja de fato realizada de forma aproximada (inexata). Vale mencionar que a única referência de nosso conhecimento em que a fase de restauração também é feita resolvendo-se efetivamente um problema de otimização é o artigo [5] , em que os autores analisam a convergência local de um método de restauração inexata e apresentam um cojunto de experimentos numéricos com problemas da coleção CUTE, cujo desempenho numérico é comparado ao do LANCELOT. Propomos ainda formas de inicializar o parâmetro espectral, que é essencial para a obtenção da direção de descida na fase de otimalidade.

Destacamos ainda como contribuição, o estudo numérico da proposta de repetição do passo tangente, na fase de Restauração.

O Capítulo 2 descreve o método de Restauração Inexata, com o passo espectral [4], que é uma maneira de aproximar a matriz Hessiana de f. O Capítulo 3 descreve o método para resolver sistemas não lineares com restrições de canalização [11], mostrando o algoritmo clássico de região de confiança, a matriz de escala e o algoritmo usado na fase de restauração. O Capítulo 4 apresenta resultados de testes numéricos para o algoritmo de Restauração Inexata, comentando a implementação de cada método e os resultados obtidos. No Capítulo 5 damos as nossas conclusões apresentando a linha de futuras pesquisas.

1.1 Notações

1. $\mathbb{R}^n$ é o conjunto dos vetores coluna
$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}.$$
2. $x^T = (x_1, x_2, ..., x_n)$.
3. $\langle x, y \rangle = x^t y = x_1 y_1 + x_2 y_2 + ... + x_n y_n$.
4. $\|x\| = (x^t x)^{\frac{1}{2}}$ (norma Euclidiana).
5. $|.|$ é uma norma monótona em $\mathbb{R}^m$ tal que $|v| \leq |w|$, sempre que $0 \leq v \leq w$ componente a componente.
6. Para $x, y \in \mathbb{R}^n, x \leq y$, significa que $x_i \leq y_i$ para todo $i \in \{1, 2, ..., n\}$.
7. $\mathcal{B}(x, \varepsilon) = \{y \in \mathbb{R}^n | \, \|y - x\| < \epsilon\}$ (vizinhança de x).
8. $\mathbb{R}^{m \times n}$ é o conjunto de matrizes $m \times n$. Se $A \in \mathbb{R}^{m \times n}$, denotamos A^T como sua matriz transposta.
9. Gradiente de f:
$$\nabla f(x) = \begin{pmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{pmatrix}.$$
10. Matriz Hessiana de f: $\nabla^2 f(x)_{ij} = \left[\frac{\partial^2 f(x)}{\partial x_i \partial x_j}\right]$.
11. Se $G : \mathbb{R}^m \to \mathbb{R}^p$, $G'(x) \in \mathbb{R}^{p \times m}$ denota a matriz Jacobiana de G em x. A j-ésima linha de $G'(x)$ é $\nabla^t G_j(x)$.
12. $\mathcal{C}^k$ denota o conjunto de funções $f : \mathbb{R}^n \to \mathbb{R}$ tais que todas as derivadas de ordem menor ou igual a k são contínuas.
13. Se a matriz A é semi definida positiva ($x^t A x \geq 0$ para todo $x \in \mathbb{R}^n$), escrevemos $A \geq 0$. Analogamente, se A é definida positiva ($x^t A x > 0$ para todo $x \in \mathbb{R}^n$, $x \neq 0$), escrevemos $A > 0$.

Capítulo 2

Restauração Inexata

2.1 Introdução

Problemas de otimização não linear são frequentemente abordados com métodos que utilizam somente pontos viáveis. O motivo é que para aplicações práticas é mais interessante determinar soluções viáveis não ótimas, que satisfaçam restrições que modelam limites inerentes ao problema, tais como: massa e tempo positivos, conservação de energia, a soma de diferentes investimentos de um portfólio menor ou igual a um determinado capital inicial, ou ainda um limite inferior e superior para produção de uma indústria, e que portanto são úteis em aplicações da engenharia, física, química, finanças e outras áreas. Uma solução que se aproxime mais da solução ótima, mas que não seja viável, pode perder seu significado e portanto a sua utilidade.

Na década de 80 poucos trabalhos da principal corrente da literatura da otimização foram dedicados a métodos viáveis. Aquela década foi dominada pelos modelos SQP (*sequential quadratic programming* - programação quadrática sequencial). A razão era a crítica usual contra os métodos viáveis que frequentemente apresentavam dificuldade em conseguir bons avanços em regiões muito curvas, especialmente quando a aproximação corrente estava muito afastada da solução.

Dentro deste cenário, foi idealizado o método de Restauração Inexata em 2000 [18], primeiramente para lidar com restrições de desigualdade e posteriormente, em 2004 [19], para tratar restrições de igualdade. Este método não trabalha necessariamente com pontos viáveis, o que facilita a convergência para solução de um problema com restrições com grande curvatura. Aplicaremos o método de Restauração Inexata para abordar o seguinte problema de otimização:

$$\text{Minimizar} \quad f(x) \tag{2.1}$$

$$\begin{aligned} \text{Sujeito a} \quad & C(x) = 0 \\ & l \leq x \leq u, \end{aligned}$$

onde $f : \mathbb{R}^n \to \mathbb{R}$ e $C : \mathbb{R}^n \to \mathbb{R}^m$, com $m \leq n$, são funções continuamente diferenciáveis e $l, u \in \mathbb{R}^n$ são os limites para as componentes de $x \in \mathbb{R}^n$.

A estratégia de Restauração Inexata usa um algoritmo iterativo que gera pontos viáveis com respeito às restrições de canalização:

$$\Omega = \{x \in \mathbb{R}^n : l \leq x \leq u\}, \tag{2.2}$$

isto é,

$$x^k \in \Omega, \quad \text{para } k = 0, 1, 2, \ldots.$$

A abordagem que escolhemos neste trabalho segue a proposta apresentada em [13]. A cada iteração são incluídos dois procedimentos diferentes: restauração e minimização. No passo de restauração, um ponto intermediário $y^k \in \Omega$ é encontrado tal que a inviabilidade em y^k seja uma fração da inviabilidade em x^k.

No passo de minimização, imediatamente após a restauração, construímos uma aproximação linear das restrições usando a informação em y^k. Calculamos o ponto teste z^k pertencente a uma aproximação linear das restrições, tal que $f(z^k)$ seja suficientemente menor que $f(y^k)$, realizando uma busca linear em uma direção de descida d.

O ponto teste z^k é aceito como uma novo iterando se o valor de uma função de mérito em z^k é suficientemente menor do que em x^k. Se z^k não é aceito, o processo de busca linear é repetido na direção de d.

Como Ω é um politopo, a região definida pela aproximação linear das restrições e o próprio Ω também será um politopo.

Como já mencionado, o algoritmo é relacionado a métodos viáveis clássicos para programação não linear, tais como o gradiente reduzido generalizado (GRG) e a família de algoritmos de restauração de gradiente sequencial (*sequential gradient restoration*), (SRGA). No entanto, em nossa abordagem, as aproximações sucessivas de (2.1) não são necessariamente viáveis (ou aproximadamente viáveis) com respeito a $C(x) = 0$.

Uma descrição do algoritmo é apresentada na Seção 2.2, juntamente com suas motivações.

2.2 Descrição do Método de Restauração Inexata

O algoritmo de Restauração Inexata neste trabalho foi adaptado do apresentando em [13], proposto para abordar problemas com restrições de desigualdade. A diferença

é que neste trabalho abordaremos apenas problemas com restrições de igualdade incluindo um procedimento específico para a fase de Restauração que será realizada de forma aproximada. Os testes computacionais da referência [13] foram realizados com um conjunto específico de problemas denominados *hard-spheres problems* que possibilitam a restauração exata de uma maneira trivial. A proposta neste trabalho é resolver problemas nos quais a restauração seja de fato realizada de forma inexata. Descreveremos a seguir as principais etapas deste algoritmo.

2.2.1 Restauração

Dada uma aproximação $x^k \in \Omega$, obtém-se um ponto intermediário $y^k \in \Omega$, que esteja mais próximo da região viável. Tal ponto y^k deve satisfazer as seguintes condições:

$$\left\|C(y^k)\right\| \leq r \left\|C(x^k)\right\|, \tag{2.3}$$

$$\left\|y^k - x^k\right\| \leq \beta \left\|C(x^k)\right\|, \tag{2.4}$$

onde $r \in [0, 1]$ e $\beta > 0$ são parâmetros dados independentemente de k. A condição (2.3) estabelece a necessidade de se obter um ponto intermediário que seja pelo menos tão viável quanto x^k. A condição (2.4) restringe y^k a uma vizinhança ao redor de x^k. O raio desta região é reduzido à medida que a condição de viabilidade (2.3) é satisfeita. Além disso, a condição (2.4) impõe que y^k deve ser igual a x^k se o ponto corrente é viável.

Para realizar esta etapa usamos um método para resolver sistemas não lineares com restrições de canalização proposto em [11] e que será descrito no Capítulo 3.

2.2.2 Espaço Tangente e Direção de Descida

Após o cálculo de y^k satisfazendo (2.3) e (2.4), definimos uma aproximação linear da região viável de (2.1), contendo o ponto intermediário y^k:

$$\pi_k = \left\{x \in \Omega \mid C'(y^k)(x - y^k) = 0\right\}. \tag{2.5}$$

Então, π_k é a intersecção de Ω com a aproximação linear de $C(x) = 0$ em torno de y^k. Perceba que π_k é um espaço afim paralelo ao $\mathcal{N}(C'(y^k))$, restrito a Ω.

A direção de descida é obtida seguindo a proposta de [13]:

$$d^k_{tan} = \mathcal{P}^k(y^k - \eta \nabla f(y^k)) - y^k, \tag{2.6}$$

onde $\mathcal{P}^k(z)$ denota a projeção ortogonal de z em π_k, conforme é ilustrado na Figura 2.1. O parâmetro $\eta > 0$ será escolhido como o parâmetro espectral [4]. O critério de convergência do algoritmo será baseado na norma deste vetor, $\left\|d^k_{tan}\right\|$.

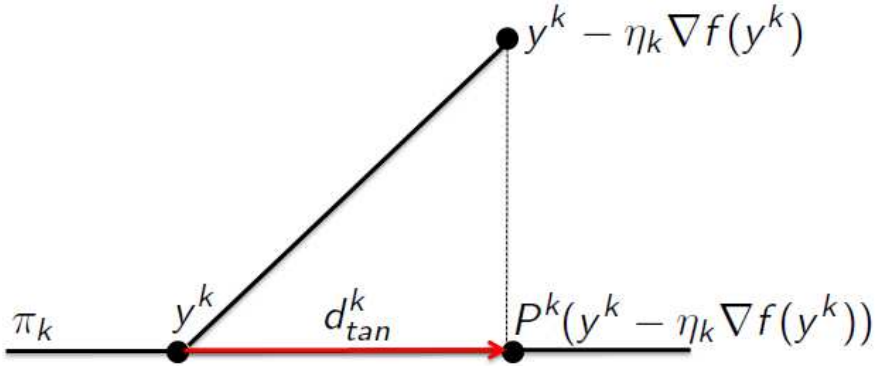

Figura 2.1: O passo tangente $d^k_{tan} = \mathcal{P}^k(y^k - \eta\nabla f(y^k)) - y^k$.

É fácil verificar que d^k_{tan} é uma direção de descida. Desde que $y^k \in \pi_k$, devido à projeção ortogonal, temos que:

$$\left\|(y^k - \eta\nabla f(y^k)) - \mathcal{P}^k(y^k - \eta\nabla f(y^k))\right\|^2 \leq \left\|(y^k - \eta\nabla f(y^k)) - y^k\right\|^2.$$

Portanto,

$$\left\|-\left[\mathcal{P}^k(y^k - \eta\nabla f(y^k)) - y^k\right] - \eta\nabla f(y^k))\right\|^2 \leq \left\|(y^k - \eta\nabla f(y^k)) - y^k\right\|^2,$$

da definição (2.6) temos:

$$\left\|-d^k_{tan} - \eta\nabla f(y^k))\right\|^2 \leq \left\|(y^k - \eta\nabla f(y^k)) - y^k\right\|^2,$$

$$\left\|d^k_{tan} + \eta\nabla f(y^k))\right\|^2 \leq \left\|(\eta\nabla f(y^k))\right\|^2,$$

$$\left\|d^k_{tan}\right\|^2 + \left\|\eta\nabla f(y^k)\right\|^2 + 2\eta(d^k_{tan})^T\nabla f(y^k) \leq \left\|\eta\nabla f(y^k)\right\|^2,$$

logo

$$(d^k_{tan})^T\nabla f(y^k) \leq -\frac{1}{2\eta}\left\|d^k_{tan}\right\|^2, \tag{2.7}$$

portanto d^k_{tan} é direção de descida.

Após determinar d^k_{tan}, podemos calcular um novo ponto $z^k \in \pi_k$ que reduz o valor de f, tal que $z^k = y^k + d^k_{tan}$. Porém devemos lembrar que $z \in \Omega$, ou seja, devemos calcular d^k_{tan} de tal maneira que $y + d^k_{tan} \in \Omega$.

Uma vez que π_k é um espaço afim paralelo ao $\mathcal{N}(C'(y^k))$, de (2.6) temos que d^k_{tan} é a projeção ortogonal do vetor $(y^k - \eta_k \nabla f(y^k))$ em π_k. Daí, d^k_{tan} é obtido como a solução do problema:

$$\begin{array}{cc} \min & \frac{1}{2}\left\|(d+y^k) - (y^k - \eta_k \nabla f(y^k))\right\|^2 \\ s.a & y^k + d \in \pi_k. \end{array} \tag{2.8}$$

Dado que $\pi_k = \left\{x \in \Omega \mid C'(y^k)(x - y^k) = 0\right\}$ temos que d^k_{tan} é solução de

$$\begin{array}{cc} \min & \frac{1}{2}\left\|d + \eta \nabla f(y^k)\right\|^2 \\ s.a & C'(y^k)d = 0 \\ & y^k + d \in \Omega \end{array} \tag{2.9}$$

2.2.3 Repetição do Passo Tangente

Uma vez que a direção de descida d^k_{tan} (2.6) tenha sido obtida através da resolução do problema (2.9) o ponto $z^k = y^k + d^k_{tan}$ é calculado e pode não satisfazer as condições de viabilidade (2.3) e (2.4) impostas para obtenção de y^k. Se forem satisfeitas, o processo pode ser repetido, isto é: a partir de z^k, calcular uma nova direção d^k_{tan}. Esta estratégia foi proposta em [13], numa tentativa de melhorar o desempenho do algoritmo e é considerada não monótona, pois a repetição dos passos é realizada sem considerar se o valor de f é reduzido ou não.

Na Figura 2.2, exemplificamos esta estratégia: a partir de um ponto y^k_0 conseguimos um novo ponto z^k_0 que ainda satisfaz as condições de viabilidade (2.3) e (2.4), e a partir dele conseguimos mais dois pontos z^k_1 e z^k_2 que ainda satisfazem estas condições, mas o ponto z^k_3 não satisfaz a condição de viabilidade e é descartado, logo o ponto z^k_2 será aceito como o ponto z^k da fase de otimização. Porém, não é aconselhável repetir este processo indefinidamente, pois a direção d^k_{tan} pode diminuir muito e o processo pode estagnar. Por isso definimos, arbitrariamente, um número máximo de repetições deste processo, que chamaremos de k^{max}_{trial} e a iteração corrente será k_{trial}. Este processo pode ser formalizado no Algoritmo 2.2 (algoritmo p. 1634 de [13]).

2.2.4 Decréscimo no Espaço Tangente

Agora consideraremos o problema de encontrar o passo de minimização t para a direção de descida d^k_{tan}.

O ideal para determinar t seria encontrar o valor que minimiza a função objetivo f na direção de descida, ou seja, minimizar a função

$$\phi(t) = f(y^k + t d^k_{tan}),$$

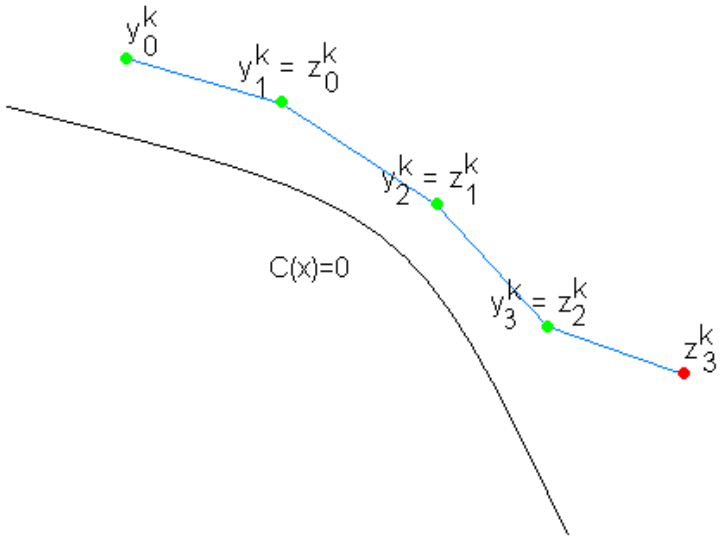

Figura 2.2: Tentando reduzir o valor de f uma vez que o critério de viabilidade é satisfeito.

em t, o que é fácil quando a função f é uma quadrática, por exemplo. Porém, na maioria dos casos, o custo computacional de se resolver este problema de otimização é muito elevado quando pensamos que isto será apenas uma etapa de um outro algoritmo de otimização. Por esta razão é conveniente encontrar um valor para t que satisfaça uma critério de decréscimo suficiente, como o critério de Armijo:

$$f(y^k + td) \leq f(y^k) + \alpha t \nabla f(y^k)^T d, \tag{2.10}$$

onde d é uma direção de descida e $\alpha \in (0, 1]$.

Mostramos que d^k_{tan} é uma direção de descida, isto é $\phi'(0) = \nabla f(y^k)^T d^k_{tan} < 0$. Portanto para satisfazer (2.10) podemos nos restringir a valores positivos de t. Por isto usamos a busca linear para encontrar um valor adequado para t.

Neste trabalho usamos o *backtracking* que é realizado do seguinte modo: o passo t será definido como o primeiro termo da seqüência $\{t_1, t_2, \ldots\}$ que satisfaça

$$f(y^k + td^k_{tan}) \leq f(y^k) + \alpha t \nabla f(y^k)^T d^k_{tan},$$

onde $\{t_j\}$ é definido por $t_1 = 1$ e $t_{j+1} \in [0.1t_j, 0.9t_j]$ para todo $j = 1, 2, \ldots$. Então, definimos $z^k = y^k + td^k_{tan}$.

2.2.5 Passo Espectral

O método do gradiente espectral projetado foi originado do trabalho de Barzilai e Borwein [1]. A ideia é que a direção do gradiente, quando conjugada com o comprimento adequado do passo espectral, produz resultados muito melhores do que aqueles da tradicional aproximação de Cauchy que, muitas vezes, são competitivos com relação àqueles obtidos pelos métodos de Newton e quasi-Newton, sendo frequentemente atrativos devido ao seu baixo custo computacional.

O método original de Barzilai e Borwein foi generalizado por Raydan [23] para problemas de minimização irrestrita de grande porte. Mais tarde, o método foi estendido para problemas convexos [4]. O método do gradiente espectral projetado definido em [4] se mostrou extremamente eficiente na resolução de problemas de grande porte nos quais as projeções podem ser calculadas facilmente [6].

Estes fatos motivaram a idéia de encontrar uma extensão natural do método do gradiente espectral para problemas com restrições não lineares, onde as projeções sobre os conjuntos viáveis são de alto custo computacional.

Observamos que d^k_{tan} é obtido como a projeção do vetor $-\eta\nabla f(x)$ em $\mathcal{N}(C'(y^k))$. Seguindo [13], η é o parâmetro ou passo espectral.

Antes de apresentar o cálculo do passo espectral, vamos comentar os métodos de Máxima Descida e de Newton para minimização irrestrita. No método de máxima descida escolhemos como direção de descida o vetor $-\nabla f(x)$. Com esta estratégia avança-se rapidamente para o mínimo da função, mas a uma certa distância do ponto ótimo, os passos em direção a solução são cada vez menores. Uma alternativa solução para este inconveniente é dada pelo método de Newton, onde a direção de descida é definida por $p = -\nabla^2 f(x^k)^{-1}\nabla f(x^k)$, que minimiza a aproximação quadrática para f em torno x^k:

$$f(x^k + p) \approx q(p) = f(x^k) + \nabla f(x^k)^T p + p^T \nabla^2 f(x^k) p.$$

Poderíamos fazer algo semelhante para o método de Restauração Inexata, mas o cálculo da matriz Hessiana pode ser muito trabalhoso. Portanto seria interessante aproximar $\nabla^2 f(x^k)$ por uma matriz H que possuísse as características da matriz Hessiana, mas que fosse de baixo custo computacional. Para construir uma aproximação H da matriz Hessiana de forma a não prejudicar o desempenho do algoritmo, podemos empregar uma matriz diagonal cujas entradas são todas iguais. Quando as segundas derivadas existem e a função é Lipschitz contínua, o teorema de Taylor implica que ([22], p. 173)

$$\nabla f(x + p) = \nabla f(x) + \nabla^2 f(x) p + \mathcal{O}(\|p\|^2). \tag{2.11}$$

Como $f \in \mathcal{C}^2$, podemos pensar em (2.11) como uma aproximação linear para o gradiente com um erro da ordem de $\|p\|^2$. Substituindo $x = x^k$ e $p = x^{k+1} - x^k$ em (2.11), obtemos:

$$\nabla f(x^{k+1}) = \nabla f(x^k) + \nabla^2 f(x^k)(x^{k+1} - x^k) + \mathcal{O}(\left\|x^{k+1} - x^k\right\|^2).$$

Quando x^k e x^{k+1} estão em uma região próxima da solução x^*, na qual $\nabla^2 f$ é definida positiva, o termo final nesta expansão é eventualmente dominado pelo termo $\nabla^2 f(x^k)(x^{k+1} - x^k)$, e podemos escrever

$$\nabla^2 f(x^k) \approx \nabla f(x^{k+1}) - \nabla f(x^k). \tag{2.12}$$

Escolhemos a aproximação H_k da matriz Hessiana impondo que a equação secante ([9] p. 195) seja satisfeita:

$$H_{k+1} s^k = u^k \tag{2.13}$$

onde definimos:

$$s^k = x^{k+1} - x^k,\ u^k = \nabla f(x^{k+1}) - \nabla f(x^k).$$

Podemos reescrever a equação secante como

$$s^k = H_{k+1}^{-1} u^k. \tag{2.14}$$

Como queremos aproximar a matriz Hessiana por uma matriz diagonal com todas as entradas iguais, a sua inversa também será uma matriz deste tipo, e multiplicar essa matriz por um vetor é o mesmo que multiplicar por um escalar que é igual ao elemento da diagonal. Chamando este elemento, convenientemente de η_k, temos:

$$s^k = \eta_k u^k, \tag{2.15}$$

multiplicando ambos os lados da equação por ${s^k}^T$,

$${s^k}^T s^k = \eta_k {s^k}^T u^k,$$

e isolando η_k na equação, temos o passo espectral:

$$\eta_k = \frac{{s^k}^T s^k}{{s^k}^T u^k}. \tag{2.16}$$

A expressão para η_k concorda com a expressão do coeficiente de Rayleigh para estimar autovalores de uma matriz numericamente ([24] p. 324). Voltando ao problema (2.9) que determina d_{tan}^k, podemos expandir a sua função objetivo e obter

$$\frac{1}{2}\left\|d + \eta_k \nabla f(y^k)\right\|^2 = \frac{1}{2}\left(\|d\|^2 + 2\eta_k d^T \nabla f(y^k) + {\eta_k}^2 \left\|\nabla f(y^k)\right\|^2\right),$$

considerando que estamos minimizando a função em d, temos que o seu último termo é constante, logo pode ser desconsiderado. Se dividirmos esta expressão por $2\eta^k$ teremos a expressão equivalente:

$$\frac{1}{2\eta_k}\|d\|^2 + d^T\nabla f(y^k).$$

Observação:
A matriz escalar η_kI, com a escolha (2.12), é a que melhor aproxima a matriz inversa da Hessiana média. Do Teorema do Valor Médio do Cálculo Integral ([22], p. 15) sabemos que dada uma função continuamente diferenciável $f : \mathbb{R}^n \to \mathbb{R}$, temos que

$$f(x+s) = f(x) + \nabla f(x+\alpha s)^T s,$$

para algum $\alpha \in (0,1)$. Ainda, se f é duas vezes continuamente diferenciável, então

$$\nabla f(x+s) = \nabla f(x) + \int_0^1 \nabla^2 f(x+\alpha s)d\alpha s.$$

Assim, o escalar:

$$\eta = \frac{s^T u}{s^t s},$$

onde $u = \nabla f(x+s) - \nabla f(x)$, ou ainda pelo Teorema do Valor Médio do Cálculo Integral

$$u = \left(\int_0^1 \nabla^2 f(x+\alpha s)d\alpha\right) s,$$

define um quociente de Rayeligh relativo à matriz inversa da Hessiana média $\left(\int_0^1 \nabla^2 f(x+\alpha s)d\alpha\right)$. O valor deste quociente está entre o menor e o maior autovalor da matriz inversa da Hessiana média, o que motiva a denominação parâmetro espectral e seu uso na aproximação da matriz Hessiana.

O sinal de η determina a curvatura da função objetivo de (2.9). Portanto η deve ser sempre positivo para que (2.9) tenha solução finita. Dessa maneira redefinimos o parâmetro

$$\eta_k = \begin{cases} \min\left\{\eta_{max}, \max\left\{\eta_{min}, \frac{s^{k^T}s^k}{s^{k^T}u^k}\right\}\right\} & \text{se } s^{k^T}u^k > 0 \\ \eta_{max} & \text{caso contrário} \end{cases} \tag{2.17}$$

Com esta definição η_k fica limitado inferiormente por η_{min}, para evitar que se aproxime muito do zero, e superiormente por η_{max} para prevenir erros, pois ηI é apenas uma aproximação para matriz Hessiana. Esta salvaguarda assegura que o modelo se manterá convexo, projetando o valor de η_k no intervalo $[\eta_{min}, \eta_{max}]$ tal que a aproximação da Hessiana fique limitada.

2.2.6 Função de Mérito e Parâmetro de Penalidade

A comparação de x^k, y^k e z^k envolve a avaliação de uma função de mérito nestes pontos. Usamos a seguinte função de mérito, utilizada em [18] e [13]:

$$\psi(x,\theta) = \theta f(x) + (1-\theta)\left\|C(x)\right\|, \tag{2.18}$$

onde $\theta \in (0,1]$ é um parâmetro de penalidade usado para dar diferentes pesos a função objetivo e à viabilidade. A escolha do parâmetro θ em cada iteração depende de considerações práticas e teóricas. Por exemplo, se $\left\|C(x^k)\right\|$ é grande, o peso associado à $f(x)$ deve ser menor, pois o ponto corrente está afastado da região viável, e a escolha do parâmetro de penalidade leva em conta automaticamente esta necessidade prática. Falando superficialmente, desejamos que a função de mérito no novo ponto seja menor que a função de mérito no ponto corrente x^k, isto é, queremos $Ared_k > 0$, onde $Ared_k$ é a redução **real** da função de mérito. Assim como em [18] e [13], $Ared_k$ definimos por:

$$Ared_k = \psi(x,\theta_k) - \psi(z,\theta_k). \tag{2.19}$$

Então,

$$Ared_k = \theta_k[f(x^k) - f(z^k)] + (1-\theta_k)[\left\|C(x^k)\right\| - \left\|C(z^k)\right\|].$$

No entanto, em otimização irrestrita, só a redução da função de mérito não é suficiente para garantir a convergência. Portanto, precisamos de um critério de redução suficiente da função de mérito, que será definido pela verificação do seguinte teste:

$$Ared_k \geq 0.1 Pred_k, \tag{2.20}$$

onde $Pred_k$ é a redução **pretendida** da função de mérito entre x^k, y^k e z^k. Assim como em [18] e [13], definimos $Pred_k$ por:

$$Pred_k = \theta_k[f(x^k) - f(z^k)] + (1-\theta_k)[\left\|C(x^k)\right\| - \left\|C(y^k)\right\|]. \tag{2.21}$$

O valor para $Pred_k$ pode ser não positivo uma vez que podemos ter $f(z^k) > f(x^k)$. Felizmente, se θ_k for pequeno o suficiente, $Pred_k$ é arbitrariamente próximo a $\left\|C(x^k)\right\| - \left\|C(y^k)\right\|$, que é necessariamente não negativo. Portanto, é sempre possível encontrar $\theta_k \in (0,1]$ tal que

$$Pred_k \geq \frac{1}{2}[\left\|C(x^k)\right\| - \left\|C(y^k)\right\|]. \tag{2.22}$$

Usamos a condição (2.22) para garantir que a redução pretendida $Pred_k$ (2.21) tenha um valor mínimo e positivo, evitando que o método de Restauração Inexata fique estagnado. Além disso, podemos ter $\left\|C(x^k)\right\| < \left\|C(z^k)\right\|$, ou seja, os dois termos de $Ared_k$ (2.19)

podem ser negativos, o que representa piora tanto da otimalidade quanto da viabilidade. Por esta razão, temos um forte motivo para realizar o teste de positividade (2.22), a fim de evitar esta situação.

Quando os critérios (2.20) e (2.22) são satisfeitos, aceitamos $x^{k+1} = z^k$, caso contrário realizamos uma nova busca linear na direção de d^k_{tan} para obter um novo valor para z^k e repetimos o teste até que os critérios sejam satisfeitos, completando uma iteração do método de Restauração Inexata, conforme é esquematizado na Figura 2.3.

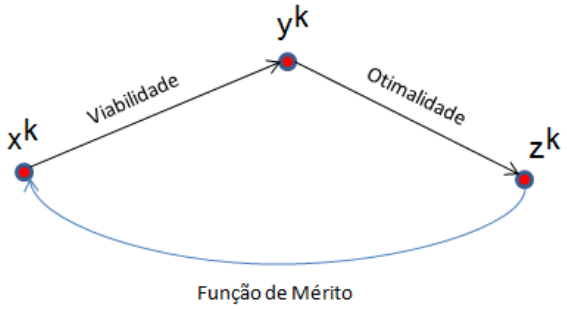

Figura 2.3: Relação entre os pontos x^k, y^k e z^k.

2.3 O Algoritmo de Restauração Inexata

Para a construção do nosso algoritmo seguimos os passos do algoritmo da referência [13] com simples adaptações, pois aqui consideramos restrições de igualdade.

Algoritmo 2.1 *Algoritmo principal:*

***Inicialização**: $k \leftarrow 0$, $\theta_{-1} \in (0,1)$, $r \in [0,1))$, $\alpha \in (0,1]$, $\beta > 0$, $\eta \in [\eta_{min}, \eta_{max}]$ com $0 < \eta_{min} < \eta_{max}$, e $\varepsilon_1, \varepsilon_2 > 0$. Assuma que $\sum_{k=0}^{\infty} \omega_k$ é uma série convergente de termos não negativos. Seja $x^0 \in \Omega$ uma aproximação inicial para solução do problema (2.1).*

***Passo 1**: Restauração.*

Calcule $y^k \in \Omega$ tal que

$$\left\|C(y^k)\right\| \leq r \left\|C(x^k)\right\| \tag{2.23}$$

e

$$\left\|y^k - x^k\right\| \leq \beta \left\|C(x^k)\right\|. \tag{2.24}$$

Se é impossível obter tal y^k, declare "falha em melhorar a viabilidade" e termine a execução.

***Passo 2**: Definição do parâmetro de penalidade.*

Calcule

$$\theta_{k,-1} = \min\left\{1, min\left\{\theta_{-1}, \ldots, \theta_{k-1}\right\} + \omega_k\right\}. \tag{2.25}$$

***Passo 3**: Definição do conjunto tangente para restrições de igualdade.*

$$\pi_k = \left\{z \in \Omega \mid C'(y^k)(z - y^k) = 0\right\}. \tag{2.26}$$

***Passo 4**: Cálculo da direção de busca e verificação do critério de parada, assim como em [18] e [13]*

$$d^k_{tan} = \mathcal{P}^k(y^k - \eta\nabla f(y^k)) - y^k, \tag{2.27}$$

onde $\mathcal{P}^k(z)$ denota a projeção ortogonal de z no conjunto π_k. Se $\left\|C(y^k)\right\| < \varepsilon_1$ e $\left\|d^k_{tan}\right\| < \varepsilon_2$ então termine o algoritmo. Caso contrário, vá para o passo 5.

***Passo 5**: Obter o tamanho do passo da direção tangente.*

Defina t_{dec} como o primeiro termo da seqüência $\{t_1, t_2, \ldots\}$ tal que

$$f(y^k + td^k_{tan}) \leq f(y^k) + \alpha t\nabla f(y^k)^T d^k_{tan}, \tag{2.28}$$

onde $\{t_j\}$ é definida por $t_1 = 1$ e $t_{j+1} \in [0.1t_j, 0.9t_j]$ para todo $j = 1, 2, \ldots$. Faça $i \leftarrow 0$, defina $t_{k,i} \leftarrow t_{dec}$ e $z^{k,i} \leftarrow y^k + t_{dec}d^k_{tan}$.

***Passo 6**: Atualização do parâmetro de penalidade.*

Escolha $\theta_{k,i}$ como o supremo dos valores de θ no intervalo $[0, \theta_{k,i-1}]$ tal que

$$\theta[f(x^k) - f(z^k)] + (1-\theta)[\left\|C(x^k)\right\| - \left\|C(y^k)\right\|] \geq \frac{1}{2}[\left\|C(x^k)\right\| - \left\|C(y^k)\right\|]. \tag{2.29}$$

***Passo 7**: Teste de aceitação.*

Defina

$$Ared_{k,i} = \theta_{k,i}[f(x^k) - f(z^{k,i})] + (1-\theta_{k,i})[\left\|C(x^k)\right\| - \left\|C(z^{k,i})\right\|] \tag{2.30}$$

e

$$Pred_{k,i} = \theta_{k,i}[f(x^k) - f(z^{k,i})] + (1-\theta_{k,i})[\left\|C(x^k)\right\| - \left\|C(y^k)\right\|]. \tag{2.31}$$

Se

$$Ared_{k,i} \geq 0.1Pred_{k,i}, \tag{2.32}$$

defina $x^{k+1} = z^{k,i}$, faça $\theta_k = \theta_{k,i}$, e $iacc(k) = i$, onde iacc significa "i aceito". Atualize $k \leftarrow k+1$ e vá para o Passo 8. Caso contrário, atualize $i \leftarrow i+1$, $t_{k,i} \leftarrow 0.5t_{k,i}$, $z^{k,i} \leftarrow y^k + t_{k,i}d^k_{tan}$ e vá para o passo 6.

***Passo 8**: Atualização do parâmetro espectral (η_k)*

Calcule

$$s^k = x^k - x^{k-1}, \tag{2.33}$$

$$u^k = \nabla f(x^k) - \nabla f(x^{k-1}), \tag{2.34}$$

$$Se\ (s^k)^T u^k \leq 0\ defina\ \eta_{k+1} = \eta_{max} \tag{2.35}$$

$$\text{Caso contrário, } \eta_{k+1} = \max\left\{\min\left\{\frac{(s^k)^T s^k}{(s^k)^T u^k}, \eta_{max}\right\}, \eta_{min}\right\} \tag{2.36}$$

e vá para o Passo 1.

Veja o Fluxograma 2.4 para melhor entendimento do Algoritmo (2.1).

A seguir o algoritmo que repete os passos no espaço tangente uma vez que as condições (4.5) e (4.6) sejam satisfeitas, a fim de melhorar o desempenho do Algoritmo 2.1.

Algoritmo 2.2 *: Repetições de passos tangentes*

$k_{trial} = 1$;
Enquanto $k_{trial} < k_{trial}^{max}$
  $z = y^k + d_{tan}^k$;
  Se *z satisfaz (2.3) e (2.4)*
    $y^k = z$;
    recalcule π_k e d_{tan}^k;
    $k_{trial} = k_{trial} + 1$;
    senão
      $k_{trial} = k_{trial}^{max}$;
    fim
  fim
fim

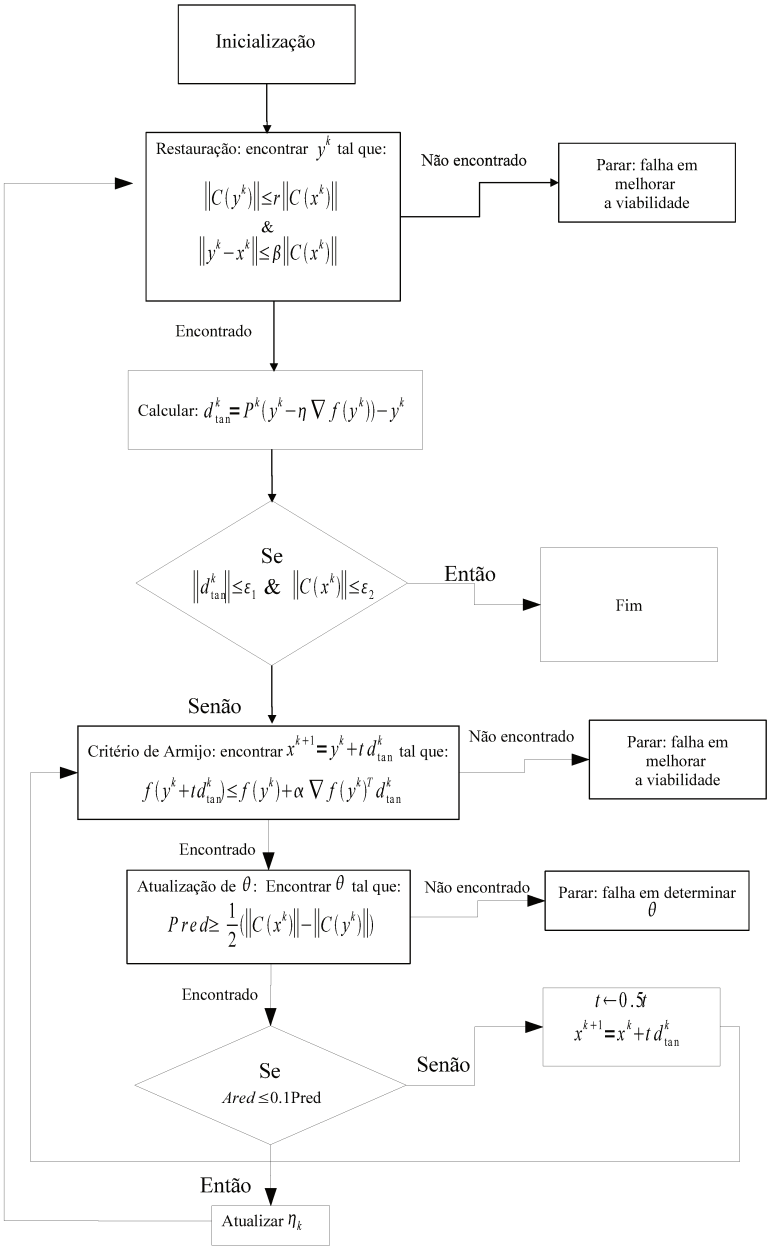

Figura 2.4: Fluxograma do Algoritmo 1

2.3.1 Observações

Um dos principais aspectos do método de Restauração Inexata são as condições (2.23) e (2.24) da fase de restauração da viabilidade. O ponto restaurado não pertence necessariamente ao conjunto viável do problema (2.1), mas deve estar próximo no sentido de (2.23). Além disso o ponto restaurado não precisa estar o mais próximo possível de x mas deve estar adequadamente próximo no sentido de (2.24). Projetar um ponto em um conjunto não convexo, como requerido pelo método do gradiente espectral projetado, é um problema tão difícil quanto o problema de otimização original (2.1). Por esta razão, as projeções usadas no método gradiente espectral devem ser trocadas por passos de restauração parcial com requerimentos mínimos. Após a restauração calculamos a direção d_{tan}^k do método do gradiente espectral na aproximação linear π_k da região factível, conforme é mostrado em [4] e [18].

Outra característica relevante é a escolha do tamanho do passo na direção de d_{tan}^k. A ideia é que direções de descida múltiplas de $-\nabla f$, quando combinadas com o comprimento de passo espectral η_k [4], produzem resultados muito melhores do que o tradicional passo de Cauchy, sendo frequentemente atrativas devido ao seu baixo custo computacional.

Por meio da introdução do parâmetro não negativo ω_k, que é um termo de uma seqüência cuja série é convergente, um aumento moderado do parâmetro de penalidade entre iterações consecutivas é permitido. Isto previne a possibilidade dos parâmetros de penalidade diminuírem artificialmente, um problema inerente ao começo do processo iterativo.

É fácil ver que a seqüência dos parâmetros de penalidade usados em cada iteração θ_k é convergente. De fato, definindo:

$$\theta_{k\ pequeno} = \min\{\theta_{-1}, \ldots, \theta_k\} \text{ e } \theta_{k\ grande} = \theta_{k\ pequeno} + \omega_k,$$

vemos que

$$\theta_{k+1} \leq \theta_{k\ grande} \ e\ \theta_k \geq \theta_{k\ pequeno}, \text{para todo } k.$$

Claramente, $\{\theta_{k\ grande}\}$ e $\{\theta_{k\ pequeno}\}$ convergem para o mesmo limite, então $\{\theta_k\}$ é também convergente pelo teorema do confronto. Podemos também provar por indução que

$$\theta_{k,i} > 0, \text{ para todo } k, i.$$

2.4 Resultados Teóricos

Nesta seção demonstraremos as propriedades teóricas para o Algoritmo 2.1. Inicialmente discutiremos a boa definição, conquista da viabilidade e a convergência a oti-

malidade seguindo as demostrações feitas em [13], mas adaptando-as para o problema (2.1), cujas restrições são de igualdade ao invés de desigualdade.

2.4.1 Boa definição do Algoritmo 2.1

As hipóteses $A1$, $A2$ e $A3$, são os únicos requerimentos sobre o problema de programação não linear (2.1), necessários para provar a convergência, então serão assumidas como válidas daqui por diante. Em particular, as hipóteses de regularidade não serão usadas e derivadas segundas de f e C não são assumidas.

$A1$. O conjunto Ω é convexo e compacto.
$A2$. A matriz Jacobiana $C'(x)$ existe e satisfaz a condição de Lipschitz:

$$\|C'(y) - C'(x)\| \leq L_1 \|y - x\| \text{ para todo } x, y \in \Omega. \tag{2.37}$$

$A3$. O gradiente de f existe e satisfaz a condição de Lipschitz

$$\|\nabla f(y) - \nabla f(x)\| \leq L_2 \|y - x\| \text{ para todo } x, y \in \Omega. \tag{2.38}$$

No problema (2.1), a hipótese $A1$ já é satisfeita, pois o conjunto Ω (2.2) é claramente convexo e compacto, quando os vetores l e u possuem todas as suas componentes finitas, por se tratar de restrições de canalização. Quando uma das componentes de l ou u não for finita, podemos, numericamente, impor limites para estas variáveis com números relativamente maiores do que os valores assumidos pelas incógnitas do problema de otimização (2.1).

Lembramos que nesta dissertação estamos utilizando $\|.\|$ como a norma Euclidiana, enquanto que em [13] $\|.\|$ se refere a uma norma arbitrária em $\mathbb{R}^n$. Além disso em [13] também é usada a notação $|C^+(x)|$, onde $|.|$ é uma norma monótona em $\mathbb{R}^n$, $C_j^+(x) = \max\{C_j(x), 0\}$ e $C^+(x) = (C_1^+(x), C_2^+(x), \ldots, C_m^+(x))$. Uma vez que C' e ∇f satisfazem a condição de Lipschitz podemos usar o seguinte lema:

Lema 2.4.1 *Seja $T : \mathbb{R}^n \to \mathbb{R}^m$ continuamente diferenciável no aberto convexo $D \subset \mathbb{R}^n$, $x \in D$, e seja T' Lipschitz contínua em x na vizinhança de D, usando uma norma vetorial, a norma matricial induzida e a constante γ. Então, para qualquer $x + p \in D$,*

$$\|T(x + p) - T(x) - T'(x)p\| \leq \frac{\gamma}{2} \|p\|^2 . \tag{2.39}$$

Dem.: ver [9] p. 75 Lema 4.1.12.
Considerando as hipóteses $A1$ e $A2$ e $p = y - x$ para o Lema 2.4.1, temos os seguintes resultados:

$$\|C(y) - C(x) - C'(x)(y-x)\| \leq \frac{L_1}{2}\|y-x\|^2 \tag{2.40}$$

e

$$|f(y) - f(x) - \langle \nabla f(x), (y-x)\rangle| \leq \frac{L_2}{2}\|y-x\|^2 \tag{2.41}$$

para todo $x, y \in \Omega$. Novamente podemos assumir, sem perda de generalidade, que (2.40) e (2.41) valem para diferentes normas com as mesmas constantes e que

$$\left|C_j(y) - C_j(x) - C_j'(x)(y-x)\right| \leq \frac{L_1}{2}\|y-x\|^2, \ \forall j = 1 \ldots m, \tag{2.42}$$

onde C_j é a j-ésima componente de C. O teorema seguinte é deduzido diretamente das hipóteses gerais. O teorema diz que a deterioração da viabilidade de $z^{k,i}$ em relação a viabilidade de y^k é limitada. Resumindo, provamos que somente uma deterioração de 2^a ordem da viabilidade pode ser esperada para um ponto teste $z \in \pi_k$ (2.5).

Teorema 2.4.1 *Existe $c_1 > 0$ (independente de k) tal que, sempre que $y^k \in \Omega$ é definido e $z \in \pi_k$, temos*

$$\|C(z)\| \leq \left\|C(y^k)\right\| + c_1\left\|z - y^k\right\|^2. \tag{2.43}$$

Dem.: se $z \in \pi_k$ temos $C'(y^k)(y^k - z) = 0$, então por (2.40) temos:

$$\left\| C(z) - C(y^k) - \underbrace{C'(y^k)(y^k - z)}_{=0} \right\| \leq \frac{L_1}{2}\left\|y^k - z\right\|^2,$$

$$\left\|C(z) - C(y^k)\right\| \leq \frac{L_1}{2}\left\|y^k - z\right\|^2.$$

Usando a desigualdade triangular:

$$\|C(z)\| = \left\|C(z) - C(y^k) + C(y^k)\right\| \leq \left\|C(z) - C(y^k)\right\| + \left\|C(y^k)\right\|,$$

subtraindo $\left\|C(y^k)\right\|$ de ambos os lados da desigualdade,

$$\|C(z)\| - \left\|C(y^k)\right\| \leq \left\|C(z) - C(y^k)\right\|,$$

então

$$\|C(z)\| - \left\|C(y^k)\right\| \leq \frac{L_1}{2}\left\|y^k - z\right\|^2,$$

$$\|C(z)\| \leq \left\|C(y^k)\right\| + \frac{L_1}{2}\left\|z - y^k\right\|^2.$$

Para $c_1 = \frac{L_1}{2}$, temos o resultado $\square$.

O decréscimo esperado da função objetivo f quando movemos de y^k para $z^{k,i}$ é analisado no próximo teorema.

Teorema 2.4.2 *Existe $c_2 > 0$ (independente de k) tal que, sempre que $y^k \in \Omega$ é definido e $z^{k,i}$ é computado no passo 5 do Algoritmo 2.1, temos que*

$$t_{dec} \geq \min\left\{1, \frac{0.09c}{L_2\eta_{max}}\right\} \tag{2.44}$$

e $f(z^{k,i}) \leq f(y^k) - c_2\left\|d_{tan}^k\right\|^2$.

Dem.: Por (2.41), desde que $y^k + d_{tan}^k \in \Omega$, temos que para todo $t \in [0,1]$.

$$\begin{aligned} f(y^k + td_{tan}^k) &\leq f(y^k) + t\left\langle \nabla f(y^k), d_{tan}^k\right\rangle + \frac{t^2L_2}{2}\left\|d_{tan}^k\right\| \\ &= f(y^k) + 0.1t\left\langle \nabla f(y^k), d_{tan}^k\right\rangle + 0.9t\left\langle \nabla f(y^k), d_{tan}^k\right\rangle \\ &\quad + \frac{t^2L_2}{2}\left\|d_{tan}^k\right\|^2. \end{aligned}$$

Se considerarmos (2.7), $\eta_{max} = \max_k \eta_k$ para $k = 1, \ldots$ e $c \in (0,1)$ temos

$$\left\langle \nabla f(y^k), d_{tan}^k\right\rangle \leq -\frac{1}{2\eta_{max}}\left\|d_{tan}^k\right\|^2 \leq \frac{-c}{2\eta_{max}}\left\|d_{tan}^k\right\|^2. \tag{2.45}$$

isto implica que:

$$\begin{aligned} f(y^k + td_{tan}^k) &\leq f(y^k) + 0.1t\left\langle \nabla f(y^k), d_{tan}^k\right\rangle - \frac{0.9ct\left\|d_{tan}^k\right\|^2}{2\eta_{max}} + \frac{t^2L_2}{2}\left\|d_{tan}^k\right\|^2 \\ &= f(y^k) + 0.1t\left\langle \nabla f(y^k), d_{tan}^k\right\rangle + \frac{t\left\|d_{tan}^k\right\|^2}{2}\left(tL_2 - \frac{0.9c}{\eta_{max}}\right). \end{aligned}$$

Portanto, se $t \leq \frac{0.9c}{L_2\eta_{max}}$, temos

$$f(y^k + td_{tan}^k) \leq f(y^k) + 0.1t\left\langle \nabla f(y^k), d_{tan}^k\right\rangle \ . \tag{2.46}$$

Temos que t_{dec} é o primeiro termo da sequência $\{t_j\}$ que satisfaz (2.46), tal que $t_1 = 1$ e $t_{j+1} \in [0.1t_j, 0.9t_j]$, portanto $t_{dec} \geq \min\left\{1, \frac{0.09c}{L_2\eta_{max}}\right\}$. Segue que:

$$f(y^k + t_{dec}d^k_{tan}) \leq f(y^k) + \min\left\{0.1, \frac{0.009c}{L_2\eta_{max}}\right\}\left\langle \nabla f(y^k), d^k_{tan}\right\rangle \ .$$

Novamente por (2.45) obtemos

$$f(y^k + t_{dec}d^k_{tan}) \leq f(y^k) - \min\left\{\frac{0.1c\left\|d^k_{tan}\right\|^2}{2\eta_{max}}, \frac{0.009c^2\left\|d^k_{tan}\right\|^2}{2L_2\eta^2_{max}}\right\}$$

Portanto:

$$f(z^{k,i}) = f(y^k + t_{dec}d^k_{tan}) \leq f(y^k) - c_2\left\|d^k_{tan}\right\|^2,$$

onde $c_2 = \min\left\{\frac{0.1c}{2\eta_{max}}, \frac{0.009c^2}{2L_2\eta^2_{max}}\right\}\square$.

O resultado a seguir mostra que o Algoritmo 2.1 está bem definido, isto é, para $t_{k,i}$ pequeno o suficiente, a desigualdade (2.32) é satisfeita e, então, o ponto teste $z^{k,i}$ é aceito como novo iterando.

Teorema 2.4.3 *O Algoritmo 2.1 está bem definido*

Dem.: Observe que de (2.30) e (2.31)

$$\begin{aligned}
Ared_{k,i} - 0.1Pred_{k,i} &= 0.9\theta_{k,i}\left[f(x^k) - f(z^{k,i})\right] + 0.9(1-\theta_{k,i})\left\|C(x^k)\right\| \\
&\quad -(1-\theta_{k,i})\left\|C(z^{k,i})\right\| + 0.1(1-\theta_{k,i})\left\|C(y^k)\right\| \\
&= 0.9\theta_{k,i}\left[f(x^k) - f(z^{k,i})\right] + 0.9(1-\theta_{k,i})\left\|C(x^k)\right\| - 0.9(1-\theta_{k,i})\left\|C(y^k)\right\| \\
&\quad +0.9(1-\theta_{k,i})\left\|C(y^k)\right\| - (1-\theta_{k,i})\left\|C(z^{k,i})\right\| + 0.1(1-\theta_{k,i})\left\|C(y^k)\right\| \\
&= 0.9\left\{\theta_{k,i}\left[f(x^k) - f(z^{k,i})\right] + (1-\theta_{k,i})\left[\left\|C(x^k)\right\| - \left\|C(y^k)\right\|\right]\right\} + \\
&\quad +(1-\theta_{k,i})\left[\left\|C(y^k)\right\| - \left\|C(z^{k,i})\right\|\right].
\end{aligned}$$

Logo temos:

$$Ared_{k,i} - 0.1Pred_{k,i} = 0.9Pred_{k,i} + (1-\theta_{k,i})\left[\left\|C(y^k)\right\| - \left\|C(z^{k,i})\right\|\right].$$

Portanto por (2.21), (2.22) e (2.29)

$$\begin{aligned}
Ared_{k,i} - 0.1Pred_{k,i} &\geq 0.45\left[\left\|C(x^k)\right\| - \left\|C(y^k)\right\|\right] - \left[\left\|C(y^k)\right\| - \left\|C(z^{k,i})\right\|\right] \\
&\geq 0.45(1-r)\left\|C(x^k)\right\| - \left[\left\|C(y^k)\right\| - \left\|C(z^{k,i})\right\|\right],
\end{aligned}$$

lembrando que $r \in (0,1)$ (2.3). Portanto, se $\left\|C(x^k)\right\| > 0$, desde que $\left\|y^k - z^{k,i}\right\| = \left\|t_{k,i} d^k_{tan}\right\|$ e $\|C(x)\|$ é contínua, temos que $Ared_{k,i} - 0.1Pred_{k,i} \geq 0$ se $\left\|t_{k,i} d^k_{tan}\right\| = t_{k,i}\left\|d^k_{tan}\right\|$ é pequeno o suficiente. Então, provamos que o algoritmo está bem definido se o ponto atual x^k não é viável.

Se x^k é viável, (2.4) implica que $y^k = x^k$ e $\left\|C(y^k)\right\| = 0$. Se $d^k_{tan} \neq 0$ temos $f(z^{k,i}) < f(y^k)$ para todo $i = 0, 1, 2, \ldots$. Portanto, neste caso, temos:

$$Ared_{k,i} - 0.1Pred_{k,i} = 0.9\theta_{k,i}\left[f(x^k) - f(z^{k,i})\right] - (1 - \theta_{k,i})\left\|C(z^{k,i})\right\|.$$

Então, pelos Teoremas 2.4.1 e 2.4.2, obtemos que:

$$\begin{aligned} Ared_{k,i} - 0.1Pred_{k,i} &\geq 0.9c_2\theta_{k,-1}\left\|d^k_{tan}\right\|^2 - c_1\left\|z^{k,i} - y^k\right\|^2 \\ &= 0.9c_2\theta_{k,-1}\left\|d^k_{tan}\right\|^2 - c_1 t^2_{k,i}\left\|d^k_{tan}\right\|^2 \\ &= (0.9c_2\theta_{k,-1} - c_1 t^2_{k,i})\left\|d^k_{tan}\right\|^2. \end{aligned} \tag{2.47}$$

Logo, (2.32) vale se

$$t_{k,i} \leq \sqrt{\frac{0.9c_2\theta_{k,-1}}{c_1}}.$$

Então, provamos que x^{k+1} é bem definido quando x^k é viável e $d^k_{tan} \neq 0$. □

2.4.2 Convergência a Pontos Viáveis

O próximo teorema é uma importante ferramenta para demonstração da convergência do Algoritmo 2.1. O teorema afirma que a redução real efetivamente atingida em cada iteração necessariamente tende a zero.

Teorema 2.4.4 *Suponha que o Algoritmo 2.1 gera uma seqüência infinita. Então de (2.18), temos que*

$$\lim_{k\to\infty} Ared_{k,iacc(k)} = \lim_{k\to\infty}\left[\psi(x^k, \theta_k) - \psi(x^{k+1}, \theta_k)\right] = 0.$$

Dem.: Por contradição, suponha que exista um conjunto infinito de índices $K_1 \subset \{0, 1, 2, \ldots\}$ e um número positivo $\gamma > 0$ tal que:

$$\psi(x^{k+1}, \theta_k) \leq \psi(x^k, \theta_k) - \gamma$$

para todo $k \in K_1$. Vamos escrever $\psi_k = \psi(x^k, \theta_k)$ para todo $k \in \{0, 1, 2\ldots\}$. Então, para todo $k \in \{0, 1, 2\ldots\}$, temos que:

$$\psi_{k+1} = \theta_{k+1} f(x^{k+1}) + (1 - \theta_{k+1})\left\|C(x^{k+1})\right\|$$

$$\begin{aligned}
&= \theta_{k+1}f(x^{k+1}) + (1-\theta_{k+1})\left\|C(x^{k+1})\right\| - \left[\theta_k f(x^{k+1}) + (1-\theta_k)\left\|C(x^{k+1})\right\|\right] \\
&\quad + \left[\theta_k f(x^{k+1}) + (1-\theta_k)\left\|C(x^{k+1})\right\|\right] \\
&= (\theta_{k+1}-\theta_k)f(x^{k+1}) + (\theta_k-\theta_{k+1})\left\|C(x^{k+1})\right\| + \theta_k f(x^{k+1}) + (1-\theta_k)\left\|C(x^{k+1})\right\| \\
&= (\theta_k-\theta_{k+1})(\left\|C(x^{k+1})\right\| - f(x^{k+1})) + \left[\theta_k f(x^k) + (1-\theta_k)\left\|C(x^k)\right\|\right] - \gamma_k.
\end{aligned}$$

Então,

$$\psi_{k+1} = (\theta_k-\theta_{k+1})(\left\|C(x^{k+1})\right\| - f(x^{k+1})) + \psi_k - \gamma_k \tag{2.48}$$

onde:

$$\gamma_k = \theta_k(f(x^k)-f(x^{k+1}))+(1-\theta_k)(\left\|C(x^k)-C(x^{k+1})\right\|) \geq 0, \text{ para todo } k \in \{0,1,2\ldots\}$$

e,

$$\gamma_k \geq \gamma > 0, \text{ para todo } k \in K_1.$$

Agora, pela definição de $\theta_{k,-1}$ no Algoritmo 2.1, ($\theta_{k,-1} = \min\{\theta_{-1},\ldots,\theta_{k-1}+\omega_k\}$) temos que:

$$\theta_k - \theta_{k+1} + \omega_k \geq 0 \text{ para todo } k \in \{0,1,2\ldots\} \tag{2.49}$$

Pela compacidade de Ω, e pela continuidade de $\left\|C(x^k)\right\|$ e $f(x^k)$, existe um limitante $c > 0$ tal que:

$$\left\|C(x^k)\right\| - \left|f(x^k)\right| \leq c,$$

para todo $k \in \{0,1,2\ldots\}$. Portanto por (2.48) e (2.49), temos que:

$$\begin{aligned}
\psi_{j+1} &= (\theta_j-\theta_{j+1}+\omega_j)(\left\|C(x^{j+1})\right\| - f(x^{j+1})) + \psi_j - \gamma_j - \omega_j(\left\|C(x^{j+1})\right\| - f(x^{j+1})) \\
&\leq (\theta_j-\theta_{j+1}+\omega_j)c + \psi_j - \gamma_j + \omega_j + c \\
&= (\theta_j-\theta_{j+1})c + \psi_j - \gamma_j + 2\omega_j c,
\end{aligned}$$

para todo $j = 0,1,2,\ldots,k-1$. Adicionando estas k desigualdades, obtemos:

$$\begin{aligned}
\psi_k &\leq \psi_0 + (\theta_0-\theta_k)c + \sum_{j=0}^{k-1} 2c\omega_j - \sum_{j=0}^{k-1}\gamma_j \\
&\leq \psi_0 + 2c + \sum_{j=0}^{k-1} 2c\omega_j - \sum_{j=0}^{k-1}\gamma_j,
\end{aligned} \tag{2.50}$$

para todo $k \geq 1$. Desde que a série $\sum_{j=0}^{k-1}\omega_j$ é convergente, e desde que γ_k é limitado acima de zero para todo $k \in K_1$ ($\gamma_k \geq \gamma > 0$), (2.50) implica que ψ_k é ilimitado inferiormente. Isto contradiz a compacidade de Ω. □

Uma conseqüência imediata do Teorema 2.4.4 é que se o Algoritmo 2.1 não para no Passo 1 e gera uma seqüência infinita. Temos que

$$\lim_{k\to\infty} \left\|C(x^k)\right\| = 0.$$

Isto significa que pontos arbitrariamente próximos à viabilidade são gerados.

Corolário 2.4.1 *Se o Algoritmo 2.1 não para no Passo 1 para todo $k = 1, 2, \ldots$, então* $\lim_{k\to\infty} \left\|C(x^k)\right\| = 0$. *(Em particular, cada ponto limite de $\left\{x^k\right\}$ é viável.)*

Dem.: Multiplicamos (2.23) por -1 e adicionamos $-\left\|C(x^k)\right\|$ em ambos os lados da inequação, obtendo:

$$\left\|C(x^k)\right\| \leq \frac{\left\|C(x^k)\right\| - \left\|C(y^k)\right\|}{1-r}.$$

Pelas inequações (2.32) e (2.29), temos

$$\left\|C(x^k)\right\| \leq \frac{2}{1-r} Pred_{k,i} \leq \frac{20}{1-r} Ared_{k,i}.$$

pela definição (2.19),

$$\left\|C(x^k)\right\| \leq \frac{20}{1-r}\left[\psi(x^k,\theta_k) - \psi(x^{k+1},\theta_k)\right].$$

Então, pelo Teorema 2.4.4, temos o resultado desejado. □

2.4.3 Convergência à Otimalidade

Na subseção anterior provamos que se o Algoritmo 2.1 não for interrompido no Passo 1, a viabilidade aproximada é alcançada para qualquer precisão desejada (ou adequada). Provaremos que, em tal caso, o indicador de otimalidade $\left\|d^k_{tan}\right\|$ não pode ser limitado inferiormente acima de zero. Na prática, isto implica que fixadas as tolerâncias para convergência, com relação à viabilidade e à otimalidade, que sejam arbitrariamente pequenas ($\varepsilon_1, \varepsilon_2 > 0$), o Algoritmo 2.1 encontrará um x^k tal que $\left\|C(x^k)\right\| \leq \varepsilon_1$ e $\left\|d^k_{tan}\right\| \leq \varepsilon_2$. Para provar este resultado, seguiremos a demonstração de [13], realizando uma prova por contradição, assumindo que $\left\|d^k_{tan}\right\|$ é limitado acima de zero para k suficientemente grande. Desta hipótese deduziremos os resultados intermediários que, finalmente, nos levarão à contradição.

Hipótese C: *O Algoritmo 2.1 gera uma seqüência infinita $\left\{x^k\right\}$ e existe $\varepsilon > 0$ e $k_0 \in \{0, 1, 2, \ldots\}$ tal que $\left\|d^k_{tan}\right\| \geq \varepsilon$ para todo $k \geq k_0$.*

Lema 2.4.2 *Suponha que a Hipótese C seja válida. Então, existe $c_3 > 0$ (independente de k) tal que $f(y^k) - f(z^{k,i}) \geq c_3$, para todo $k \geq k_0$ e $i = 0, 1, 2, \ldots, iacc(k)$.*

Dem.: do Teorema 2.4.2 temos:

$$f(y^k) - f(z^{k,i}) \geq c_2 \left\| d^k_{tan} \right\|,$$

para $c_3 = c_2 \left\| d^k_{tan} \right\|$, temos:

$$f(y^k) - f(z^{k,i}) \geq c_3. \ \square$$

Lema 2.4.3 *Suponha que a Hipótese C seja válida. Então, existe $\varepsilon_1 > 0$, independente de k e i, tal que $\left\| C(x^k) \right\| \leq \varepsilon_1$ implica que $\theta_{k,i} = \theta_{k,i-1}$.*

Dem.: Desde que $\theta_{k,i} \leq 1$, vamos definir $Pred(1) \equiv Pred_{k,i}$ com $\theta_{k,i} \equiv 1$. De (2.31)

$$Pred(1) = f(x^k) - f(z^{k,i}) \geq f(y^k) - f(z^{k,i}) - \left| f(x^k) - f(y^k) \right| \geq f(y^k) - f(z^{k,i}) - M \left\| y^k - x^k \right\|$$

onde M é uma constante que depende somente de $\|\nabla f(x)\|$ em Ω. Portanto, por (2.24) e pelo Lema 2.4.1,

$$Pred(1) - 0.5 \left\| C(x^k) \right\| \geq c_3 - (M\beta + 0.5) \left\| C(x^k) \right\|.$$

Defina $\varepsilon_1 = 2c_3/(2M\beta + 1)$. Se $\left\| C(x^k) \right\| \leq \varepsilon_1$ então $Pred(1) - 0.5 \left\| C(x^k) \right\| \geq 0$. Isto implica que a condição (2.29) é valida para qualquer valor de θ_k no intervalo $[0, 1]$. Em particular, $\theta_{k,i}$ satisfaz (2.29), e a demonstração está completa. $\square$

No próximo lema, provamos que, sob a Hipótese C, os parâmetros de penalidade $\{\theta_k\}$ são limitados acima de zero. Devemos lembrar que esta é uma propriedade de sequências que satisfaz a Hipótese C (que, oportunamente, será provada como não existente) e nem todas as sequências são efetivamente geradas pelo algoritmo.

Lema 2.4.4 *Suponha que a Hipótese C seja válida. Então, existe $\bar{\theta} > 0$ e $K_1 \in \{0, 1, 2, \ldots\}$ tal que $\theta_k \geq \bar{\theta}$ para todo $k \geq K_1$.*

Dem.: Primeiro mostraremos que, se $\left\| C(x^k) \right\|$ é suficientemente pequeno, um passo $t_{k,i} \left\| d^k_{tan} \right\|$ que satisfaz:

$$\left\| C(x^k) \right\| \geq 0.5 t_{k,i} \left\| d^k_{tan} \right\| \geq 0.5 t_{k,i} \varepsilon \tag{2.51}$$

é necessariamente aceito, com $\varepsilon > 0$ da Hipótese C. De fato, assuma (2.51) como válida. Então, por (2.29) e (2.23):

$$Pred_{k,i} \geq 0.5 \left[\left\| C(x^k) \right\| - \left\| C(y^k) \right\| \right] \geq 0.5(1 - r) \left\| C(x^k) \right\| \geq 0.25(1 - r) \varepsilon t_{k,i}$$

e então,

$$t_{k,i} \leq \frac{4}{(1-r)\varepsilon} Pred_{k,i}. \tag{2.52}$$

Por (2.30) e (2.31) e o Teorema 2.4.1:

$$\begin{aligned} Ared_{k,i} &= Pred_{k,i} + (1-\theta_{k,i}) \left[\left\| C(y^k) \right\| - \left\| C(z^{k,i}) \right\| \right] \\ &\geq Pred_{k,i} - (1-\theta_{k,i}) c_1 \left\| z^{k,i} - y^k \right\|^2 \\ &\geq Pred_{k,i} - c_1 \left\| z^{k,i} - y^k \right\|^2. \end{aligned} \tag{2.53}$$

Portanto, pela Hipótese C e (2.52), temos:

$$Ared_{k,i} \geq Pred_{k,i} - c_1 t_{k,i}^2 \varepsilon^2 \geq Pred_{k,i} - c_1 t_{k,i} \varepsilon^2 \frac{4}{(1-r)\varepsilon} Pred_{k,i} = Pred_{k,i} \left(1 - \frac{4 c_1 \varepsilon t_{k,i}}{1-r} \right).$$

Mas por (2.51), $-t_{k,i} \geq \frac{-2\left\|C(x^k)\right\|}{\varepsilon}$, e então, $Ared_{k,i} \geq \left(1 - \frac{8 c_1 \left\|C(x^k)\right\|}{1-r} \right) Pred_{k,i}$. Portanto, se (2.51) é válido e $\left\|C(x^k)\right\| \leq 0.9 \frac{(1-r)}{8c_1}$, então a condição (2.32) é verificada e o ponto teste $z^{k,i}$ é necessariamente aceito. Vamos definir:

$$\varepsilon_2 = \min \left\{ \varepsilon_1, \frac{0.9(1-r)}{8c_1}, \varepsilon t_{min} \right\},$$

onde ε_1 é definido no Lema 2.4.2 e,

$$t_{min} = \min \left\{ 1, \frac{0.09c}{L_2 \eta_{max}} \right\},$$

vem da relação (2.44) do Teorema 2.4.2. Pelo Corolário 2.4.1 (Teorema 2.4.4) existe $k_1 \geq k_0$ tal que $\left\|C(x^k)\right\| \leq \varepsilon_2$ para todo $k \geq k_1$. Portanto $\left\|C(x^k)\right\| \leq \varepsilon t_{min}$ e $t_{min} \geq \left\|C(x^k)\right\| / \varepsilon$, para todo $k \geq k_1$.

Isto implica que $t_{k,i} < \left\|C(x^k)\right\| / \varepsilon$ não pode corresponder a $i = 0$, então é precedido por $t_{k,i-1}$ que necessariamente verifica $t_{k,i-1} < 2 \left\|C(x^k)\right\| \varepsilon$, (admitindo que estamos fazendo bissecção para realizar o Passo 5 do Algoritmo 2.1) e por razões prévias, o ponto $z^{k,i}$ é aceito para todo $k \geq k_1$. Portanto, $t_{k,i} \geq \left\|C(x^k)\right\| / \varepsilon$ para todo $k \geq k_1$ e $i = 0, 1, \ldots, iacc(k)$. Então, pelo Lema 2.4.2, o parâmetro de penalidade $\theta_{k,i}$ nunca decresce para $k > k_1$, o que implica no resultado desejado.

Finalmente, provaremos que a Hipótese C não pode ser válida.

Teorema 2.4.5 *Seja $\{x^k\}$ uma seqüência infinita gerada pelo Algoritmo 2.1. Então, existe K, um subconjunto infinito de $\{0,1,2,\ldots\}$, tal que:*

$$\lim_{k\in K} \left\|d^k_{tan}\right\| = 0 \tag{2.54}$$

Dem.: Suponha que a tese do teorema não é verdadeira. Então existe $k_0 \in \{0,1,2,\ldots\}$ e $\varepsilon > 0$ tal que a Hipótese C é válida .

Como no Teorema 2.4.3, observe que, por (2.23) e o Teorema 2.4.1,

$$Ared_{k,i} \geq 0.9\theta_{k,i}\left[f(y^k) - f(z^{k,i})\right] + 0.9\theta_{k,i}\left[f(x^k) - f(y^k)\right] - (1-r)\left\|C(x^k)\right\| - c_1 t^2_{k,i}\left\|d^k_{tan}\right\|^2.$$

Além disso, usando expansão em série de Taylor e (2.24),

$$\begin{aligned}
\left|f(x^k) - f(y^k)\right| &= \left|\nabla f(x^k)^T(y^k - x^k) + \mathcal{O}(\left\|y^k - x^k\right\|)\right| \\
&\leq \left\|\nabla f(x^k)\right\|\left\|y^k - x^k\right\| + \mathcal{O}(\left\|y^k - x^k\right\|) \\
&\leq M_1\left\|y^k - x^k\right\| \\
&\leq M_1\beta\left\|C(x^k)\right\|, \qquad (2.55)
\end{aligned}$$

onde M_1 é uma norma constante que também depende do valor de $\|\nabla f(x)\|$ em Ω. Agora pelo Lema 3, existe $\bar{\theta} > 0$ e $k_1 \geq k_0$ tal que $\theta_{k,i}$ para todo $k \geq k_1$ e então

$$0.9\theta_{k,i}(f(x^k) - f(y^k)) \geq -0.9\bar{\theta}M_1\beta\left\|C(x^k)\right\| \equiv -M_2\left\|C(x^k)\right\|.$$

Então pelo Lema 2.4.1, temos

$$Ared_{k,i} - 0.1Pred_{k,i} \geq 0.9\bar{\theta}c_3 - M_2\left\|C(x^k)\right\| - c_1 t^2_{k,i}\varepsilon^2$$

para todo $k \geq k_1$ e $i = 0,1,\ldots,iacc(k)$.

Vamos definir $\bar{t} = \sqrt{t0.45\bar{\theta}c_3/(\varepsilon^2 c_1)}$. Se $t_{k,i} \leq \bar{t}$, temos que $c_1\varepsilon^2 t^2_{k,i} \leq 0.45\bar{\theta}c_3$, e então:

$$Ared_{k,i} - 0.1Pred_{k,i} \geq 0.45\bar{\theta}c_3 - M_2\left\|C(x^k)\right\| \tag{2.56}$$

para todo $k \geq k_1$ e $i = 0,1,\ldots,iacc(k)$. Seja $k_2 \geq k_1$ tal que:

$$M_2\left\|C(x^k)\right\| \leq 0.45\bar{\theta}c_3, \tag{2.57}$$

para todo $k \geq k_2$. Por (2.56) e (2.57) temos que, para todo $k \geq k_2$, se $\{i \in 0,1,2,\ldots\}$ corresponde ao primeiro $t_{k,i} \leq \bar{t}$, então:

$$Ared_{k,i} - 0.1Pred_{k,i} \geq 0.$$

Isto significa que $t_{k,i} \geq \bar{t}/2$ deve ser aceito, uma vez que estejamos usando bipartição para realizar o Passo 5 do Algoritmo 2.1. Portanto, $t_{k,iacck} \geq \bar{t}/2$ para todo $k \geq k_2$.

Então se $k \geq k_2$ temos, pelo Lema 2.4.2, Lema 2.4.3, (2.23) e (2.24), que

$$\begin{aligned} Pred_{k,iacck} &= \theta_{k,iacc(k)} \left[f(x^k) - f(z^{k,i})\right] + (1 - \theta_{k,iacc(k)}) \left\|C(x^k) - C(y^k)\right\| \\ &= \theta_{k,iacc(k)} \left[f(y^k) - f(z^{k,i})\right] + \theta_{k,iacc(k)} \left[f(x^k) - f(y^k)\right] \\ &\quad +(1 - \theta_{k,i}) \left[\left\|C(x^k)\right\| - \left\|C(y^k)\right\|\right] \\ &\geq \bar{\theta} \left[f(y^k) - f(z^{k,i})\right] - \left|f(x^k) - f(y^k)\right| - \left\|C(x^k)\right\| \\ &\geq \bar{\theta} c_3 - M_3 \left\|C(x^k)\right\|, \end{aligned} \tag{2.58}$$

para todo $k \geq k_2$, onde M_3 é uma constante que depende do valor de $\|\nabla f(x)\|$ em Ω. Seja, $k_3 \geq k_2$ tal que $M_3 \left\|C(x^k)\right\| \leq 0.5\bar{\theta} c_3$ para todo $k \geq k_3$. Por (2.58), $Pred_{k,iacck}$ é limitado acima de zero para todo $k \geq k_3$. Por (2.32), temos que $Ared_{k,iacc(k)}$ também é limitado acima de zero para todo $k \geq k_3$. Claramente, isto contradiz o Teorema 2.4.4. Isto significa que a Hipótese C não pode ser verdadeira , e o resultado desejado está demonstrado. □

Capítulo 3

Algoritmo para Sistemas Não Lineares Canalizados

3.1 Introdução

No capítulo anterior descrevemos o método de Restauração Inexata. Vimos que a etapa de restauração (Passo 1 do Algoritmo 2.1) requer um procedimento para obter um ponto y^k que satisfaça as condições:

$$\begin{aligned} \left\|C(y^k)\right\| &\leq r \left\|C(x^k)\right\|, \\ \left\|y^k - x^k\right\| &\leq \beta \left\|C(x^k)\right\|. \end{aligned} \tag{3.1}$$

Lembrando que $r \in (0,1)$, $\beta > 0$ e que y^k e x^k pertencem a Ω dado em (2.2), ou seja, temos restrições de canalização assim como o sistema:

$$C(x) = 0, \ x \in \Omega, \tag{3.2}$$

onde $C : \mathbb{R}^n \to \mathbb{R}^m$ é uma função continuamente diferenciável. Para resolver um sistema não linear da forma de (3.2), uma estratégia é associar a este sistema o problema de minimização onde a função objetivo é definida por $F(x) = \frac{1}{2}\|C(x)\|$.

Para resolver este problema de minimização usamos a estratégia de região de confiança que, a cada iteração, aproxima o problema original por um subproblema, onde é feita a aproximação quadrática de $F(x)$. A solução deste subproblema define um passo p restrito a uma região de confiança $\|p\| < \Delta$. Porém, no caso do sistema (3.2), Ω define restrições de canalização e por isso nem todas as soluções de $C(x) = 0$ serão admissíveis. Para abordar este problema, Coleman e Li, [7], introduziram uma matriz de escala D à região de confiança, definindo uma nova região elíptica $\|Dp\| < \Delta$. A matriz

D é definida a partir das restrições de canalização e do ponto corrente do algoritmo de região de confiança. A finalidade desta estratégia é permitir passos maiores dentro da região viável Ω. Em [3], Bellavia, Maconi e Morini apresentam uma versão do algoritmo de Coleman e Li [7] para resolver sistemas não lineares quadrados ($m = n$).

No trabalho de J. B. Francisco e J. M. Martínez [11], este algoritmo é proposto para realizar a fase de restauração do método de Restauração Inexata e é testado numericamente.

Neste capítulo desenvolvemos a teoria deste método e apresentamos testes numéricos similares aos de [11]. Nesta parte do trabalho definimos x como a variável do sistema não linear (3.2) que corresponde à variável y da fase de restauração do método de Restauração Inexata.

3.2 Descrição do Método de Região de Confiança

O método proposto em [11] gera pontos estritamente viáveis com relação a Ω para resolver o sistema (3.2).

Assim sendo, podemos chamá-lo de um método de pontos interiores, que utiliza a estratégia de região de confiança com uma matriz de escala especial D para determinar estes pontos.

Considerando o problema de resolver o sistema (3.2), e usando a função de mérito quadrática, temos o problema equivalente:

$$\min \ F(x) = \frac{1}{2} \|C(x)\|^2 , x \in \Omega. \tag{3.3}$$

A função de mérito $F(x) = \frac{1}{2} \|C(x)\|^2$ é amplamente usada em métodos de região de confiança, assume valores não negativos e é continuamente diferenciável desde que C seja continuamente diferenciável. Por isso (3.3) é uma boa abordagem para resolver (3.2) por meio de métodos de otimização. Porém não podemos garantir que um minimizador local de F será solução do sistema (3.2). Usando a regra da cadeia em F, o gradiente da função objetivo é:

$$\nabla F(x) = C'(x)^T C(x), \tag{3.4}$$

onde $C'(x)$ é matriz Jacobiana de $C(x)$. Para um minimizador local x^* de F temos $\nabla F(x^*) = 0$, o que, pela equação (3.4), implica que $C(x) = 0$ somente se o posto de $C'(x)$ for completo. Por esta razão assumimos a hipótese de que a matriz Jacobiana $C'(x)$ tem posto completo.

Para o cálculo da matriz Hessiana, podemos aplicar a regra da cadeia novamente para obter a expressão:

$$\nabla^2 F(x) = C'(x)^T C'(x) + \sum_{j=1}^{m} C_j(x) \nabla^2 C_j(x). \tag{3.5}$$

Dado x, a aproximação linear de $C(x+p)$ em torno de x é $C(x+p) \approx C'(x)p + C(x)$. Calculando o quadrado da norma Euclidiana desta aproximação linear, temos a aproximação quadrática $m(x,p)$ para $F(x+p)$:

$$\begin{aligned} m(x,p) &= \frac{1}{2} \|C'(x)p + C(x)\|^2 \\ &= \frac{1}{2} \|C(x)\|^2 + C(x)^T C'(x)p + \frac{1}{2} p^T C'(x)^T C'(x) p^T \\ &= F(x) + \nabla F(x)^T p + \frac{1}{2} p^T B(x) p^T, \end{aligned} \tag{3.6}$$

Note que $B(x) = C'(x)^T C'(x)$ é uma aproximação para $\nabla^2 F(x)$ quando x está próximo da solução de $C(x) = 0$, ou seja, quando estamos próximos da solução, $m(x,p)$ torna-se uma aproximação por série de Taylor de segunda ordem para $F(x)$. Esta aproximação é interessante por que B é simétrica e definida positiva, desde que a hipótese de que $C'(x)$ tenha posto completo seja satisfeita.

Podemos aplicar a estratégia de região de confiança para encontrar a solução de (3.3). Este método consiste em encontrar uma passo p^k tal que $F(x^k + p^k) < F(x^k)$, resolvendo um subproblema de otimização,

$$\min_p \left\{ m_k(p) : \|D_k p\| \leq \Delta \right\}, \tag{3.7}$$

onde $m_k(p) \equiv m(x^k, p)$ e D_k é uma matriz de escala definida a cada iteração do método, diagonal e definida positiva. O usual é escolher $\|p\| \leq \Delta$ como região de confiança, mas usamos $\|D_k p\| \leq \Delta$ para evitar problemas com o escalamento. Em otimização, um problema é chamado de mal escalado se uma pequena variação em x em uma certa direção resulta em grandes variações no valor da função f relativamente às variações em x em outras direções. Um exemplo simples é dado pela função $f(x) = 10^9 x_1^2 + x_2^2$, que é muito sensível a mudanças em x_1, mas não é tão sensível a mudanças em x_2. É fácil ver que a região de confiança esférica, $\|p\| \leq \Delta$, não é apropriada para o caso de um problema mal escalado. Construímos a aproximação m_k (3.7) para ter uma precisão razoável somente para curtas distâncias ao longo das direções altamente sensíveis, enquanto ela é adequada para distâncias maiores ao longo das direções menos

sensíveis. Desde que a forma da região de confiança deve ser tal que que nossa confiança no modelo é mais ou menos a mesma em todos os pontos na fronteira da região, somos levados naturalmente a considerar uma região de confiança elíptica na qual os eixos são menores nas direções sensíveis e maiores nas direções menos sensíveis.

Uma vez encontrado o passo p^k, devemos avaliar se o aceitaremos e como será alterado o raio da região de confiança. Para isso avaliamos o decréscimo da função F em relação ao decréscimo do modelo quadrático $m_k(p)$ (3.7), esperando que a seguinte condição se cumpra:

$$F(x^k) - F(x^k + p^k) \geq m_k(0) - m_k(p^k), \tag{3.8}$$

ou seja, queremos que o decréscimo real,

$$\bar{A}_{red}(x^k) = F(x^k) - F(x^k + p^k), \tag{3.9}$$

seja maior que o decréscimo previsto pelo modelo quadrático:

$$\bar{P}_{red}(x^k) = m_k(0) - m_k(p^k). \tag{3.10}$$

Nem sempre podemos garantir que o decréscimo real seja maior que o decréscimo previsto, por isso exigimos apenas que a redução real seja um fração da redução pretendida, ou seja, um critério de decréscimo mínimo no lugar de (3.8), e que é expresso na seguinte inequação:

$$\bar{A}_{red} \geq \beta_1 \bar{P}_{red}, \tag{3.11}$$

com $\beta_1 \in (0, 1)$. Definindo ρ_F^k:

$$\rho_F^k = \frac{\bar{A}_{red}(x^k)}{\bar{P}_{red}(x^k)}, \tag{3.12}$$

reescrevemos o critério de decréscimo mínimo e verificamos se este é satisfeito:

$$\rho_F^k \geq \beta_1. \tag{3.13}$$

Se o critério de decréscimo mínimo é satisfeito, aceitamos o passo p^k e o ponto da próxima iteração será $x^{k+1} = x^k + p^k$. Se além disto p^k satisfaz:

$$\rho_F^k \geq \beta_2. \tag{3.14}$$

com $0 < \beta_1 < \beta_2 < 1$, além de aceitar o passo p^k, aumentamos o valor de Δ, ou seja, expandimos o tamanho da região de confiança para permitir que a norma de p^k seja maior para atingir a solução do sistema mais rapidamente.

Mas se o critério de decréscimo mínimo não é satisfeito, reduzimos do raio da região de confiança. A Figura 3.1 resume a idéia do método de região de confiança comparando-o ao método de Newton com busca linear. Esta figura mostra a região de confiança ao redor de um ponto e as direções de descida que podem ser escolhidas: a direção de busca linear do método de Newton, que minimiza o modelo quadrático m_k e a direção p do método de região de confiança, obtida ao se resolver o subproblema de minimização (3.7). Observando a figura, vemos que utilizando o raio de região de confiança adequado e usando a direção p, nos aproximamos mais do ponto de mínimo local de f.

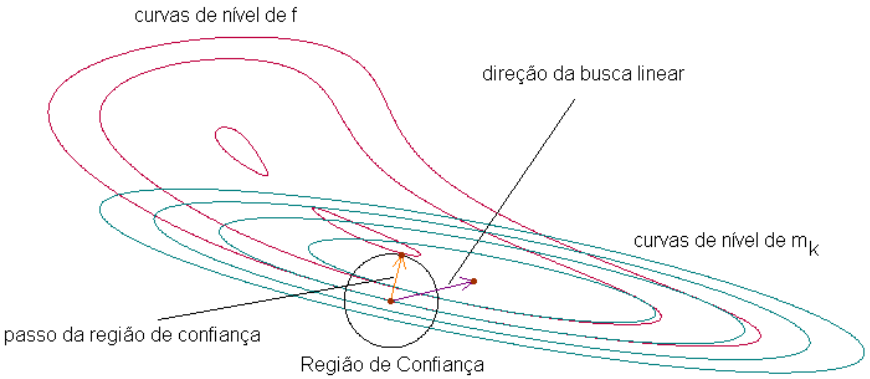

Figura 3.1: Exemplo: O passo da região de confiança e o passo de Newton

O Algoritmo 3.1 descreve o processo da estratégia de região de confiança.

Algoritmo 3.1 *Algoritmo de Região de Confiança:*

Dados $\bar{\Delta} > 0$, $\Delta_0 \in (0, \bar{\Delta})$, β_1 e $\beta_2 \in (0, 1)$ e $\delta_1 \in (0, 1)$ e $\delta_2 \in (1, 2]$;
Para $k = 0, 1, 2, \ldots$

1. Resolva o subproblema de otimização (3.7);
2. Avalie ρ_F^k de (3.14);
3. Se $\rho_F^k < \beta_1$

   3.1 $\Delta_{k+1} = \delta_1 \Delta_k$;

4. Senão

   4.1 Se $\rho_F^k > \beta_2$ e $\left\|p^k\right\| = \Delta_k$

   $$\Delta_{k+1} = \min(\delta_2 \Delta_k, \bar{\Delta});$$

   4.2 Senão

   $$\Delta_{k+1} = \Delta_k;$$

5. Se $\rho_F^k > \beta_1$

   5.1 $x^{k+1} = x^k + p^k$;

6. Senão

   6.1 $x^{k+1} = x^k$;

Fim

3.3 Matriz de Escala

Apresentamos a seguir a matriz afim-escala D definida a partir das restrições de canalização l e u, usada nos trabalhos de Coleman e Li [7] e de Bellavia, Maconi e Morini [3], para definir uma região elíptica na estratégia de região confiança, permitindo maiores passos em direção da solução do sistema não linear (3.2).

Podemos reescrever o problema (3.3) da seguinte maneira,

$$\begin{array}{lll} \min & F(x) & \\ s.a & g_1(x) = x_1 - l_1 & \geq 0 \\ & g_2(x) = x_2 - l_2 & \geq 0 \\ & \vdots & \\ & g_n(x) = x_n - l_n & \geq 0 \\ & g_{n+1}(x) = u_1 - x_1 & \geq 0 \\ & g_{n+2}(x) = u_2 - x_2 & \geq 0 \\ & \vdots & \\ & g_{2n}(x) = u_n - x_n & \geq 0. \end{array} \tag{3.15}$$

Se x^* é um minimizador local de (3.15) e portanto de (3.3), temos pelas condições de otimalidade Karush-Kuhn-Tucker (KKT), que são condições necessárias de primeira ordem, ver [22], as seguintes equações para $i = 1 \ldots 2n$,

$$\nabla_x L(x^*, \lambda^*) = \nabla F(x^*) - \sum_{i=1}^{2n} \lambda_i^* \nabla g_i(x^*) = 0, \tag{3.16}$$

$$g_i(x^*) \geq 0, \tag{3.17}$$

$$\lambda_i^* \geq 0, \tag{3.18}$$

e

$$\lambda_i^* g_i(x^*) = 0, \tag{3.19}$$

onde $\lambda^* \in \mathbb{R}^n$ é o vetor de multiplicadores de Lagrange.
Usando a notação $\nabla F = \nabla F(x)$ e $\nabla F^* = \nabla F(x^*)$, teremos:

$$\nabla g_i = e_i,$$

$$\nabla g_{n+i} = -e_i, \ i = 1 \ldots n$$

e,

$$\nabla F^* = \sum_{i=1}^{n} \lambda_i^* \nabla g_i(x^*) + \sum_{i=n+1}^{2n} \lambda_i^* \nabla g_i(x^*) = 0,$$

onde e_i é o vetor canônico,

$$e_i(j) = \begin{cases} 0 \text{ se } i \neq j \\ 1 \text{ se } i = j \end{cases},$$

com $j = 1 \ldots 2n$. Então cada componente de ∇F^* será,

$$\nabla F_i^* = \lambda_i^* - \lambda_{n+i}^*. \tag{3.20}$$

Vamos analisar as possibilidades:

1. se $l_i < x_i < u_i$, então $g_i(x^*) = x_i - l_i > 0$ e $g_{n+i}(x^*) = u_i - x_i > 0$, pela equação (3.19), $\lambda_i^* = \lambda_{n+i}^* = 0$ e por (3.20) temos $\nabla F_i^* = 0$;

2. se $g_{n+i}(x^*) = u_i - x_i = 0$ então $g_i(x^*) = x_i - l_i > 0$, pela equação (3.19), $\lambda_i^* = 0$ e $\lambda_{n+i}^* \geq 0$, por (3.20) temos $\nabla F_i^* \geq 0$;

3. se $g_i(x^*) = x_i - l_i = 0$ então $g_{n+i}(x^*) = x_i - l_i > 0$, pela equação (3.19), $\lambda_i^* = 0$ e $\lambda_{n+i}^* \geq 0$, por (3.20) temos $\nabla F_i^* \leq 0$;

em resumo, quando x^* é um ponto estacionário, obtemos:

$$\begin{cases} \nabla F_i^* = 0 & se\ l_i < x_i < u_i, \\ \nabla F_i^* \leq 0 & se\ x_i = u_i, \\ \nabla F_i^* \geq 0 & se\ x_i = l_i, \end{cases} \tag{3.21}$$

Para qualquer $s \in \mathbb{R}^n$, $diag(s)$ denota uma matriz diagonal $n \times n$ com o vetor s definindo as entradas da diagonal em sua ordem natural. Essas condições de primeira ordem podem ser reescritas de uma outra forma, para isto fazemos a seguinte definição:

Definição 3.3.1 *O vetor $v(x) \in \mathbb{R}^n$ é definido como se segue.*

Para cada componente $1 \leq i \leq n$,

$$\begin{array}{rll} (i) & se\ \nabla F_i < 0\ e\ u_i < \infty & \text{então } v_i = x_i - u_i; \\ (ii) & se\ \nabla F_i \geq 0\ e\ l_i > -\infty & \text{então } v_i = x_i - l_i; \\ (iii) & se\ \nabla F_i < 0\ e\ u_i = \infty & \text{então } v_i = -1; \\ (iv) & se\ \nabla F_i \geq 0\ e\ l_i = -\infty & \text{então } v_i = 1. \end{array} \tag{3.22}$$

Usando o vetor v, definimos a matriz de escala diagonal como

$$D(x) = \begin{bmatrix} |v_1(x)|^{-1/2} & & \\ & \ddots & \\ & & |v_n(x)|^{-1/2} \end{bmatrix}, \tag{3.23}$$

e podemos denotar a matriz $D(x)$ por

$$D(x) = diag(|v_1(x)|^{-1/2}, |v_2(x)|^{-1/2}, \ldots, |v_n(x)|^{-1/2}). \tag{3.24}$$

A matriz de escala define um região elíptica que permite passos maiores em direção à solução do problema, como mostra a Figura 3.2.

A matriz D, embora não definida na fronteira de Ω, pode ter a sua inversa

$$D(x)^{-1} = diag(|v_1(x)|^{1/2}, |v_2(x)|^{1/2}, \ldots, |v_n(x)|^{1/2}), \tag{3.25}$$

estendida continuamente até a fronteira, ou seja a matriz D está definida apenas no $int\Omega$. Assim redefinimos as condições necessárias de primeira ordem por um sistema diagonal.

Lema 3.3.1 *Seja $x \in \Omega$. Então $D(x)^{-1}\nabla F = 0$ se e somente se x é um ponto estacionário.*

Dem.: ver página 14 de [11] e referência [7].

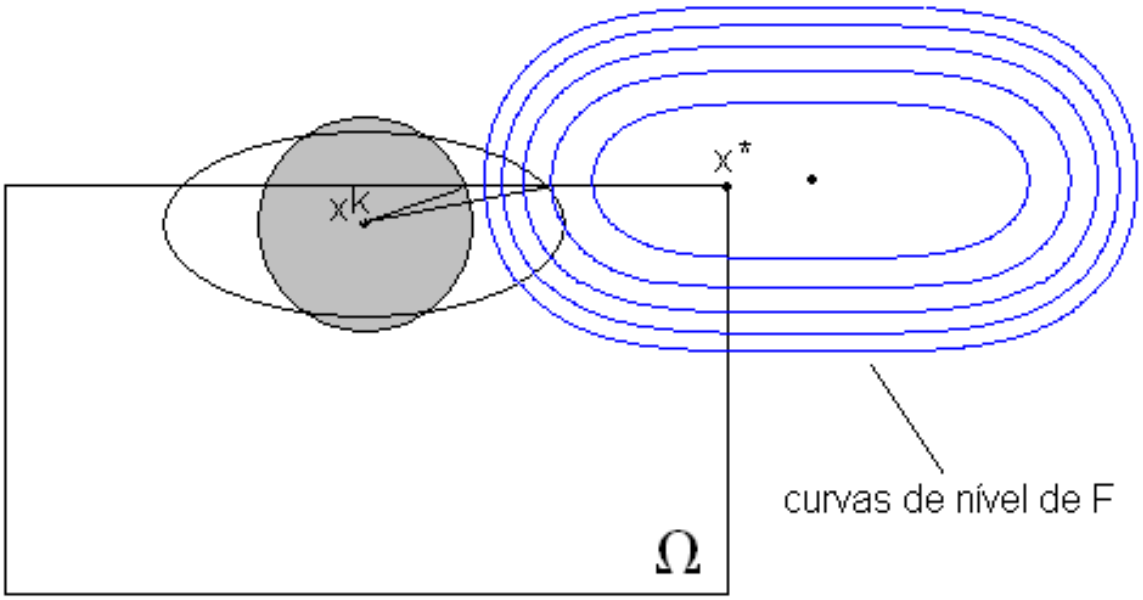

Figura 3.2: Matriz de Escala

3.4 Viabilidade

Devemos lembrar que queremos resolver o problema (3.3) que está restrito a Ω, então escolheremos o passo p^k de tal maneira que não viole as restrições de canalização. Uma vez determinada a solução p^k do subproblema (3.7), não temos garantia que, para um x^k viável, $x^{k+1} = x^k + p^k$ pertença a Ω. Por isso devemos escolher um parâmetro $\alpha > 0$ tal que $l \leq x^k + \alpha p^k \leq u$, logo, seguindo o desenvolvimento feito em [3], para $i = 1, 2, ..., n$, temos que:

$$l_i \leq x_i^k + \alpha p_i^k \leq u_i,$$

então:

$$\frac{l_i - x_i}{p_i^k} \leq \alpha \leq \frac{u_i - x_i}{p_i^k},$$

para $p_i^k \neq 0$.

O passo na direção de p^k deve ser o maior possível, mas sem violar a restrição de canalização, por isso definimos:

$$\gamma(p^k) = \begin{cases} \infty & \text{se } \Omega = \mathbb{R}^n, \\ \min_i \Gamma_i(p^k) & \text{se } \Omega \subset \mathbb{R}^n, \end{cases} \tag{3.26}$$

onde, para cada $i = 1, 2 \ldots, n$, $\Gamma_i(p^k)$ é dado por

$$\Gamma_i(p^k) = \begin{cases} \max\left\{\frac{l_i - x_i^k}{p_i^k}, \frac{u_i - x_i^k}{p_i^k}\right\} & \text{se } p_i^k \neq 0, \\ \infty & \text{se } p_i^k = 0. \end{cases} \tag{3.27}$$

Mas quando x^{k+1} pertence à fronteira de Ω, para pelo menos uma componente i de x, teremos $x_i = u_i$, ou $x_i = l_i$, com $i = 1, \ldots, n$. Desta maneira a componente v_i do vetor v (3.22) se anula e a matriz de escala D (3.24) não estará definida. Portanto não basta determinar iterações que pertençam a Ω, mas pertençam ao interior de $\Omega = \{x \in \mathbb{R}^n \mid l \leq x \leq u\}$, ou seja, devemos manter viabilidade estrita. Se $\gamma(p^k) > 1$ temos que cada componente de p^k é menor que as diferenças $x_i - u_i$ ou $x_i - l_i$, portanto $x^{k+1} = x^k + p^k \in int(\Omega)$, caso contrário, uma redução no passo ao longo de p^k será necessária para que a próxima iteração $x^{k+1} \in int(\Omega)$.

Sejam $\theta \in (0, 1)$ uma constante fixa, e $\zeta(p^k)$ dado por:

$$\zeta(p^k) = \begin{cases} 1 & \text{se } \gamma(p^k) > 1, \\ \max\left\{\theta, 1 - \left\|p^k\right\|\right\} \gamma(p^k) & \text{caso contrário} \end{cases} \tag{3.28}$$

e,

$$\alpha(p^k) = \zeta(p^k) p^k. \tag{3.29}$$

Então para assegurar que a nova iteração é estritamente viável com respeito à restrição de canalização, definimos:

$$x^{k+1} = x^k + \alpha(p^k). \tag{3.30}$$

3.5 Resolvendo o Subproblema de Minimização

Para pôr em prática o método de região de confiança precisamos resolver o subproblema (3.7). Para isto descrevemos duas estratégias para encontrar soluções aproximadas de (3.7). A primeira proposta é baseada no ponto de Cauchy, que consiste em encontrar o minimizador do modelo m_k ao longo da direção de máxima descida $-\nabla F(x^k)$, sujeita à região de confiança. A segunda é o método Dogleg, que é apropriado para caso onde $B(x^k)$ é definida positiva.

Deste ponto em diante escreveremos, quando necessário, $F(x^k) = F_k$ e $B(x^k) = B_k$, e qualquer outra função de x^k, pode ser representada de forma semelhante.

3.5.1 O Ponto de Cauchy

Os métodos de busca linear não exigem que se calcule o tamanho de passo ótimo para que sejam globalmente convergentes, isto é, convergência garantida a partir de qualquer aproximação inicial. Somente uma aproximação do tamanho de passo ótimo satisfazendo um critério de decréscimo suficiente, como o critério de Armijo, é necessária. Uma condição semelhante se aplica aos métodos de região de confiança. Embora em princípio estejamos procurando pela solução ótima do subproblema (3.7), é suficiente para convergência global propor uma solução aproximada p^k que esteja na região de confiança e que nos dê uma redução suficiente no modelo. A redução suficiente pode ser quantificada em termos do ponto de Cauchy, que denotamos por p_C^k e definimos através do seguinte procedimento mostrado em [22], página 69:

Algoritmo 3.2 *Cálculo do Ponto de Cauchy*

1. Encontre o vetor p_s^k que resolve uma aproximação linear de (3.7), isto é,

$$p_s^k = \arg\min_{p\in\mathbb{R}^n}(F_k + \nabla {F_k}^T p) \;\text{ sujeito a } \|D_k p\| \leq \Delta_k. \tag{3.31}$$

2. Calcule o escalar $\sigma^k > 0$ que minimiza $m_k(\sigma p_s^k)$ sujeito a região de confiança:

$$\sigma^k = \arg\min_{\sigma>0} m_k(\sigma p_s^k) \text{ sujeito a } \left\|\sigma D_k p_s^k\right\| \leq \Delta_k. \tag{3.32}$$

3. Defina $p_C^k = \sigma^k p_s^k$.

A solução de (3.31) pode ser obtida através da troca de variável $\tilde{p} = D_k p$. Assim reescrevemos o problema (3.31) da seguinte forma:

$$\tilde{p}_s^k = \arg\min_{\tilde{p}\in\mathbb{R}^n}(F(x_k) + \nabla {F_k}^T D_k^{-1}\tilde{p}) \;\text{ sujeito a } \|\tilde{p}\| \leq \Delta_k. \tag{3.33}$$

A direção de máxima descida com relação a $\tilde{p}$ é $-\tilde{g}_C^k$, onde $\tilde{g}_C^k = D_k^{-1}\nabla F_k$. Como se trata de uma aproximação linear, quanto mais caminharmos nesta direção menor será o valor da função. Mas estamos restritos a região de confiança, então devemos escolher um parâmetro real a tal que

$$\left\|a(-D_k^{-1}\nabla F_k)\right\| = \Delta_k.$$

Isolando a nesta equação, o minimizador será:

$$\tilde{p}_s^k = -\frac{\Delta_k}{\left\|D_k^{-1}\nabla F_k\right\|} D_k^{-1}\nabla F_k.$$

Da troca de variáveis, temos que: $p_s^k = D_k^{-1}\tilde{p_s^k}$, portanto:

$$p_s^k = -\frac{\Delta_k}{\left\|D_k^{-1}\nabla F_k\right\|} D_k^{-2}\nabla F_k. \tag{3.34}$$

Então, a direção de máxima descida em relação a p, ou seja, a direção de máxima descida considerando a matriz de escala é:

$$g_D^k = D_k^{-2}\nabla F_k. \tag{3.35}$$

Para obter o valor de σ^k explicitamente, vamos minimizar:

$$m_k(\sigma p_s^k) = F_k + \sigma {\nabla F_k}^T D_k^{-1} p_s^k + \frac{1}{2}\sigma^2 {p_s^k}^T B_k p_s^k, \tag{3.36}$$

com relação a σ. Para isso, consideramos os casos ${p_s^k}^T B_k p_s^k \leq 0$ e ${p_s^k}^T B_k p_s^k > 0$, separadamente.

Para o primeiro caso, a função $m_k(\sigma p_s^k)$ decresce monotonicamente com σ sempre que $p_s^k \neq 0$, então escolhemos simplesmente o maior valor que satisfaz o limite da região de confiança, ou seja, $\sigma^k = 1$.

Para o segundo caso, ${g_C^k}^T D_k^{-1} B_k D_k^{-1} g_C^k > 0$, $m_k(\sigma p_s^k)$ é uma função quadrática convexa em σ, cujo minimizador σ_m pode ser determinado derivando $m_k(\sigma p_s^k)$ com relação a σ e igualando a zero:

$$\nabla F_k D_k^{-1} p_s^k + \sigma_m^k p_s^k B_k p_s^k = 0.$$

Isolando σ_m nesta equação e substituindo o valor de p_s^k temos,

$$\sigma_m = \frac{\left\|D_k^{-1}\nabla F_k\right\|^3}{{\nabla F_k}^T D_k^{-2} B_k D_k^{-2}\nabla F_k}. \tag{3.37}$$

Tanto σ_m quanto o valor 1, que toca a fronteira da região de confiança, podem ser escolhidos como o novo valor de σ^k. Em resumo, temos:

$$p_C^k = -\sigma^k \frac{\Delta_k}{\left\|D_k^{-1}\nabla F_k\right\|} D_k^{-2}\nabla F_k, \tag{3.38}$$

onde:

$$\sigma^k = \begin{cases} 1 & \text{se } {\nabla F_k}^T D_k^{-2} B_k D_k^{-2}\nabla F_k \leq 0; \\ \min(\|D^{-1}\nabla F_k\|^3 / {\nabla F_k}^T D_k^{-2} B_k D_k^{-2}\nabla F_k, 1) & \text{caso contrário.} \end{cases} \tag{3.39}$$

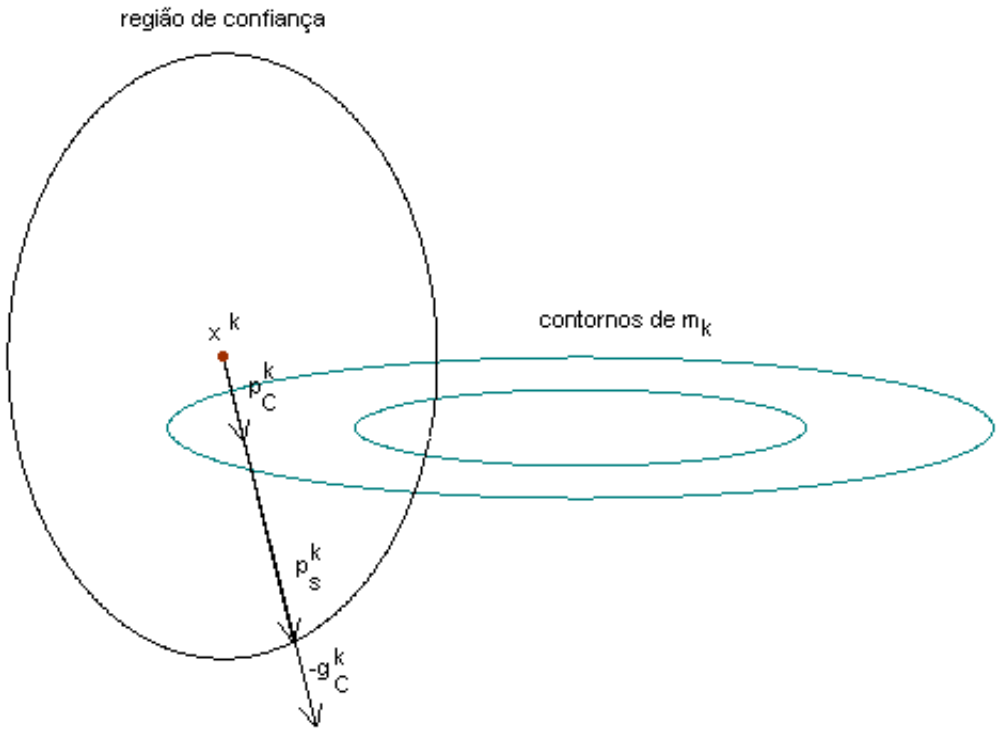

Figura 3.3: O Ponto de Cauchy

A Figura 3.3 mostra o Ponto de Cauchy para um subproblema no qual B_k é definida positiva. Neste exemplo, p_C^k está estritamente dentro da região de confiança.

Desde que $\frac{\Delta_k}{\left\|D_k^{-1}\nabla F_k\right\|}$ em (3.34) é uma constante dentro do algoritmo do passo de Cauchy, podemos reescrever:

$$p_C^k = -\tau^k D_k^{-2}\nabla F_k,$$

onde

$$\tau^k = \sigma^k \frac{\Delta_k}{\left\|D_k^{-1}\nabla F_k\right\|},$$

além disso, como $B_k = {C'_k}^T C'_k$, temos:

$$\nabla {F_k}^T D_k^{-2} B_k D_k^{-2}\nabla F_k = \left\|C'_k D_k^{-2}\nabla F_k\right\|^2 \geq 0,$$

então,

$$\tau^k = \min\left\{ \frac{\|D^{-1}\nabla F_k\|^2}{\left\|C_k' D_k^{-2}\nabla F_k\right\|}, \frac{\Delta_k}{\left\|D_k^{-1}\nabla F_k\right\|} \right\}, \tag{3.40}$$

Se o ponto de Cauchy fornece uma redução suficiente no modelo m_k da função para levar à convergência global, e se o custo computacional é pequeno, por que deveríamos procurar por uma solução aproximada melhor para (3.7)? A razão é que sempre que tomamos o passo de Cauchy, estamos implementando o método de máxima descida com uma escolha particular de tamanho de passo. E como pode ser visto em [22], Capítulo 3, o método de máxima descida tem um desempenho ruim mesmo se o tamanho do passo ótimo é usado em cada iteração. Por esta razão apresentaremos na próxima subseção um método para resolver o subproblema de otimização (3.7), que combina o passo de Cauchy e a direção de Newton de norma-2 mínima.

3.5.2 Método Dogleg

Consideramos agora um método para encontrar soluções aproximadas para o problema (3.7) que tem as características descritas anteriormente. Neste tópico estamos interessados descrever uma iteração deste método, então descartamos o índice k das quantidades Δ_k, p^k, D_k, $\tilde{m}_k$ e afins para simplificar a notação. Com esta simplificação, mais a transformação $\tilde{p} = Dp$ e $g = \nabla F$ redefinimos o subproblema de região de confiança (3.7) como:

$$\min_{\tilde{p}\in\mathbb{R}^n} \ \tilde{m}(\tilde{p}) = g^T D^{-1}\tilde{p} + \tfrac{1}{2}\tilde{p}^T D^{-1} B D^{-1}\tilde{p} \quad \text{s. a } \|\tilde{p}\| \leq \Delta. \tag{3.41}$$

Observação: retiramos o valor $F \equiv F_k$ do modelo quadrático $\tilde{m}(\tilde{p})$ presente no problema (3.41), pois se trata de uma constante que não altera a solução deste problema.
Denotamos a solução de (3.41) por $p^*(\Delta)$, para enfatizar a dependência de Δ.

Iniciamos examinando o efeito do raio da região de confiança Δ na solução $p^*(\Delta)$ do subproblema (3.41). Lembrando que estamos resolvendo o sistema $C(x) = 0$ com $x \in \Omega$, um passo válido para resolver o problema seria similar ao passo de Newton para resolver sistemas não lineares quadrados. Seguindo o desenvolvimento feito em [22], (páginas 69 a 73 e 94 a 96), usamos a aproximação linear:

$$C(x^k + p^k) \approx C_k + C_k'(x)p^k,$$

em seguida encontramos o vetor de p^k, resolvendo o sistema:

$$C_k' p^k = -C_k, \tag{3.42}$$

mas como $C_k' \in (\mathbb{R}^{m\times n})$ com $m < n$, o sistema acima admite infinitas soluções.

A solução de norma-2 mínima será a solução do seguinte problema:

$$\begin{array}{ll} \min & \frac{1}{2}\|p\|^2 \\ \text{s. a} & C'_k p^k = -C_k. \end{array} \tag{3.43}$$

Aplicando as condições KKT, [22], (Capítulo 12), ao problema verificamos primeiro o seu Lagrangiano:

$$L(x,\lambda) = \frac{1}{2}p^T p + \lambda^T (C'_k p^k + C_k),$$

e a condição de otimalidade será:

$$\nabla_x L(x,\lambda) = p + {C'_k}^T \lambda = 0.$$

Desta condição extraímos $p = -{C'_k}^T \lambda$. Substituindo este valor no sistema $C'_k p = -C_k$ temos:

$$C'_k {C'}^T_k \lambda \equiv C_k,$$

como a matriz $C'_k \in \mathbb{R}^{m\times n}$ com $m \leq n$ e exigindo $posto(C'_k) = m$, temos que $C'_k {C'_k}^T \in \mathbb{R}^{m\times m}$ é inversível, daí podemos escrever:

$$\lambda \equiv (C'_k {C'_k}^T)^{-1} C_k.$$

Portanto a solução de norma-2 mínima, a qual chamaremos de $p^k_{\mathcal{N}}$, será:

$$p^k_{\mathcal{N}} = -{C'_k}^{\dagger} C_k, \tag{3.44}$$

onde:

$${C'_k}^{\dagger} = {C'_k}^T (C'_k {C'_k}^T)^{-1},$$

é a chamada matriz pseudo-inversa de C'_k. Se $\left\|D_k p^k_{\mathcal{N}}\right\| \leq \Delta$ então $p^k_{\mathcal{N}}$ é solução de (3.41). Para mostrar isto vamos calcular o vetor gradiente de $\tilde{m}_k(\tilde{p})$.

$$\nabla \tilde{m}_k(\tilde{p}) = B_k D_k^{-1} \tilde{p} + D_k^{-1} g_k, \tag{3.45}$$

$B_k = {C'}^T_k C'_k$ e D_k^{-1} são simétricas, então $B_k D_k^{-1} = D_k^{-1} B_k$ e $g_k = \nabla F_k = {C'}^T_k C_k$. Substituindo estes termos em (3.45), temos:

$$\nabla \tilde{m}_k(\tilde{p}) = D_k^{-1}({C'}^T_k C'_k \tilde{p} + {C'}^T_k C_k).$$

Para $\tilde{p} = p^k_N$, temos ${C'}_k p^k_N = -C_k$, então

$$\nabla \tilde{m}_k(p^k_N) = D_k^{-1}(-{C'}^T_k C_k + {C'}^T_k C_k) = 0.$$

Como B_k é definida positiva , temos que o minimizador de (3.41) é

$$\tilde{p}^*(\Delta) = p_{\mathcal{N}}^k, \text{ quando } \left\|p_{\mathcal{N}}^k\right\| \leq \Delta. \tag{3.46}$$

Quando Δ é pequeno, a restrição $\|\tilde{p}\| \leq \Delta$ assegura que o termo quadrático em $\tilde{m}$ tem um pequeno efeito na solução de (3.41). A solução $\tilde{p}^*(\Delta)$ é aproximadamente a mesma que obteríamos pela minimização da função linear $F + g^T D^{-1}\tilde{p}$ sobre $\|\tilde{p}\| \leq \Delta$, isto é,

$$\tilde{p}^*(\Delta) \approx -\Delta \frac{D^{-1}g}{\|D^{-1}g\|}, \quad \text{quando } \Delta \text{ é pequeno.} \tag{3.47}$$

Para valores intermediários de Δ, a solução $\tilde{p}^*(\Delta)$ tipicamente segue uma trajetória curva, [22], Seção 4.1. O método Dogleg encontra uma solução aproximada trocando a trajetória curva por um caminho consistindo de dois segmentos de reta. O primeiro segmento sai da origem para o minimizador irrestrito ao longo da direção de máxima descida,

$$p^U = -\frac{\|D^{-1}g\|^2}{g^T D^{-2} B D^{-2} g} D^{-2} g, \tag{3.48}$$

que é um dos possíveis valores para a direção de Cauchy, enquanto o segundo segmento de reta sai de p^U para $p_{\mathcal{N}}^k$. Formalmente, denotamos esta trajetória por $\tilde{p}(\tau)$ para $\tau \in [0, 2]$, onde

$$\tilde{p}(\tau) = \begin{cases} \tau p^U, & 0 \leq \tau \leq 1 \\ p^U + (\tau - 1)(p_{\mathcal{N}}^k - p^U), & 1 \leq \tau \leq 2 \end{cases} \tag{3.49}$$

O método Dogleg escolhe o valor de p que minimiza o modelo $\tilde{m}$ ao longo deste caminho, sujeito aos limites da região de confiança. De fato, nem é mesmo necessário levar em conta uma busca, porque o caminho descrito pelo método Dogleg intercepta a fronteira da região de confiança uma única vez e o ponto de intersecção pode ser computado analiticamente resolvendo a equação

$$\left\|D_k(p^U + (\tau - 1)(p_{\mathcal{N}}^k - p^U))\right\|^2 = \Delta^2, \tag{3.50}$$

ver([11],Capítulo 3, lema 3.3). A Figura 3.4 resume a idéia do Dogleg.

O método Dogleg não resolve exatamente o subproblema de otimização (3.7), mas garante reduções na função objetivo que satisfazem a condição de decréscimo (3.12) da estratégia de região de confiança, o que permite avanços na direção da solução do problema.

Sob certas condições o passo $p_{\mathcal{N}}^k$ estará sempre contido na região de confiança e portanto será solução de (3.7) o que nos garante convergência quadrática. Este resultado está demonstrado em [12]. No final deste Capítulo apresentaremos este resultados, pois estas condições só fazem sentido após a definição do Algoritmo 3.3 para resolver um sistema não linear com restrições de canalização.

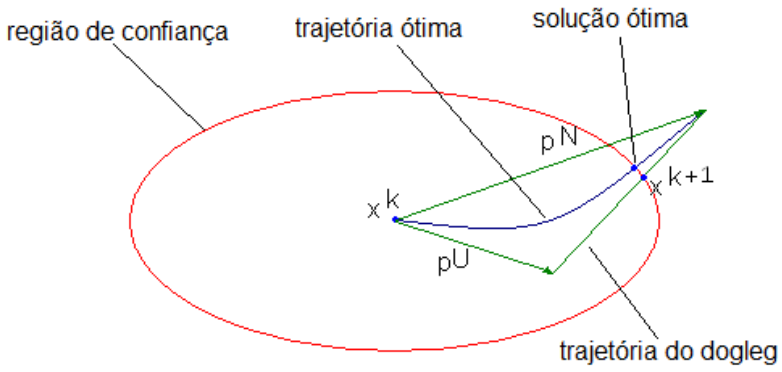

Figura 3.4: A trajetória do Dogleg aproxima a trajetória das soluções ótimas.

3.6 Algoritmo de Região de Confiança com Duplo Critério de Decréscimo

O trabalho apresentado em [11] mostra um método de região de confiança para resolver o problema (3.3), baseado na solução do subproblema (3.7), que por sua vez usa uma região de confiança com matriz de escala, esta matriz sendo definida a partir das condições de otimalidade, como pode ser visto na Seção 3.3. Uma vez encontrada a solução p^k para o subproblema (3.7), esta é truncada, ou seja, ao invés de usar p^k usaremos $\alpha(p^k)$, como foi definido em (3.29), para assegurar a viabilidade estrita. A idéia essencial por trás do método de região de confiança é ajustar o tamanho da região de confiança para assegurar uma decréscimo suficiente da função objetivo. Considere o caso irrestrito: o tamanho da região de confiança é atualizado para assegurar que a redução de uma função não linear $F(x)$ seja pelo menos uma fração da redução do modelo quadrático. Especificamente, a atualização do tamanho da região de confiança nos leva à condição (3.8)

$$\rho_f^k = \frac{F(x^k + p^k) - F(x^k)}{m_k(0) - m_k(p^k)} > \mu, \tag{3.51}$$

para alguma constante $\mu \in (0, 1)$.

Para obter a convergência de primeira ordem do método de região de confiança irrestrito, seguimos a proposta, apresentada em [3], de um novo critério de redução suficiente do modelo quadrático $m_k(p)$ na região de confiança. Usando a definição (3.13) para a redução do modelo quadrático,

$$\bar{P}_{red}(\alpha(p^k)) > \beta \max_{\tau} \left\{ \bar{P}_{red}(p) : p = -\tau D_k^2 \nabla F_k,\ \|D_k p\| \leq \Delta_k \right\}, \tag{3.52}$$

para alguma constante $\beta \in (0,1)$.

Levando em consideração o passo de Cauchy (3.38) e o truncamento, o novo critério (3.13) pode ser escrito como:

$$\bar{P}_{red}(\alpha(p^k)) > \beta \bar{P}_{red}(\alpha(p_C^k)). \tag{3.53}$$

e assim definimos,

$$\rho_C^k = \frac{\bar{P}_{red}(\alpha(p^k))}{\bar{P}_{red}(\alpha(p_C^k))} > \beta. \tag{3.54}$$

Mas infelizmente, o passo truncado ao longo da solução p^k do subproblema (3.7) pode não reduzir suficientemente o modelo quadrático $m_k(p)$ por causa do efeito do truncamento, isto é, a condição (3.52) pode não ser satisfeita quando escolhemos $p = \alpha(p^k)$ e p^k resolve (3.7). No entanto, a solução de qualquer subproblema de região de confiança com um gradiente não nulo, se aproxima da direção do gradiente quando o tamanho da região de confiança se aproxima de zero. Podemos observar isso, considerando a solução ótima do subproblema de otimização (3.7) como p^* cujo raio da região de confiança é $\Delta > 0$, pela Figura 3.5.

Assumindo que um passo ao longo da direção de máxima descida escalada $D_k^{-2}\nabla F_k$ produz uma redução suficiente de $\bar{P}_{red}(p^k)$, o ajuste do tamanho de Δ_k pode ser usado como último recurso para garantir um decréscimo suficiente. Em particular, a redução do tamanho da região de confiança pode ser usada para que a condição (3.54) seja satisfeita. Portanto, podemos redimensionar a região de confiança tal que o modelo quadrático $m_k(p)$ e a função $F(x)$ são suficientemente reduzidas.

As considerações feitas até aqui nos levam à definição do seguinte algoritmo, conforme proposto em [11]:

Algoritmo 3.3 *: Algoritmo para sistemas não lineares com restrições de canalização:*

Sejam $x^0 \in int(\Omega)$, $\Delta_{min} > 0$, $\Delta > \Delta_{min}$ *e* $\theta,\ \beta_1,\ \beta_2,\ \beta_3,\ \delta_0,\ \delta_1 \in (0,1)$ *e* δ_2 *tal que* $\beta_1 < \beta_2 < \beta_3 < 1$ *e* $\delta_0 < \delta_1 < 1 < \delta_2$.
Para $k = 1, 2, \ldots$

1. *Calcule* $C'(x^k)$, $C(x^k)$ *e* $D(x^k)$

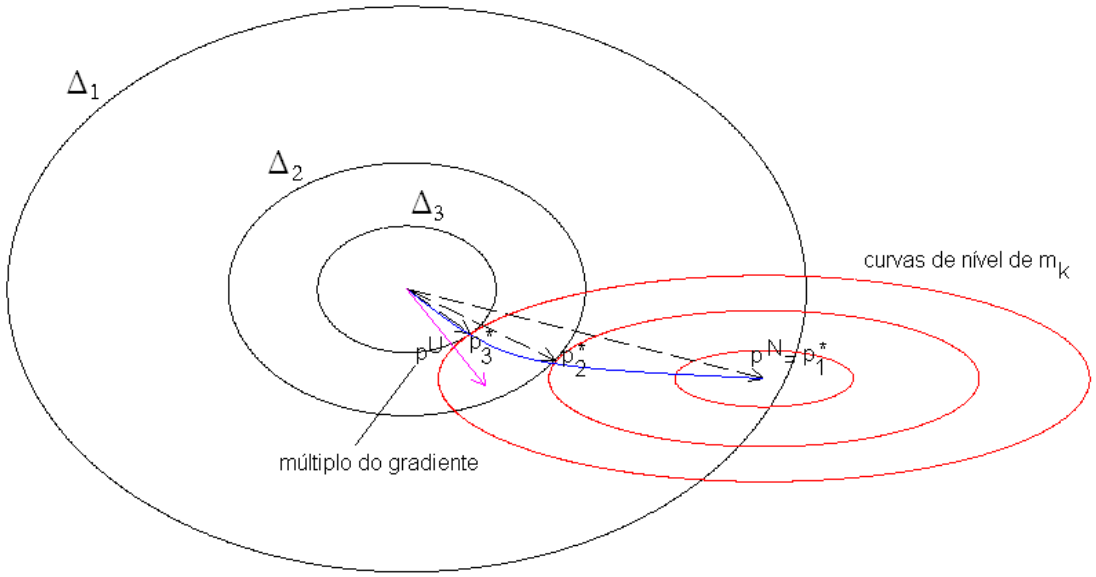

Figura 3.5: A solução do subproblema de otimização se aproxima da direção do gradiente quando o raio da região de confiança diminui.

2. *Se* $\left\|\nabla F(x^k)\right\| = 0$ *termine a execução do algoritmo. Neste caso* x^k *é um ponto estacionário do problema (3.3).*

3. *Calcule a direção de Newton de norma-2 mínima:*

$$p_{\mathcal{N}}^k = -C(x^k)^\dagger C(x^k).$$

4. *Se*

$$\left\|D(x^k)p_{\mathcal{N}}^k\right\| \leq \Delta \text{ e } \rho_C^k(p_{\mathcal{N}}^k) \geq \beta_1, \tag{3.55}$$

defina $p^k = p_{\mathcal{N}}^k$. *Senão encontre* p^k *tal que* $\left\|D(x^k)p^k\right\| \leq \Delta$ *e* $\rho_C^k(p^k) \geq \beta_1$.

5. *Se*

$$\rho_f^k(p^k) \geq \beta_2 \tag{3.56}$$

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

6. *Se*

$$\rho_f^k(p^k) \geq \beta_3 \tag{3.57}$$

Então $\Delta_{novo} = \max\{\delta_2\Delta, \Delta_{max}\}$.
Faça $k \leftarrow k+1$, $\Delta \leftarrow \Delta_{novo}$ *e vá para 1.*

Fim Para

Agora façamos alguns comentários para o melhor entendimento do Algoritmo 3.3.

3.6.1 Observações

No Algoritmo 3.3 calculamos primeiramente o passo de Newton de norma-2 mínima (3.44) e verificamos se $p^k = p^k_{\mathcal{N}}$ satisfaz a condição (3.55), ou seja, se p^k está contido na região de confiança e se está, esta escolha de passo é melhor do que o passo de Cauchy na redução do modelo quadrático m_k da função objetivo (3.7). Em caso positivo o passo de Newton é a solução do subproblema de otimização (3.41). Caso contrário, para encontrar a solução de (3.41) podemos usar o passo definido pelo método Dogleg (3.49) e caso esta opção falhe, podemos usar como último recurso o próprio passo de Cauchy p^k_C (3.38) que satisfaz a condição (3.55).

Definido o passo p^k verificamos se a condição de decréscimo em relação à função objetivo (3.56) é satisfeita. Neste caso aceitamos o passo p^k e calculamos $x^{k+1} = x^k + \alpha(p^k)$, lembrando que $\alpha(p^k)$ é um múltiplo de p^k definido de maneira a fazer $x^{k+1} \in \Omega$, como foi descrito na Seção 3.4. Caso contrário diminuímos o valor de Δ, ou seja, encolhemos o tamanho da região de confiança.

No passo 6 do Algoritmo 3.3, presente apenas no algoritmo desenvolvido em [3], verificamos se a condição (3.57) foi satisfeita, que é a mesma condição de decréscimo (3.56), mas para um valor $\beta_3 > \beta_2$. Uma vez que esta condição é satisfeita significa que obtemos uma redução suficiente da função objetivo quando comparada à redução prevista pelo modelo quadrático (3.7), por isso aumentamos Δ, o raio da região de confiança. Caso esta condição não seja satisfeita, mantemos o valor de Δ inalterado.

3.7 Alguns resultados relevantes

Após a definição do Algoritmo 3.3 é necessário apresentar alguns resultados sobre a definição do passo p^k que resolve o subproblema de otimização (3.41) apresentado na Seção 3.5. Tais resultados garantem a existência de um passo τ que intercepta a fronteira na definição da direção do método Dogleg, e a convergência do Algoritmo 3.3 (ver [11]). Para isso assumimos as seguintes hipóteses:

H1 A seqüência $\{x^k\}$ gerada pelo Algoritmo 3.3 é limitada. Uma condição suficiente para isto é que o conjunto de nível

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

é limitado.

H2 Para todo par $(x,\ y)$ em um conjunto aberto, limitado e convexo L que contém toda seqüência gerada pelo Algoritmo 3.3 e todos os pontos da forma $x^k + \alpha(p^k)$, temos que:

$$\|C'(x) - C'(y)\| \leq 2\gamma_0 \|x - y\|. \tag{3.58}$$

Uma condição suficiente para isto é que (3.58) seja válida para todo $x,\ y \in \Omega$.

H3 $C'(x)$ tem posto completo m para todo $x \in L$

Uma vez que estas hipóteses sejam satisfeitas, valem os seguintes resultados:

Teorema 3.7.1 *Assuma que H1 e H2 são satisfeitas e que o Algoritmo 3.3 gera uma sequência infinita* $\{x^k\}$. *Então todos os pontos limites são pontos estacionários de*

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

Corolário 3.7.1 *Assuma que as hipóteses* $H1$, $H2$ *e* $H3$ *são satisfeitas. Suponha que a seqüência* $\{x^k\}$ *gerada pelo Algoritmo 3.3 é infinita. Seja* $x^* \in int(\Omega)$ *um ponto-limite de* $\{x^k\}$, *então* $C(x^*) = 0$.

Lema 3.7.1 *Assuma que* $H1$, $H2$ *e* $H3$ *são satisfeitas,* K *é uma seqüência de índices tal que*

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

e

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

Então existe $k_o \in \{1, 2, \ldots\}$ *tal que para* $k \geq k_0$, $k \in K$, *o passo de Newton* $p_{\mathcal{N}}^k$ *dado por (3.44) satisfaz as condições dos passos 4 e 5 do Algoritmo 3.3.*

Este lema nos diz que a partir de uma certa iteração k_0, o Algoritmo 3.3 escolherá sempre o passo de Newton como solução do subproblema de otimização (3.41), o que nos garante convergência quadrática. Isto é evidenciado no seguinte teorema:

Teorema 3.7.2 *Suponha que as hipóteses $H1$, $H2$ e $H3$ valem e seja $x^* \in \Omega$ um ponto de acumulação da sequência $\{x^k\}$ gerada pelo Algoritmo 2.1. Então, $C(x^*) = 0$, e a taxa de convergência é localmente quadrática.*

Antes de alcançarmos a iteração k_0 usamos o passo definido pelo método Dogleg para resolver o subproblema de otimização. E caso as condições definidas nos passos 4 e 5 do algoritmo não sejam satisfeitas usamos o passo de Cauchy (3.38) como último recurso para resolver o subproblema (3.41), pois esta direção sendo um múltiplo do $-\nabla F$ nos garante um decréscimo da função objetivo F.

A demonstração destes resultados e de outros resultados sobre a convergência e boa definição do Algoritmo 3.3 são encontrados em [11].

3.8 Testes Numéricos para o Algoritmo de Sistemas Não Lineares com Restrições de Canalização

Nesta seção apresentamos testes numéricos para o Algoritmo 3.3. O propósito destes testes é validar a implementação deste algoritmo realizada em MatLab 7.0 em um computador com CPU de 2.80 GHz e 496 MB de RAM com o sistema operacional Windows XP. Os problemas teste usados para o Algoritmo 3.3 são da forma:

$$F(x) = 0, \ x \in \Omega \tag{3.59}$$

onde Ω é uma caixa e $F : \mathbb{R}^n \to \mathbb{R}^m$.

Empregamos o mesmo conjunto de testes apresentado em [11]. Os 11 primeiros testes são sistemas não lineares canalizados que definem conjuntos viáveis de problemas de programação não linear propostos por Hock e Schittkowski [15], mais dois problemas apresentados no artigo de C. Kanzow [16] sobre o método de Levenberg-Marquardt, e o último foi retirado do próprio trabalho do J. B. Francisco e J. M. Martínez [11].

A Tabela 3.1 lista os problemas testados, sendo a primeira coluna o número do problema, a segunda coluna a fonte do problema e terceira e quarta colunas as respectivas dimensões m e n.

Os 14 problemas teste mencionados acima são apresentados no Apêndice no final deste trabalho. Os resultados da primeira parte são similares aos resultados apresentados em [11], onde foi empregado o mesmo algoritmo, mas foram usadas diferenças finitas para aproximar a matriz Jacobiana de cada problema.

Os problemas 1, 3, 6 e 7 eram, originalmente, irrestritos. Então, introduzimos limites artificiais $0 \leq x_i \leq 2.5$ para todo i de 1 a m. No problema 10 as variáveis

Problema	Origem do problema	m	n
1	problema 46 de [15]	2	5
2	problema 53 de [15]	3	5
3	problema 56 de [15]	4	7
4	problema 63 de [15]	2	3
5	problema 75 de [15]	3	4
6	problema 77 de [15]	2	5
7	problema 79 de [15]	3	5
8	problema 81 de [15]	3	5
9	problema 87 de [15]	4	6
10	problema 107 de [15]	6	9
11	problema 111 de [15]	3	10
12	problema 2 de [16]	150	300
13	problema 4 de [16]	150	300
14	problema 15 de [11]	1	2

Tabela 3.1: Problemas teste.

x_1 e x_2 são ilimitadas superiormente, e no problema 4 todas as variáveis são ilimitadas superiormente. Problema 12 é um sistema linear e o Problema 13 é um sistema quadrático.

Vamos enumerar algumas características da implementação:

1. não foram usadas diferenças finitas, mas as expressões originais das matrizes Jacobianas e demais derivadas;
2. o ponto x^k foi aceito como solução se $\left\|F(x^k)\right\| \leq 10^{-6}$;
3. o número máximo de iterações permitidas foi 5000;
4. os parâmetros do Algoritmo 3.3 foram: $\Delta = \|D(x^0)^{-1}\nabla f(x^0)\|$, $\Delta_{min} = 5 \times 10^{-4}$, $\theta = 0.99995$, $\beta_1 = 0.1$, $\beta_2 = 0.25$, $\beta_3 = 0.75$ e $\delta_1 = 0.25$;
5. a escolha do Δ_{novo} no passo 5 do Algoritmo 3.3 foi:
$$\Delta_{novo} = \min\left\{\delta_1\Delta, \frac{1}{2}\left\|D_k\alpha(p_k)\right\|\right\};$$
6. no passo 6, se $\rho_f^k(p^k) \geq \beta_3$ então,
$$\Delta = max\left\{\Delta_{min}, \Delta, 2\left\|D_k\alpha(p^k)\right\|\right\},$$

caso contrário

$$\Delta = max\left\{\Delta_{min}, \Delta\right\};$$

7. no passo 4 do Algoritmo 3.3 precisamos encontrar um direção p^k satisfazendo $\rho_C^k(p^k) \geq \beta_1$ e $\left\|D^k p^k \leq \Delta\right\|$. Com este objetivo, usamos o método Dogleg apresentado na Seção 2.3.2. De acordo com as expressões (3.48), (3.49) e (3.50) a direção p_d^k do Dogleg é computada como:

$$p_d^k = \begin{cases} \frac{-\Delta D_k^{-2}\nabla f_k}{\left\|D_k^{-1}\nabla f_k\right\|}, & \text{se } \left\|D_k^{-1}\nabla f_k\right\| \geq \Delta \\ p_C^k + (\mu - 1)(p_{\mathcal{N}}^k - p_C^k), & \text{caso contrário,} \end{cases} \tag{3.60}$$

onde μ é a solução positiva de

$$\left\|D_k(p_C^k + (\mu - 1)(p_{\mathcal{N}}^k - p_C^k))\right\|^2 = \Delta^2.$$

Se p_d^k não satisfaz a condição $\rho_C^k(p_d^k) \geq \beta_1$, escolhemos $p^k = p_C^k$. Desta maneira garantimos que o passo que satisfaz a condição de decréscimo (3.54) é obtido.

8. Quando a condição $\rho_C^k(p_d^k) \geq \beta_1$ não é satisfeita, usamos na prática $p^k = 0.962 p_C^k$, o que diminuiu o número de iterações do método. Multiplicamos p_C^k pela constante 0.962 para que não fiquemos muito próximos da fronteira do problema e o algoritmo possa ter mais opções de pontos a partir da direção p_C^k. A constante 0.962 foi definida por testes sucessivos como a que mais diminuiu o número de iterações.

9. Para encontrar a direção de norma-2 mínima de Newton foi usada fatoração QR na matriz Jacobiana.

10. para variáveis ilimitadas usamos a constante $\pm 2.0 \times 10^{16}$ para representar o infinito.

A aproximação inicial x^0 foi escolhida como $x^0 = (l + u)/2$ exceto nas seguintes situações:

1. nos problemas 2 e 8 tomamos $x^0 = l + \frac{1}{4}(u - l)$, porque o ponto inicial coincide com o ponto estacionário de $\min_{x\in\Omega} F(x)$.

2. no problema 10 tomamos o ponto inicial como $x_i^0 = 3$, pois as variáveis são ilimitadas superiormente;

3. nos problemas 13 e 14 tomamos $x^0 = 150(1, \ldots, 1)^T$, que é um dos pontos iniciais sugerido em [16];

4. no problema 15 consideramos $x^0 = (-\frac{1}{2}, \frac{1}{2})$.

Nas Tabelas 3.2 e 3.3 apresentamos os resultados do Algoritmo 3.3 para este conjunto de problemas e os resultados apresentados em [11], respectivamente. A quarta coluna mostra a norma de F avaliada na respectiva aproximação inicial, a quinta coluna mostra a norma de F na solução encontrada x^*, a sexta coluna mostra o número de avaliações da função F e a a última coluna mostra o número de iterações na qual a direção do passo de Cauchy (p_C^k), a direção do Dogleg (p_d^k) e a direção de Newton ($p_{\mathcal{N}}^k$) foram aceitas, respectivamente. A notação $MAXIT$ significa que o algoritmo ultrapassou o limite máximo de iterações e $SINGUL$ nos diz que a matriz $C'C'^T$ se tornou singular.

Problema	(m,n)	Iterações	$\|F(x^0)\|$	$\|F(x^*)\|$	Avaliações	$(p_C^k)/(p_d^k)/(p_{\mathcal{N}}^k)$
1	(2,5)	5	3.2	5.3×10^{-10}	6	0/0/5
2	(3,5)	1	2.0×10^{10}	5.9×10^{-15}	2	0/0/1
3	(4,5)	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)	4	8.5×10^2	2.6×10^{-11}	5	0/0/4
6	(2,5)	13	4.4	9.0×10^{-7}	14	0/0/13
7	(3,5)	8	1.5	4.9×10^{-7}	9	0/0/8
8	(3,5)	10	8.3	1.2×10^{-9}	11	2/5/7
9	(4,6)	96	3.3×10^2	4.9×10^{-9}	97	89/0/7
10	(6,9)	62	1.4×10^{16}	8.3×10^{-12}	63	0/0/13
11	(3,10)	6	8.1	1.1×10^{-12}	7	0/0/6
12	(150,300)	1	2.2×10^4	2.5×10^{-12}	2	0/0/1
13	(150,300)	12	2.2×10^4	5.5×10^{-12}	13	0/0/12
14	(1,2)	10	5.1×10^{-1}	4.7×10^{-7}	11	1/1/2

Tabela 3.2: Resultados numéricos para o Algoritmo 3.3

A diferença entre os resultados se deve primeiro ao uso de pontos inciais diferentes. Apesar de usarmos o mesmo método indicado em [11] para a escolha dos pontos iniciais, estes pontos diferem em alguns problemas, o que é indicado pelas diferenças dos valores entre as duas tabelas na coluna $\|F(x^0)\|$ e por não usarmos diferenças finitas para aproximar a matriz Jacobiana. O Algoritmo 3.3 para sistemas não lineares já apresentou bons resultados no artigo apresentado por J. B. Francisco e J. M. Martínez [11],

Problema	(m,n)	Iterações	$\|F(x^0)\|$	$\|F(x^*)\|$	Avaliações	$(p_C^k)/(p_d^k)/(p_{\mathcal{N}}^k)$
1	(2,5)	5	3.2	5.3×10^{-10}	6	0/0/5
2	(3,5)	1	1.0×10^{10}	3.2×10^{-15}	2	0/0/1
3	(4,5)	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^{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)	62	1.4×10^{16}	8.3×10^{-12}	63	0/0/13
11	(3,10)	16	9.5	8.8×10^{-7}	17	0/0/16
12	(150,300)	2	6.5×10^{3}	0.0	3	0/0/2
13	(150,300)	11	2.7×10^{5}	8.3×10^{-12}	12	0/0/11
14	(1,2)	53	5.1×10^{-1}	5.8×10^{-14}	54	51/0/2

Tabela 3.3: Resultados numéricos da versão do Algoritmo 3.3 mostrada em [11]

mas usando a expressão original da matriz Jacobiana, ao invés de utilizar diferenças, obtivemos resultados ainda melhores do que era esperado. Uma vez que verificamos que a implementação do algoritmo está funcionando bem, o utilizaremos na fase de restauração do algoritmo de Restauração Inexata. Os resultados são mostrados na próxima seção.

Capítulo 4

Testes Numéricos para a Restauração Inexata

4.1 Introdução

Neste capítulo apresentamos os testes numéricos para o método de Restauração Inexata usando os mesmos problemas apresentados na Tabela 3.1, utilizando agora a formulação como um problema de minimização e adicionaremos um problema retirado de [14] que se trata de um teste de otimização global restrita. Descrevemos os detalhes da implementação, seguindo os passos do Algoritmo 2.1 do método de Restauração Inexata, explorando os seus aspectos computacionais e dificuldades que surgiram durante os testes. Para realizar os testes usamos o MatLab 7.7 com um computador com CPU com dois processadores de 2.40GHz e 3Gb de RAM.

4.2 Implementação do Algoritmo 2.1

4.2.1 Inicialização

O Algoritmo 2.1 do método de Restauração Inexata possui convergência bastante sensível a mudanças em seus parâmetros e com relação ao próprio problema de otimização a ser resolvido. O principal parâmetro que afeta o desempenho do algoritmo é o valor inicial do passo espectral η_0. Para alguns problemas onde a função objetivo f sempre apresentou curvatura positiva ao longo das direções geradas, é interessante que η seja grande, que é o caso do problema 1. Mas para outros problemas onde a concavidade é até negativa, um valor muito grande de η leva à divergência do método. Temos ainda dificuldades que podem surgir com o posto da matriz Jacobiana $C'(x)$

que deve ser completo de acordo com a teoria. Optamos por iniciar o valor $\eta_0 = 2n$, lembrando que n é o número de variáveis do problema. Porém este valor ainda era ruim para alguns problemas. Por isso tomamos dois pontos na vizinhança de x^0 para realizar uma avaliação do valor inicial de η da forma $x_e^0 = x^0 - \epsilon$ e $x_d^0 = x^0 + \epsilon$, onde ϵ é um vetor com todas as componentes iguais 10^{-2}. Com estes dois pontos usamos o procedimento padrão para calcular η_k que foi descrito no passo 8 do Algoritmo 2.1

$$s^0 = x_d^0 - x_e^0 = 2\epsilon \tag{4.1}$$

$$u^0 = \nabla f(x_d^0) - \nabla f(x_e^0) \tag{4.2}$$

$$\eta_0 = \frac{(s^0)^T s^0}{(s^0)^T u^0} = \frac{2 * \|\epsilon\|^2}{\epsilon^T u^0}. \tag{4.3}$$

Como η_0 é um parâmetro sensível a mudanças também impomos limites a seu valor, ou seja,

$$\eta_0 \in [\eta_{min}, \eta_{max}]$$

onde $\eta_{min} = 10^{-3}$ e $\eta_{max} = 10^3$. A partir destes fatos redefinimos a escolha η_0 da seguinte maneira: primeiro calculamos η_0 conforme (4.3) e

$$\eta_0 \leftarrow \begin{cases} \min(\eta_{max}, \max(\eta_{min}, \eta)) & \text{se } 0 \leq \eta_0 \leq \eta_{max} \\ m/n & \text{caso contrário} \end{cases} \tag{4.4}$$

O valor $\eta_0 = \frac{m}{n}$ é escolhido no caso de η_0 ser negativo ou ultrapassar o limite do η_{max}. No caso de $\eta_0 < 0$ a função pode não ter curvatura positiva ao longo da direção s^0 no ponto x^0, portanto escolhemos um valor positivo e pequeno para η_0, no caso m/n que é sempre menor que 1, pois nos problemas teste o número de restrições m é sempre menor que o número de variáveis n.

4.2.2 Passo 1: Restauração

Na fase de restauração usamos o Algoritmo 3.3 para obter y^k que satisfaça as condições:

$$\left\|C(y^k)\right\| \leq r \left\|C(x^k)\right\| \tag{4.5}$$

$$\left\|y^k - x^k\right\| \leq \beta \left\|C(x^k)\right\|. \tag{4.6}$$

A condição (4.5) é usada como critério de parada para o Algoritmo 3.3 para sistemas não lineares com restrições de canalização com o parâmetro $r = 0.5$ fixo. O parâmetro r também pode variar dentro do intervalo $(0, 1)$ de acordo com o desempenho do algoritmo, para acelerar a convergência do método, aumentando o valor de r gradativamente, como foi feito no algoritmo apresentado no artigo de J. M. Martínez e E. A. Pillota de 2004 [19].

A segunda inequação (4.6) é construída de forma a evitar que os pontos x^k e o ponto restaurado y^k se distanciem. Porém, a norma $\left\|C(x^k)\right\|$ pode ser muito pequena, mesmo quando x^k e y^k estão relativamente distantes. Por isso escolhemos $\beta = 1 \times 10^6$, grande se comparado aos outros parâmetros, para diminuir esta discrepância. Para esta inequação usamos a norma infinito, ou seja,

$$\max_i \left|y_i^k - x_i^k\right| \leq \beta \left\|C(x^k)\right\| \text{ para } i = 1 \ldots n,$$

então cada componente do novo ponto y^k deve ficar restrita a:

$$x_i^k - \beta \left\|C(x^k)\right\| \leq y_i^k \leq x_i^k + \beta \left\|C(x^k)\right\| \text{ para } i = 1 \ldots n.$$

Por este motivo trocamos os limites l e u no Algoritmo 3.3, para realizar o processo de restauração, por l^{max} e u^{min}, respectivamente, definidos componente a componente da seguinte maneira:

$$l_i^{max} = \max(l_i, x_i - \beta \left\|C(x^k)\right\|), \tag{4.7}$$

$$u_i^{min} = \min(u_i, x_i + \beta \left\|C(x^k)\right\|). \tag{4.8}$$

Desta maneira os pontos obtidos no processo de restauração pertencem a Ω e satisfazem a condição (4.6). Também aplicamos o processo de restauração com os mesmos limites l_i^{max} (4.7) e l_i^{min} (4.8) ao ponto z^k obtido após a fase de otimalidade, para acelerar a convergência.

4.2.3 Passo 2: Definição do Parâmetro de Penalidade

Para inicializar o parâmetro de penalidade θ que dá os pesos da viabilidade e da otimalidade para função de mérito (2.18)

$$\psi(x, \theta) = \theta f(x) + (1 - \theta) \left\|C(x)\right\| \tag{4.9}$$

escolhemos o valor $\theta_0 = 0.5$ para dar pesos iguais a viabilidade e a otimalidade, seguindo a mesma ideia da fase de restauração. Durante as iterações para atualizar o valor de θ_k, de forma a reequilibrar a otimalidade e a viabilidade e reduzir o valor da função de mérito, usamos um vetor dinâmico v_θ cuja dimensão é igual ao número da iteração corrente $k + 1$, pois precisamos guardar os valores de θ para calcular θ_k em função do valor mínimo dos θ's anteriores. Além disso adicionamos a θ_k o k-ésimo termo da seqüência

$$\{\omega_k\} = \left\{ \frac{n}{(1+k)^{1.01}} \right\}$$

que converge lentamente a zero para evitar que θ_k se anule rapidamente conforme foi discutido na Seção 2.3.1. Portanto, armazenamos os θ_k's da seguinte maneira: seja $dim\ v_\theta$ a dimensão do vetor v_θ, então

$$\text{Para } k = 1; \tag{4.10}$$

$$\begin{aligned} dim\ v_\theta &\leftarrow 2; \\ \theta_0 &\leftarrow 0.5; \\ v_\theta(1) &\leftarrow 1; \\ v_\theta(2) &\leftarrow \theta_0; \end{aligned} \tag{4.11}$$

$$\begin{aligned} \text{Para } k > 1 \\ dim\ v_\theta \leftarrow dim\ v_\theta + 1; \\ \omega_k = \frac{n}{(1+k)^{1.01}}; \\ \theta_{min} \leftarrow \min v_\theta \\ \theta_k \leftarrow \min\left(1, \theta_{min} + \omega_k\right) \end{aligned}$$

4.2.4 Passos 3 e 4: Definição do Conjunto Tangente, Cálculo da Direção de Busca e Critério de Parada

A determinação da direção de descida d^k_{tan} requer a solução do problema:

$$\begin{aligned} \min_d \quad & \left\|d + y^k\right\| \\ s.a \quad & C'(y^k)d = 0 \\ & l \leq y^k + d \leq u. \end{aligned} \tag{4.12}$$

Usamos a função `fmincon` do Matlab 7.7 para resolução de problemas de otimização restrita da forma:

$$\begin{aligned} \min \quad & FUN(x) \\ s.a \quad & Ax \leq b \\ & Aeqx = beq \\ & N(x) \leq 0 \\ & Neq(x) = 0 \\ & L \leq x \leq U, \end{aligned} \tag{4.13}$$

onde A, b e Aeq, beq são matrizes que representam as restrições lineares e $N(x)$ e $Neq(x)$ representam as restrições não lineares. Para resolver o problema usamos a seguinte linha de comando, dado o ponto inicial x_0:

```
X=FMINCON(FUN,x0,A,B,Aeq,Beq,LB,UB,NONLCON,OPTIONS)
```

onde $NONLCON = [N(x), Neq(x)]$ e OPTIONS=optimset, que define parâmetros como critério de parada e número máximo de iterações. Os parâmetros do FUN e NONLCON devem ser guardados como funções do Matlab.

Para resolver o problema (4.13) usamos

$$d_{tan}^k = \texttt{FMINCON(f,yk,'','',C'(yk),zeros(m,1),l-yk,u-yk,'',OPTIONS)}$$

e

$$\texttt{OPTIONS=optimset('TolCon', 1.d-04,'MaxFunEvals',1000)},$$

onde $zeros(m, 1)$ é um vetor de dimensão m com todas as entradas nulas, $Tolcon = 10^{-4}$ é a tolerância do método em relação a restrição $C'(y^k)d = 0$ e $MaxFunEvals = 1000$ é o número máximo de iterações para o `fmincon`. E a escolha do ponto inicial é o próprio y^k.

Após o cálculo de d_{tan}^k, verificamos o critério de parada do Algoritmo 2.1:

$$\left\|C(y^k)\right\| \leq \varepsilon_1 \ e \ \left\|d_{tan}^k\right\| \leq \varepsilon_2. \tag{4.14}$$

Os valores escolhidos foram $\varepsilon_1 = 10^{-8}$ e $\varepsilon_2 = 10^{-4}$. A diferença entre os dois valores se deve ao fato de $\left\|d_{tan}^k\right\|$, durante os testes, sempre convergir a zero mais lentamente do que $\left\|C(y^k)\right\|$ e além disso um d_{tan}^k com norma pequena produziria poucas alterações no valor de x^k e $f(x^k)$, portanto não vale a pena exigir que ε_2 seja muito pequeno. O valor para ε_1 é adequado se pensarmos no valores que obtivemos para $\left\|F(x^k)\right\|$, nos testes para o Algoritmo 2.1, que quase sempre ficaram abaixo de 10^{-8}.

4.2.5 Passo 5: Determinação do Passo da Direção de Descida

Uma vez que a direção d_{tan}^k foi determinada temos que calcular o passo nesta direção para satisfazer a condição de Armijo (2.10) conforme foi descrito no passo 5 do Algoritmo 2.1. Para isto fazemos o *backtracking* da seguinte maneira:

$$\begin{aligned}
&t \leftarrow 1;\\
&z^k \leftarrow y^k + td_{tan}^k;\\
&\text{Enquanto} f(z^k) \geq f(y^k) + \alpha \nabla f(y^k)^T d_{tan}^k \ e \ t > \varepsilon_t\\
&t \leftarrow 0.5t;\\
&z \leftarrow y^k + td_{tan}^k;
\end{aligned} \tag{4.15}$$

A condição $t > \varepsilon_t$ é uma salvaguarda para prevenir o risco de t ficar muito pequeno e estagnar o algoritmo. Usamos $\varepsilon_t = 10^{-8}$. Mas caso esta restrição não seja satisfeita o algoritmo é interrompido. Nos testes realizados fixamos $\alpha = 0.1$.

Depois de θ ser atualizado, realizamos o passo 7 exatamente como está descrito no Algoritmo 2.1, onde também há um *backtracking*. Neste laço também impomos a condição de $t > \varepsilon_t$.

4.2.6 Passo 6: Atualização do Parâmetro de Penalidade.

Para atualizar o parâmetro θ_k da maneira que o passo 6 do Algoritmo 2.1 exige, recordemos que:

$$P_{red} = \theta_k \left[f(x^k) - f(z^k)\right] + (1-\theta_k)\left[\left\|C(x^k)\right\| - \left\|C(y^k)\right\|\right],$$

então construímos o seguinte laço:

$$\text{Enquanto } P_{red} < \frac{1}{2}\left[\left\|C(x^k)\right\| - \left\|C(y^k)\right\|\right] \text{ e } \theta_k > \varepsilon_\theta \tag{4.16}$$

$$\theta_k = 0.9\theta_k; \tag{4.17}$$

Mesmo com a adição de ω_k a seqüência θ_k pode assumir valores muito pequenos. Então impomos que $\theta_k > \varepsilon_\theta$, com $\varepsilon_\theta = 10^{-8}$, caso contrário o algoritmo é interrompido, pois não conseguimos determinar θ que satisfaça o passo 6 do Algoritmo 2.1.

4.2.7 Passo 7: Decréscimo Mínimo

O Passo 7 é realizado exatamente como descrito no Algoritmo 2.1, verificando se $Ared \geq 0.1Pred$ (2.20) e caso não seja fazemos uma busca linear em z na direção d^k_{tan} fazendo $z \leftarrow z + t_{k,i}d^k_{tan}$.

4.2.8 Passo 8: Passo Espectral

Durante as iterações, o cálculo de η_k também é um fator crítico do algoritmo. Não podemos tomar valores muito grandes para η, pois corremos o risco da matriz Jacobiana $C'(x)$ ficar com posto incompleto, e não podemos escolher valores pequenos, pois o algoritmo pode ficar estagnado dando passos muito pequenos em direção à solução. Portanto introduzimos um valor variável de η quando este toma valores negativos. Dessa maneira redefinimos η_k

$$s^k = x^k - x^{k-1}, \tag{4.18}$$

$$u^k = \nabla f(x^k) - \nabla f(x^{k-1}), \tag{4.19}$$

$$\text{Se } (s^k)^T u^k \leq 0 \text{ defina } \eta_{k+1} = 0.99\eta_k \tag{4.20}$$

$$\text{Caso contrário, } \eta_{k+1} = \max\left\{\min\left\{\frac{(s^k)^T s^k}{(s^k)^T u^k}, \eta_{max}\right\}, \eta_{min}\right\} \tag{4.21}$$

O fator 0.99 tem a finalidade de diminuir o valor de η_k durante as iterações do Algoritmo 2.1. Isto é necessário quando a função não é convexa, e $(s^k)^T u^k < 0$ sempre, então $\eta_{k+1} \leftarrow \eta_k$ e seu valor não varia. Por isso diminuímos um pouco o valor de η a cada iteração, o que melhora a convergência em alguns testes. Escolhemos η_k como próximo valor do passo espectral, ao invés do η_{max}, ou seja, o último valor bem sucedido, para preservar as propriedades de convergência local. Quando escolhemos η_{max} estamos escolhendo um valor que nos retiraria dessa região onde a função é não convexa, mas não possuímos informações sobre a convexidade da região para onde o algoritmo iria. Nesta situação a convergência do algoritmo é similar a do método de máxima descida para problemas irrestritos.

4.3 Testes Numéricos

Para testar o algoritmo de Restauração Inexata aplicamos os mesmos problemas teste apresentados na Tabela 3.1 com as mesmas restrições de canalização usadas para o teste do Algoritmo 3.3 para sistemas não lineares. Além disso acrescentamos mais um problema que será designado como problema 15 retirado do conjunto de problemas teste para otimização global com restrições, de A. Hedar, [14]

O valor da função objetivo f para o minimizador local de cada problema teste é apresentado na Tabela 4.1. Mais detalhes sobre os problemas teste são encontrados no Capítulo 6: Apêndice dos Problemas Teste.

Os problemas 12, 13 e 14 utilizados no teste do Algoritmo 3.3, não são problemas de otimização do tipo (2.1), mas apenas sistemas não lineares com restrições de canalização. Os limites l e u para a variável x são os mesmos usados nos testes para o algoritmo para sistemas não lineares com restrições de canalização. Os pontos iniciais para os testes do algoritmo de Restauração Inexata são os pontos sugeridos pela referência [15], sendo que estes pontos são viáveis com relação a l e a u. No caso do problema 15, usamos como ponto inicial um vetor cujas componentes são valores aleatórios entre 0 e 1. O número máximo de iterações é 500.

Como existem 4 situações diferentes onde o Algoritmo 2.1 pode parar, introduzimos a variável **CP** que indica o critério de parada do algoritmo e pode assumir 4 valores para cada uma destas situações, mostrados na Tabela 4.2.

problema	$f(x^*)$
1	0
2	4.0930
3	-3.4560
4	9.6172×10^2
5	5.1744×10^3
6	2.4151×10^{-1}
7	7.8777×10^{-2}
8	5.395×10^{-2}
9	8.9276×10^3
10	5.0550×10^3
11	-4.7761×10^1
15	0

Tabela 4.1: Mínimos locais das funções objetivo dos problemas teste.

CP stop	motivo
0	o algoritmo convergiu
1	$t < \varepsilon_t$ (4.15)
2	$\theta_k < \varepsilon_\theta$ (4.17)
3	o n^o máximo de iterações foi atingido

Tabela 4.2: Definição do CP stop.

4.3.1 Testes Considerando a Variação do k_{trial}^{max}

No Capítulo 2 destacamos que é interessante repetir os passos no espaço tangente, enquanto as condições da fase de restauração (4.5) e (4.6) forem válidas. Esta estratégia foi formalizada no Algoritmo 2.2, porém não determinamos o valor do parâmetro k_{trial}^{max}. Nos testes mostrados a seguir, variamos o valor de k_{trial}^{max} e definimos uma estratégia para escolher o valor deste parâmetro.

Nesta subseção organizamos os resultados numéricos em tabelas, e apresentamos alguns destes dados em gráficos para efeito comparativo. Também registramos o tempo de execução, em segundos, do Algoritmo 2.1 para os problemas teste.

Nas Tabelas 4.3, 4.4, 4.5, 4.6, 4.7, 4.8 e 4.9 vemos que os problemas mudam o comportamento de sua convergência conforme variamos o parâmetro k_{trial}^{max}.

O problema 1 converge com certa facilidade, mas o número de iterações é bastante sensível a mudanças no k_{trial}^{max}; quanto maior, menor é o número de iterações, exceto para o caso com $k_{trial}^{max} = 5$, onde observamos um aumento no número de iterações.

problema	iterações	$\left\|d_{tan}^k\right\|$	$\|C(x^*)\|$	$f(x^*)$	CP	tempo(s)
1	59	9.5638×10^{-5}	0	4.0102×10^{-6}	0	1.3728
2	6	4.8568×10^{-5}	2.7420×10^{-15}	4.0930	0	9.3601×10^{-2}
3	227	8.4462×10^{-5}	3.1402×10^{-16}	-3.4560	0	7.3320
4	5	7.2126×10^{-5}	5.0816×10^{-8}	9.6172×10^{2}	0	9.3601×10^{-2}
5	15	5.6560×10^{-3}	4.0990×10^{-13}	5.1265×10^{3}	2	8.7361×10^{-1}
6	67	9.6064×10^{-5}	5.0300×10^{-7}	2.5186×10^{-1}	0	1.6692
7	37	2.7209×10^{-4}	1.2230×10^{-12}	7.9001×10^{-2}	1	8.5801×10^{-1}
8	22	7.6347×10^{-5}	1.9860×10^{-15}	5.3950×10^{-2}	0	4.8360×10^{-1}
9	69	1.8322×10^{-6}	6.0054×10^{-7}	8.9721×10^{3}	0	8.3305
10	31	9.9441×10^{-5}	4.7915×10^{-10}	5.0550×10^{3}	0	1.4196
11	480	9.9298×10^{-5}	0	-4.7726×10^{1}	0	8.4085
15	20	4.3767×10^{-5}	1.7915×10^{-12}	4.5119×10^{-7}	0	1.7084×10^{2}

Tabela 4.3: resultados da Restauração Inexata com $k_{trial}^{max} = 0$.

problema	iterações	$\left\|d_{tan}^k\right\|$	$\|C(x^*)\|$	$f(x^*)$	CP	tempo(s)
1	29	4.5795×10^{-5}	3.2942×10^{-8}	4.0222×10^{-6}	0	9.0481×10^{-1}
2	7	8.6054×10^{-6}	4.1674×10^{-15}	4.0930	0	9.3601×10^{-2}
3	187	9.2182×10^{-5}	0	-3.4560	0	6.3336
4	5	2.6820×10^{-5}	5.5877×10^{-8}	9.6172×10^{2}	0	1.2480×10^{-1}
5	11	6.0955×10^{-3}	2.2737×10^{-13}	5.1265×10^{3}	2	5.4600×10^{-1}
6	42	9.6501×10^{-5}	4.8117×10^{-7}	2.5186×10^{-1}	0	1.1232
7	12	1.4688×10^{-5}	3.1907×10^{-7}	7.8996×10^{-2}	0	4.2120×10^{-1}
8	22	6.8790×10^{-5}	2.5511×10^{-15}	5.3950×10^{-2}	0	6.0840×10^{-1}
9	35	6.5973×10^{-5}	9.9282×10^{-7}	8.9721×10^{3}	0	8.8453
10	31	2.8903×10^{-5}	8.6441×10^{-9}	5.0550×10^{3}	0	1.8252
11	402	9.9254×10^{-5}	2.2204×10^{-16}	-4.7747×10^{1}	0	1.7940×10^{1}
15	19	4.2121×10^{-5}	2.6397×10^{-12}	4.4240×10^{-7}	0	2.1087×10^{2}

Tabela 4.4: resultados da Restauração Inexata com $k_{trial}^{max} = 1$.

problema	iterações	$\left\|d_{tan}^k\right\|$	$\|C(x^*)\|$	$f(x^*)$	CP	tempo(s)
1	19	9.3209×10^{-5}	2.2204×10^{-16}	3.9772×10^{-6}	0	8.2681×10^{-1}
2	5	9.8007×10^{-5}	3.9252×10^{-15}	4.0930	0	1.7160×10^{-1}
3	119	7.9320×10^{-5}	7.6919×10^{-16}	-3.4560	0	3.9936
4	5	5.5791×10^{-6}	5.5983×10^{-8}	9.6172×10^{2}	0	1.5600×10^{-1}
5	14	1.5866×10^{-3}	1.1369×10^{-13}	5.1265×10^{3}	2	7.4880×10^{-1}
6	28	4.9019×10^{-5}	4.0719×10^{-7}	2.5186×10^{-1}	0	1.0920
7	11	7.1314×10^{-5}	3.3514×10^{-7}	7.8995×10^{-2}	0	5.3040×10^{-1}
8	20	7.9915×10^{-5}	8.8818×10^{-16}	5.3950×10^{-2}	0	7.0200×10^{-1}
9	27	4.6940×10^{-5}	6.6851×10^{-7}	8.9721×10^{3}	0	1.2199×10^{1}
10	32	3.1839×10^{-5}	1.0063×10^{-8}	5.0550×10^{3}	0	2.1060
11	23	2.0783×10^{-7}	5.4028×10^{-8}	-4.7369×10^{1}	0	1.5756
15	21	1.1763×10^{-6}	1.3229×10^{-12}	-3.1423×10^{-10}	0	2.4129×10^{2}

Tabela 4.5: resultados da Restauração Inexata com $k_{trial}^{max} = 3$.

problema	iterações	$\left\|d_{tan}^k\right\|$	$\|C(x^*)\|$	$f(x^*)$	CP	tempo(s)
1	67	9.6147×10^{-5}	2.2204×10^{-16}	4.1445×10^{-6}	0	1.9032
2	5	4.2074×10^{-6}	4.5776×10^{-15}	4.0930	0	1.4040×10^{-1}
3	119	7.9320×10^{-5}	7.6919×10^{-16}	-3.4560	0	3.9624
4	5	5.5791×10^{-5}	5.5983×10^{-8}	9.6172×10^{2}	0	1.4040×10^{-1}
5	16	4.5795×10^{-3}	5.6843×10^{-13}	5.1265×10^{3}	2	1.3728
6	22	9.6927×10^{-5}	9.0103×10^{-7}	2.5190×10^{-1}	0	1.0452
7	11	4.1718×10^{-5}	5.2346×10^{-7}	7.8996×10^{-2}	0	5.1480
8	17	9.8945×10^{-5}	8.8818×10^{-16}	5.3950×10^{-2}	0	8.4241×10^{-1}
9	19	7.5178×10^{-6}	8.1157×10^{-7}	8.9721×10^{3}	0	8.6425
10	32	3.1839×10^{-5}	1.0063×10^{-8}	5.0550×10^{3}	0	1.9968
11	23	1.2804×10^{-7}	4.0163×10^{-8}	-4.7369×10^{1}	0	1.6380
15	23	1.8466×10^{-5}	1.8874×10^{-12}	8.42870×10^{-8}	0	3.3614×10^{2}

Tabela 4.6: resultados da Restauração Inexata com $k_{trial}^{max} = 5$.

problema	iterações	$\left\|d_{tan}^k\right\|$	$\|C(x^*)\|$	$f(x^*)$	CP	tempo(s)
1	27	8.7768×10^{-5}	0	4.5740×10^{-6}	0	1.0608
2	4	3.5939×10^{-5}	5.1518×10^{-15}	4.0930	0	1.7160×10^{-1}
3	119	7.9320×10^{-5}	7.6919×10^{-16}	-3.4560	0	3.7752
4	5	5.5791×10^{-5}	5.5983×10^{-8}	9.6172×10^{2}	0	1.5600×10^{-1}
5	13	4.6054×10^{-3}	0	5.1265×10^{3}	2	1.3104
6	28	8.8287×10^{-5}	6.0384×10^{-7}	2.5188×10^{-1}	0	1.4196
7	12	7.0119×10^{-5}	3.9540×10^{-7}	7.8994×10^{-2}	0	6.2400×10^{-1}
8	16	8.1238×10^{-5}	4.4409×10^{-16}	5.3950×10^{-2}	0	8.1121×10^{-1}
9	15	4.0916×10^{-6}	9.0972×10^{-7}	8.9721×10^{3}	0	7.3632
10	32	3.1839×10^{-5}	1.0063×10^{-8}	5.0550×10^{3}	0	1.7784
11	23	1.5580×10^{-7}	4.7007×10^{-8}	-4.7369×10^{1}	0	1.6224
15	22	5.9391×10^{-5}	3.8065×10^{-12}	8.5791×10^{-7}	0	3.6186×10^{2}

Tabela 4.7: resultados da Restauração Inexata com $k_{trial}^{max} = 7$.

problema	iterações	$\left\|d_{tan}^k\right\|$	$\|C(x^*)\|$	$f(x^*)$	CP	tempo(s)
1	16	6.1674×10^{-5}	3.3473×10^{-13}	4.4653×10^{-6}	0	9.6721×10^{-1}
2	4	3.3230×10^{-6}	6.7796×10^{-15}	4.0930	0	2.3400×10^{-1}
3	119	7.9320×10^{-5}	7.6919×10^{-16}	-3.4560	0	3.9000
4	5	5.5791×10^{-5}	5.5983×10^{-8}	9.6172×10^{2}	0	1.2480×10^{-1}
5	8	3.2667×10^{-5}	6.7214×10^{-10}	5.1265×10^{3}	0	1.3572
6	24	9.3206×10^{-5}	4.7198×10^{-7}	2.5190×10^{-1}	0	1.2636
7	11	4.1565×10^{-5}	6.1401×10^{-7}	7.8996×10^{-2}	0	8.7361×10^{-1}
8	14	8.2970×10^{-5}	1.8310×10^{-15}	5.3950×10^{-2}	0	1.0452
9	21	6.1444×10^{-6}	1.5002×10^{-7}	8.9721×10^{3}	0	1.2574×10^{1}
10	32	3.1839×10^{-5}	1.0063×10^{-8}	5.0550×10^{3}	0	1.9188
11	23	1.5580×10^{-7}	4.7007×10^{-8}	-4.7369×10^{1}	0	1.6068
15	27	7.7188×10^{-7}	2.2871×10^{-14}	2.2692×10^{-10}	0	4.6994×10^{2}

Tabela 4.8: resultados da Restauração Inexata com $k_{trial}^{max} = 10$.

problema	iterações	$\left\|d_{tan}^k\right\|$	$\|C(x^*)\|$	$f(x^*)$	CP	tempo(s)
1	9	9.8499×10^{-5}	3.0677×10^{-8}	4.7872×10^{-6}	0	2.9328
2	2	5.3597×10^{-5}	7.7127×10^{-15}	4.093	0	2.0280×10^{-1}
3	119	7.9320×10^{-5}	7.6919×10^{-16}	-3.4560	0	3.9000
4	5	5.5791×10^{-5}	5.5983×10^{-8}	9.6172×10^{2}	0	1.2480×10^{-1}
5	6	5.7809×10^{-5}	9.4842×10^{-8}	5.1265×10^{3}	0	7.0044
6	64	9.5356×10^{-5}	4.9833×10^{-7}	2.5186×10^{-1}	0	5.0856
7	12	5.4560×10^{-5}	2.0973×10^{-7}	7.8996×10^{-2}	0	2.9328
8	5	9.1823×10^{-5}	6.0361×10^{-13}	5.3950×10^{-2}	0	1.2636
9	26	7.2321×10^{-5}	4.0311×10^{-7}	8.9721×10^{3}	0	3.7674×10^{1}
10	32	3.1839×10^{-5}	1.0063×10^{-8}	5.0550×10^{3}	0	1.9500
11	23	1.5580×10^{-7}	4.7007×10^{-8}	-4.7369×10^{1}	0	1.5756
15	9	6.6525×10^{-5}	1.1210×10^{-7}	-5.4942×10^{-5}	0	6.5948×10^{2}

Tabela 4.9: resultados da Restauração Inexata com $k_{trial}^{max} = 100$.

O problema 5 mostra dificuldades na definição de θ_k para satisfazer a condição do passo 6 do Algoritmo 2.1, apesar de estarem próximos da solução.

O problema 11 sempre se mostrou insensível ao uso do passo espectral η_k, mas encontra uma solução um pouco maior que a solução original quando usamos k_{trial}^{max} igual a 3, 5, 7, 10 e 100.

Os outros problemas teste tiveram variações no seu comportamento conforme a variação de k_{trial}^{max}, mas sempre convergiram, assim como o problema 1.

Para os problemas que foram bem sucedidos observamos que no cálculo do parâmetro espectral (2.17) a condição $u^T s^k > 0$ foi sempre satisfeita a partir de uma certa iteração, mostrando que a estratégia do passo espectral está sendo bem sucedida, ou seja, que estávamos em uma região onde a função possui curvatura positiva ao longo das direções geradas, onde provavelmente existe um mínimo local. Então resolvemos alterar k_{trial}^{max} usando o valor de η_k. Conforme já foi discutido, η_k é uma medida de aproximação da inversa da matriz Hessiana, por isso este parâmetro nos dá informação sobre a curvatura de f ao longo da direção s^k. Assim sendo definimos

$$k_{trial}^{max} = round(\eta_k), \tag{4.22}$$

onde *round* é o arredondamento do valor de η_k. Além disso o cálculo de η_k usa tanto os valores de y^k, obtidos na fase de restauração, quanto os de z^k obtidos na fase de otimalidade. Desta forma η_k também nos dá informação sobre a convexidade do conjunto de restrições $C(x) = 0$. Usando esta estratégia obtivemos os resultados apresentados

na Tabela 4.10.

prob	iterações	$\|d^k_{tan}\|$	$\|C(x^*)\|$	$f(x^*)$	CP	tempo(s)	η_k^{max}
1	47	8.5500×10^{-5}	2.2204×10^{-16}	4.1117×10^{-6}	0	1.2792	10.431
2	6	4.8568×10^{-5}	2.7420×10^{-15}	4.0930	0	9.3601×10^{-2}	0.57613
3	187	9.2182×10^{-5}	0	-3.4560	0	7.7064	1×10^3
4	5	2.6820×10^{-5}	5.5877×10^{-8}	9.6172×10^2	0	1.5600×10^{-1}	0.66667
5	6	8.0561×10^{-5}	8.7878×10^{-9}	5.1265×10^3	0	6.3336	500.47
6	66	9.7843×10^{-5}	5.1587×10^{-7}	2.5186×10^{-1}	0	2.0592	20.221
7	12	2.7378×10^{-5}	3.8398×10^{-7}	7.8996×10^{-2}	0	5.1480×10^{-1}	2.5
8	22	6.8790×10^{-5}	2.5511×10^{-15}	5.3950×10^{-3}	0	7.3320×10^{-1}	20.568
9	35	6.5973×10^{-5}	9.9282×10^{-7}	8.9721×10^3	0	9.1105	0.66667
10	27	3.4659×10^{-5}	1.2105×10^{-8}	5.0550×10^3	0	1.9032	0.66667
11	480	9.9298×10^{-5}	0	-4.7726×10^1	0	1.4539×10^1	0.3
15	29	7.1537×10^{-6}	4.3294×10^{-12}	1.0629×10^{-8}	0	2.4831×10^2	0.15218

Tabela 4.10: Resultado da Restauração Inexata com $k^{max}_{trial} = round(\eta_k)$.

Nesta tabela, podemos observar que o uso de η_k para determinar k^{max}_{trial} possibilitou a convergência de todos os problemas e nenhuma **CP** foi ativada (indicamos o valor máximo de η_k por η_k^{max}). Outra vantagem no uso (4.10) é o baixo custo computacional para lidar com o problema da convergência do Algoritmo 2.1, porém não há nenhum respaldo teórico que garanta a eficiência desta estratégia. No entanto, como foi visto aqui, pode melhorar os resultados. Na Tabela 4.10 verificamos que o valor η_k^{max} para os dois últimos problemas está abaixo de 0.5, logo o valor de k^{max}_{trial}, pela equação (4.22), é zero, ou seja, o recurso de determinar o número máximo de iterações do Algoritmo 2.2 como o arredondamento do valor de η_k não foi acionado. Por isso definimos $k^{max}_{trial} = \max(10, round(\eta_k))$ para que haja um valor mínimo de iterações do Algoritmo 4.10. Após esta modificação obtemos novos resultados exibidos na Tabela 4.11, onde indicamos o valor máximo de k^{max}_{trial} por $k^{max^2}_{trial}$.

Na Tabela 4.11 vemos que o valor mínimo de 10 para o valor de k^{trial}_{max} só não reduziu o tempo de execução do Algoritmo 2.1 para os problemas 2, 7, 8, 9 e 12 e não reduziu o número de iterações apenas para o problema 10, com relação à estratégia original de arredondar o valor de η_k. A maior diferença com relação à estratégia original é notada com relação ao problema 11, que teve o número de iterações reduzido de 480 para 23. Comparamos o tempo de execução do Algoritmo 2.1 e o número de iterações nos gráficos a seguir definindo como "1, 3, 5, 7, 10 e 100" como os testes feitos com o valor de k^{max}_{trial} fixo, *simples* com a estratégia onde definimos $k^{max}_{trial} = round(\eta_k)$ e *composto*

prob	iterações	$\left\|d_{tan}^k\right\|$	$\|C(x^*)\|$	$f(x^*)$	CP	tempo(s)	$k_{trial}^{max^2}$
1	16	6.1674×10^{-5}	3.3473×10^{-13}	4.4653×10^{-6}	0	1.2480	10
2	4	3.3230×10^{-6}	6.7796×10^{-15}	4.0930	0	2.3400×10^{-1}	1
3	119	7.9320×10^{-5}	7.6919×10^{-16}	-3.4560	0	4.9608	10^3
4	5	5.5791×10^{-5}	5.5983×10^{-8}	9.6172×10^{2}	0	1.5600×10^{-1}	1
5	6	8.0397×10^{-5}	4.7816×10^{-12}	5.1265×10^{3}	0	2.5272	502
6	24	9.3206×10^{-5}	4.7198×10^{-7}	2.5190×10^{-1}	0	1.7940	10
7	11	8.2970×10^{-5}	6.1401×10^{-7}	7.8996×10^{-2}	0	1.1700	10
8	14	6.1444×10^{-5}	1.8310×10^{-15}	5.3950×10^{-3}	0	1.4352	19
9	21	3.1839×10^{-5}	1.5002×10^{-7}	8.9721×10^{3}	0	1.3120×10^{1}	10
10	32	1.5580×10^{-5}	1.0063×10^{-8}	5.0550×10^{3}	0	1.7160	10
11	23	1.5580×10^{-7}	4.7007×10^{-8}	-4.7369×10^{1}	0	2.3712	10
15	16	8.4977×10^{-8}	4.4409×10^{-16}	8.1855×-12	0	4.1910×10^{2}	10

Tabela 4.11: Resultado da Restauração Inexata com $k_{trial}^{max} = max(10, round(\eta_k))$.

quando definimos $k_{trial}^{max} = max(10, round(\eta_k))$. Seguem os gráficos para cada um dos problemas teste:

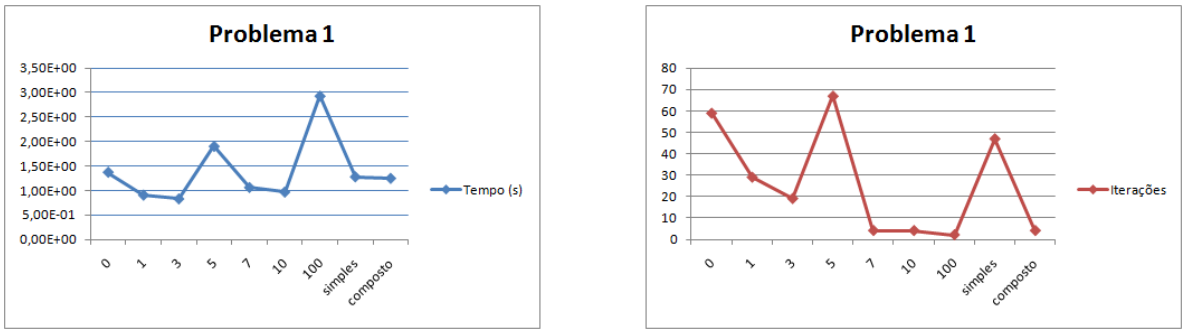

Figura 4.1: Gráficos de Tempo e Iterações por Estratégia para o Problema 1.

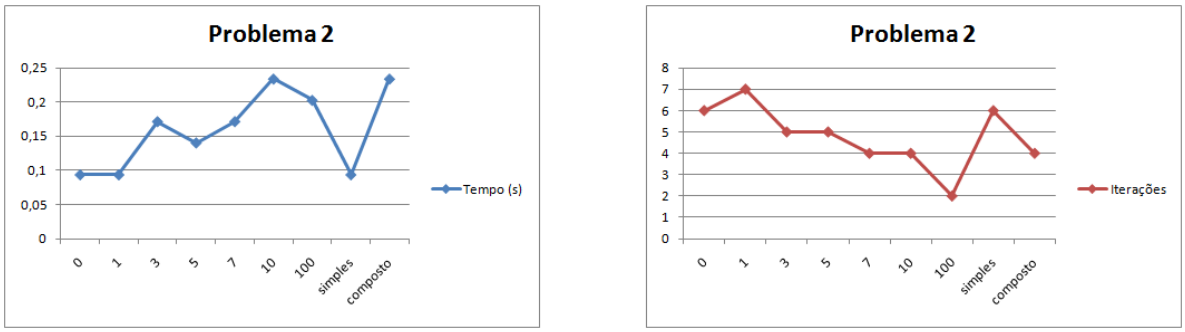

Figura 4.2: Gráficos de Tempo e Iterações por Estratégia para o Problema 2.

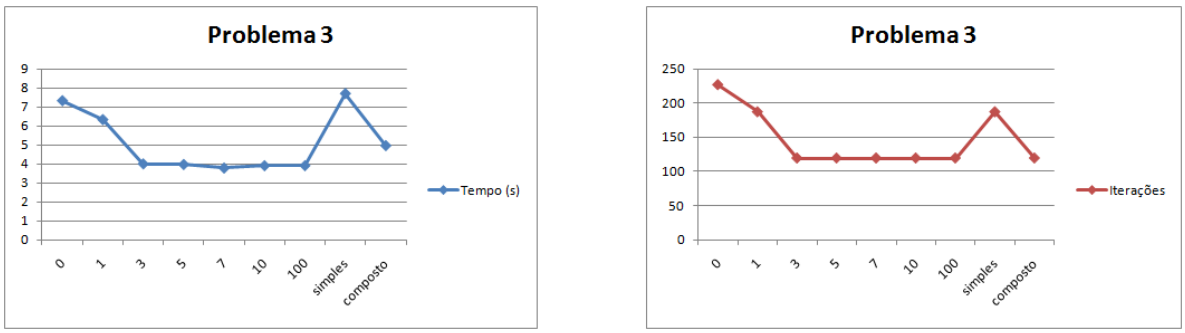

Figura 4.3: Gráficos de Tempo e Iterações por Estratégia para o Problema 3.

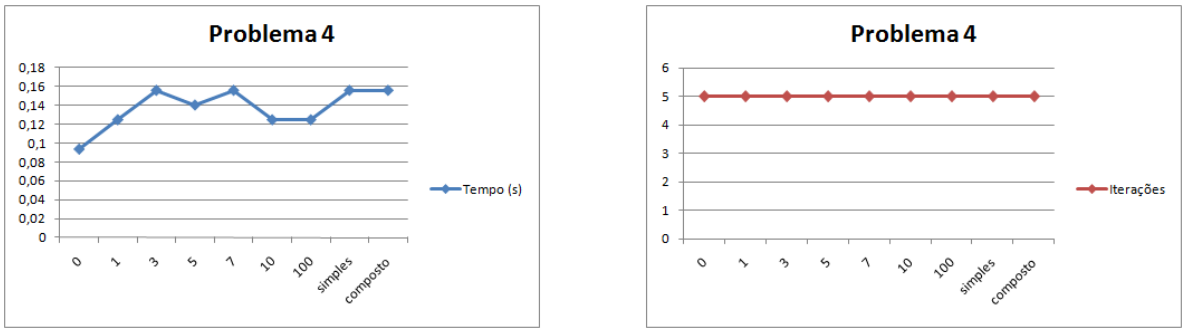

Figura 4.4: Gráficos de Tempo e Iterações por Estratégia para o Problema 4.

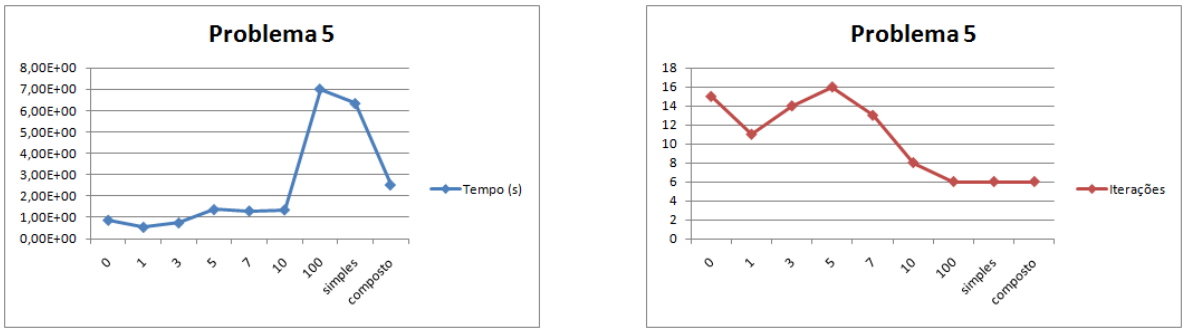

Figura 4.5: Gráficos de Tempo e Iterações por Estratégia para o Problema 5.

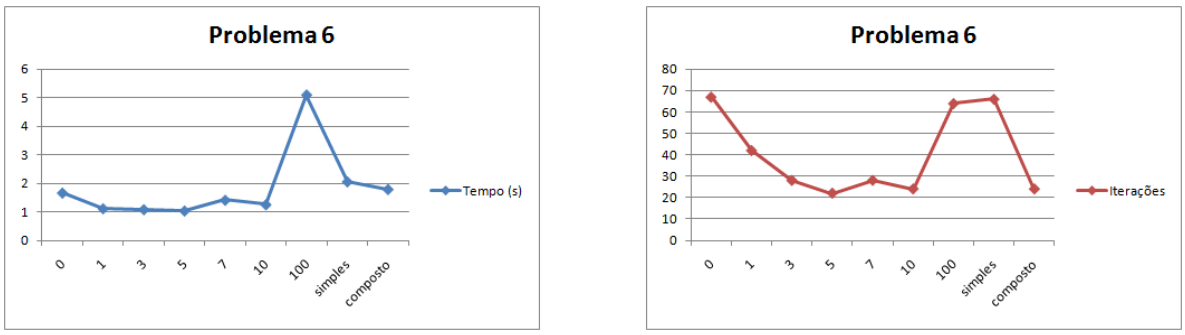

Figura 4.6: Gráficos de Tempo e Iterações por Estratégia para o Problema 6.

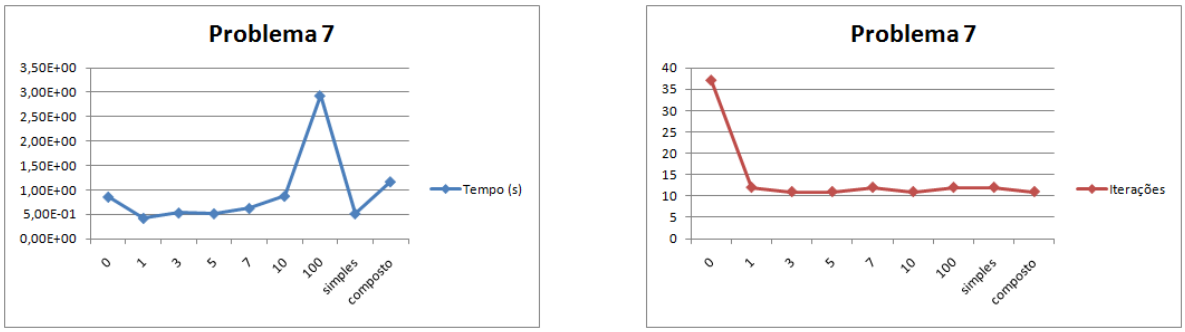

Figura 4.7: Gráficos de Tempo e Iterações por Estratégia para o Problema 7.

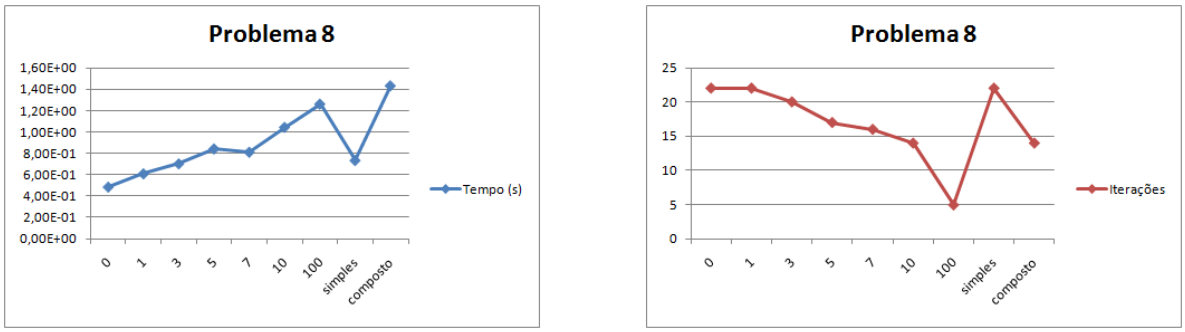

Figura 4.8: Gráficos de Tempo e Iterações por Estratégia para o Problema 8.

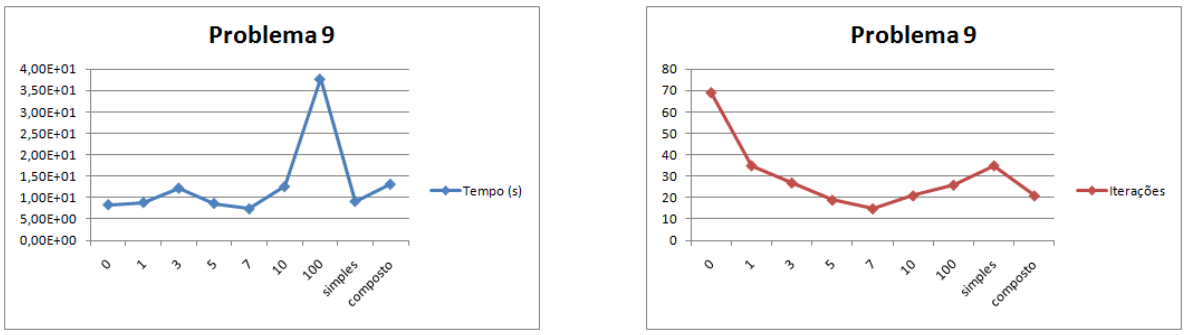

Figura 4.9: Gráficos de Tempo e Iterações por Estratégia para o Problema 9.

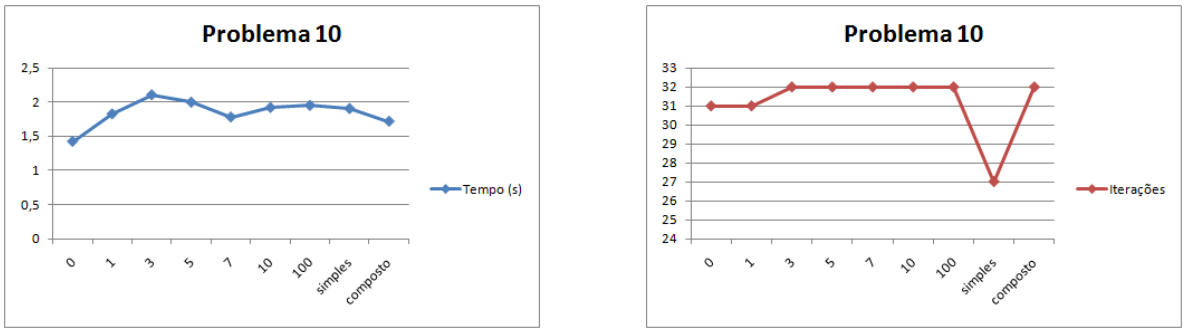

Figura 4.10: Gráficos de Tempo e Iterações por Estratégia para o Problema 10.

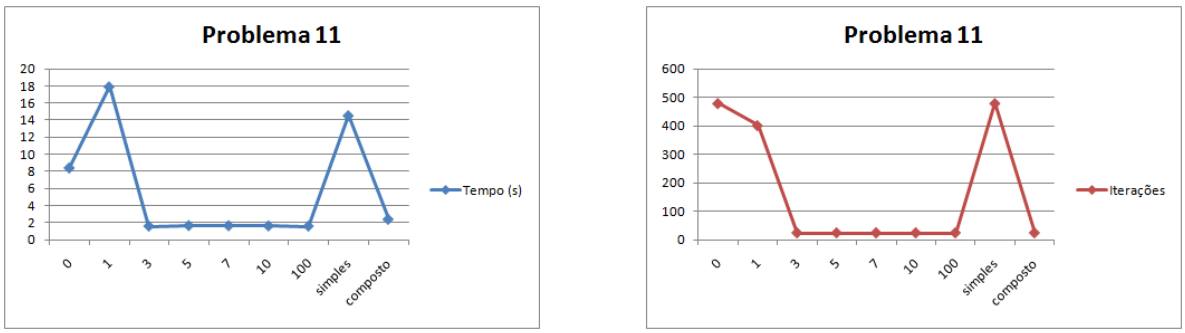

Figura 4.11: Gráficos de Tempo e Iterações por Estratégia para o Problema 11.

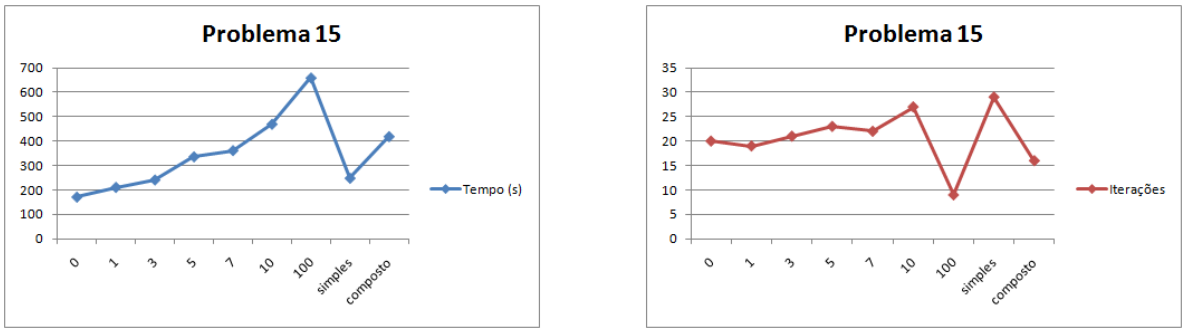

Figura 4.12: Gráficos de Tempo e Iterações por Estratégia para o Problema 15.

Conforme mostram as Figuras de 4.1 a 4.12, um número menor de iterações não significa necessariamente um tempo de execução menor, porém lembramos que o algoritmo de restauração inexata implementado neste trabalho, a cada iteração, repete o cálculo da direção de descida d_{tan}^k, enquanto as condições de viabilidade 2.3 e 2.4 forem válidas. Como é descrito no Algoritmo 2.2, o número de iterações é limitado ao parâmetro k_{max}^{trial}, isto nos leva a considerar iterações internas do Algoritmo 2.1 feitas durante a execução do Algoritmo 2.2 e a um número de iterações totais igual a soma das iterações do algoritmo principal e a do algoritmo interno. As tabelas e gráficos a seguir mostram a análise feita considerando estas iterações internas.

problema	iterações	iterações internas	iterações totais
1	59	0	59
2	6	0	6
3	227	0	227
4	5	0	5
5	15	0	15
6	67	0	67
7	37	0	37
8	22	0	22
9	69	0	69
10	31	0	31
11	480	0	480
15	24	0	24

Tabela 4.12: Iterações internas com $k_{trial}^{max} = 0$.

Mostramos os gráficos com as iterações totais comparando-os novamente com os gráficos de tempo de execução para cada um dos problemas teste.

problema	iterações	iterações internas	iterações totais
1	29	28	57
2	7	5	12
3	187	185	372
4	5	3	8
5	11	10	21
6	42	36	78
7	12	8	20
8	22	20	42
9	35	29	64
10	31	29	60
11	402	400	802
15	21	19	40

Tabela 4.13: Iterações internas com $k_{trial}^{max} = 1$.

problema	iterações	iterações internas	iterações totais
1	19	27	46
2	5	5	10
3	119	125	244
4	5	4	9
5	14	23	37
6	28	33	61
7	11	15	26
8	20	26	46
9	27	49	76
10	32	37	69
11	23	9	32
15	20	34	54

Tabela 4.14: Iterações internas com $k_{trial}^{max} = 3$.

problema	iterações	iterações internas	iterações totais
1	67	81	148
2	5	7	12
3	119	125	244
4	5	4	9
5	16	38	54
6	22	34	56
7	11	20	31
8	17	31	48
9	19	43	62
10	32	37	69
11	23	11	34
15	27	65	92

Tabela 4.15: Iterações internas com $k_{trial}^{max} = 5$.

problema	iterações	iterações internas	iterações totais
1	27	48	75
2	4	8	12
3	119	125	244
4	5	4	9
5	13	45	58
6	28	54	82
7	12	26	38
8	16	38	54
9	15	53	68
10	32	37	69
11	23	11	34
15	10	32	42

Tabela 4.16: Iterações internas com $k_{trial}^{max} = 7$.

problema	iterações	iterações internas	iterações totais
1	16	41	57
2	4	11	15
3	119	125	244
4	5	4	9
5	8	42	50
6	24	51	75
7	11	33	44
8	14	48	62
9	21	149	170
10	32	37	69
11	23	11	34
14	10	42	52

Tabela 4.17: Iterações internas com $k_{trial}^{max} = 10$.

problema	iterações	iterações internas	iterações totais
1	9	143	152
2	2	12	14
3	119	125	244
4	5	4	9
5	6	263	269
6	64	260	324
7	12	137	149
8	5	68	73
9	26	646	672
10	32	37	69
11	23	11	34
15	25	217	242

Tabela 4.18: Iterações internas com $k_{trial}^{max} = 100$.

problema	iterações	iterações internas	iterações totais
1	47	27	74
2	6	2	8
3	187	176	363
4	5	3	8
5	6	133	139
6	66	51	117
7	12	8	20
8	22	20	42
9	35	29	64
10	27	1	28
11	480	0	480
15	20	0	20

Tabela 4.19: Iterações internas com $k_{trial}^{max} = round(\eta_k)$.

problema	iterações	iterações internas	iterações totais
1	16	41	57
2	4	11	15
3	119	125	244
4	5	4	9
5	6	63	69
6	24	51	75
7	11	33	44
8	14	48	62
9	21	149	170
10	32	37	69
11	23	11	34
15	27	76	103

Tabela 4.20: Iterações internas com $k_{trial}^{max} = \max(10, round(\eta_k))$.

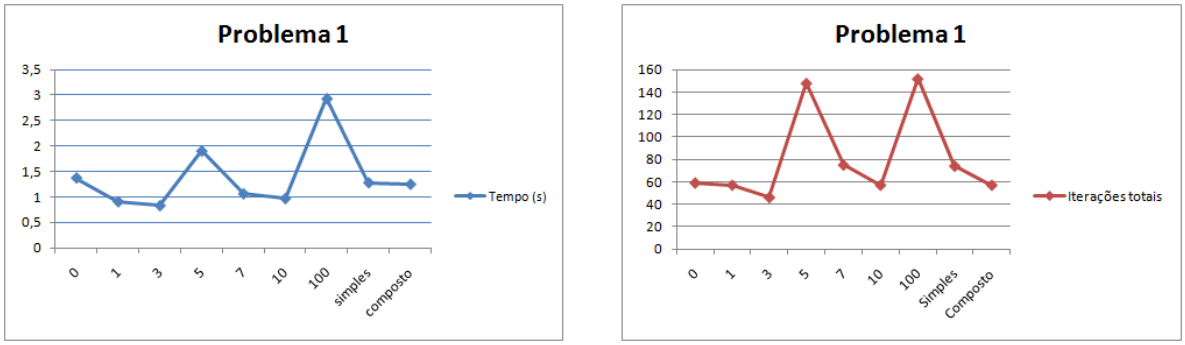

Figura 4.13: Gráficos de Tempo e Iterações Totais por Estratégia para o Problema 1.

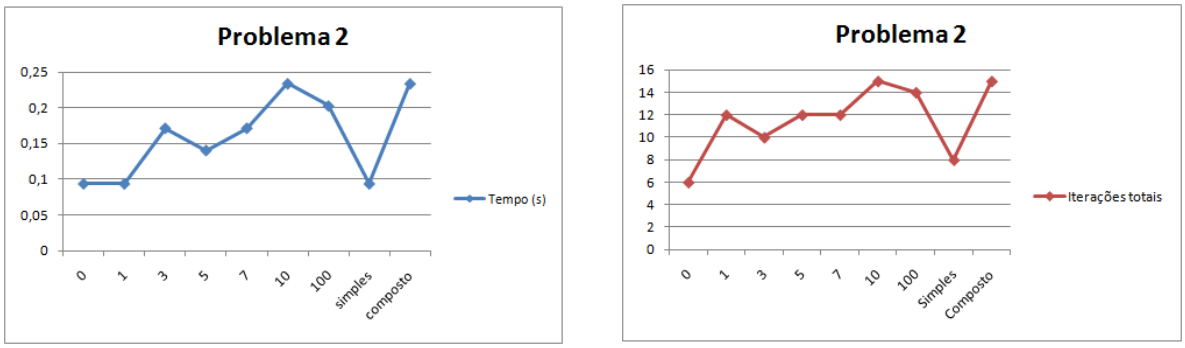

Figura 4.14: Gráficos de Tempo e Iterações Totais por Estratégia para o Problema 2.

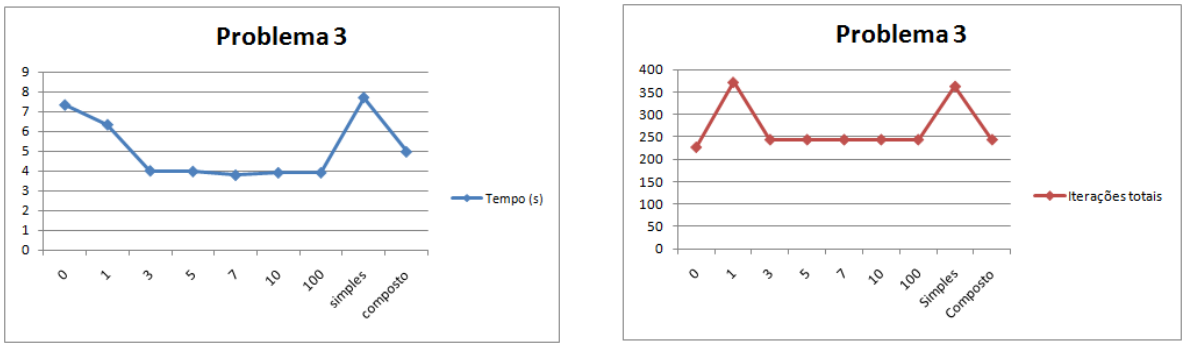

Figura 4.15: Gráficos de Tempo e Iterações Totais por Estratégia para o Problema 3.

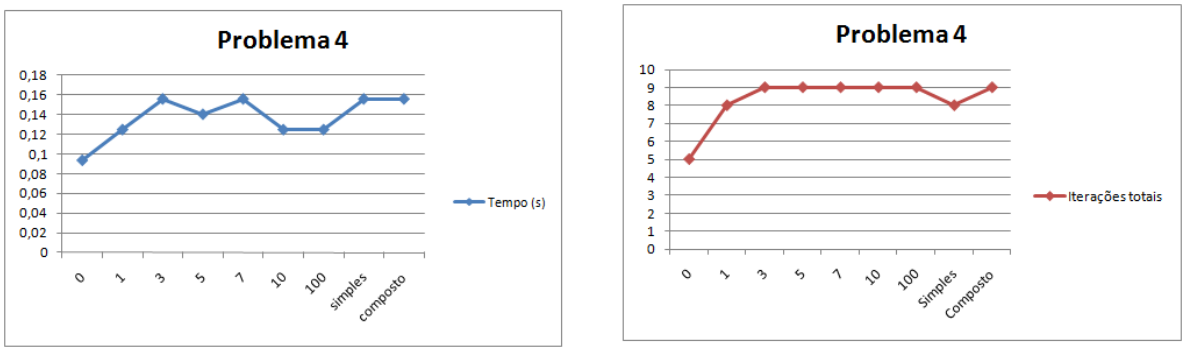

Figura 4.16: Gráficos de Tempo e Iterações Totais por Estratégia para o Problema 4.

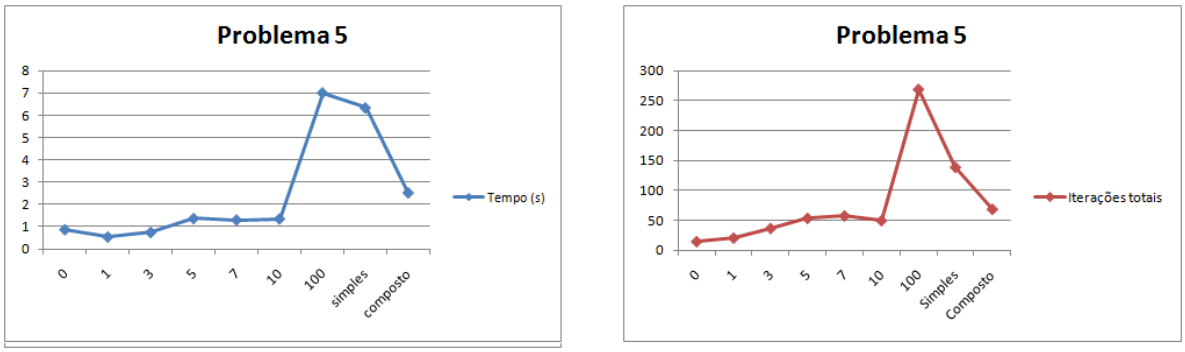

Figura 4.17: Gráficos de Tempo e Iterações Totais por Estratégia para o Problema 5.

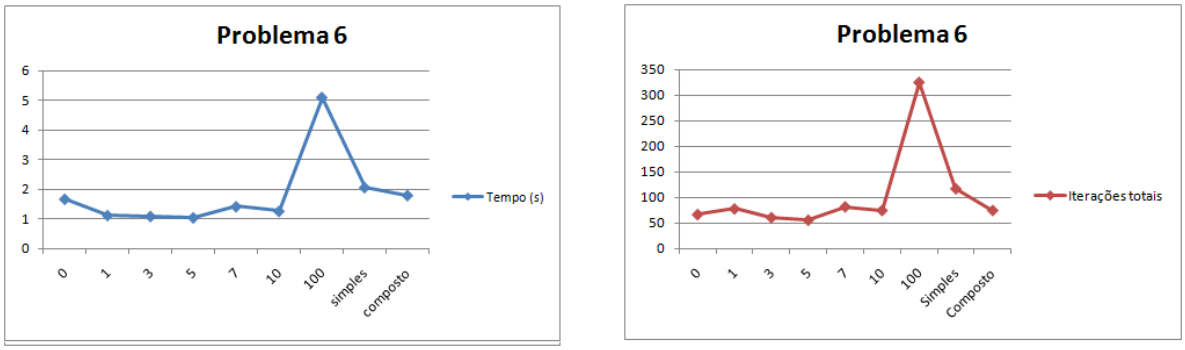

Figura 4.18: Gráficos de Tempo e Iterações Totais por Estratégia para o Problema 6.

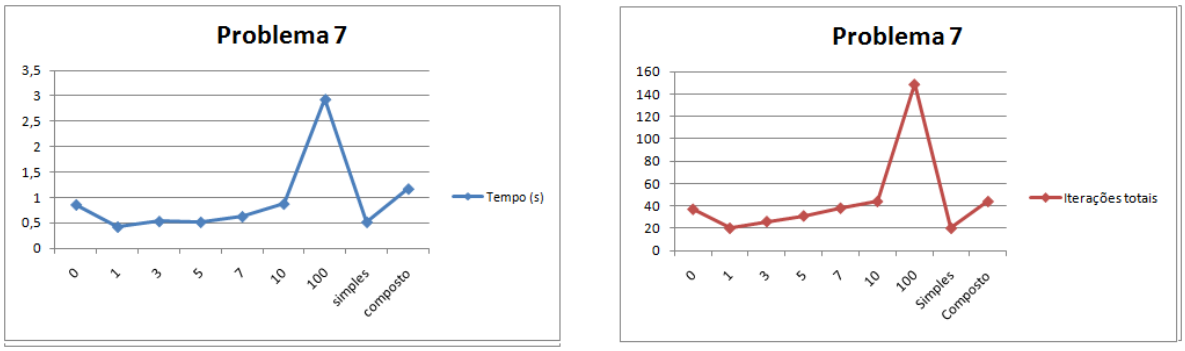

Figura 4.19: Gráficos de Tempo e Iterações Totais por Estratégia para o Problema 7.

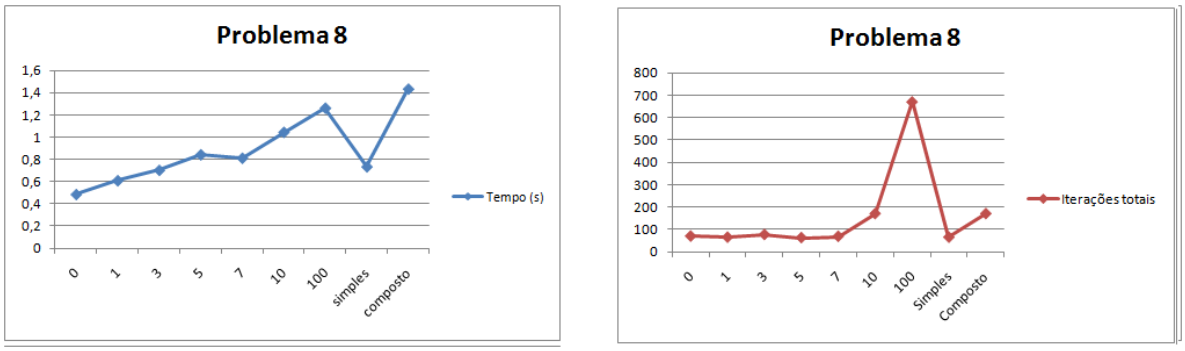

Figura 4.20: Gráficos de Tempo e Iterações Totais por Estratégia para o Problema 8.

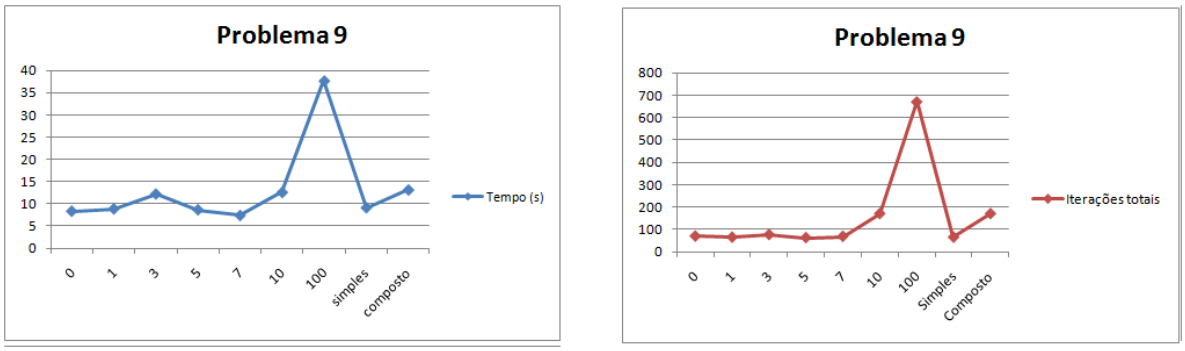

Figura 4.21: Gráficos de Tempo e Iterações Totais por Estratégia para o Problema 9.

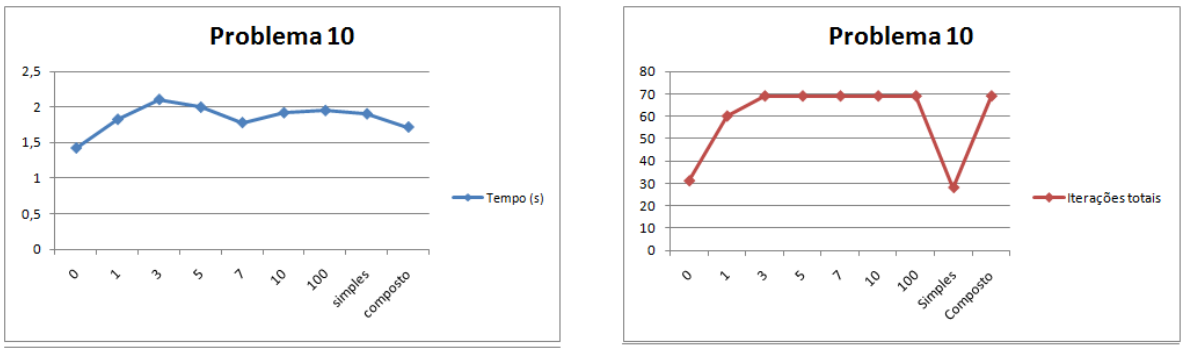

Figura 4.22: Gráficos de Tempo e Iterações Totais por Estratégia para o Problema 10.

Considerando as iterações internas, os gráficos de iterações totais e tempo de execução têm a mesma forma, exceto para o problema 10, onde o tempo de execução fica em torno de 1,5 s e 2,5 s, independentemente do número de iterações.

As estratégias que usam o arredondamento de η_k garantiram a convergência de todos os problemas teste, inclusive do problema 5 que não convergiu quando usamos k_{trial}^{max} fixo e igual a 0, 1, 3, 5 e 7. Por esta razão, é interessante considerar a estratégia de arredondamento do valor de η_k como, pelo menos, uma maneira de escolher um valor fixo para k_{trial}^{max}.

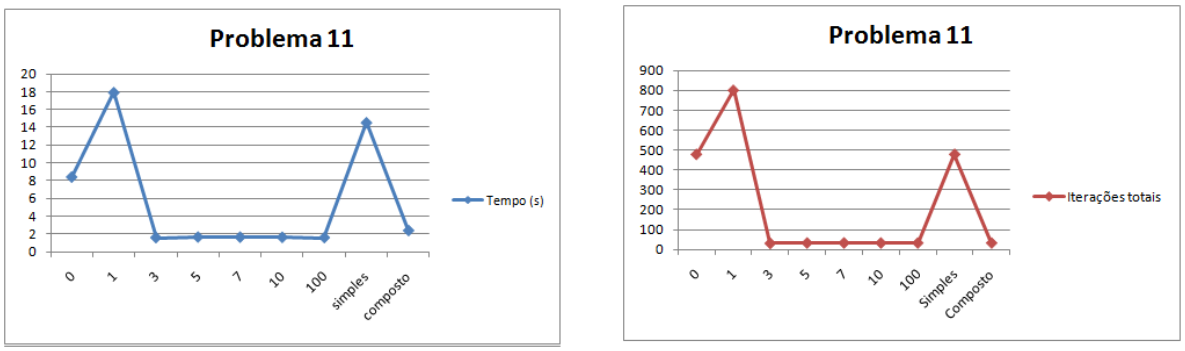

Figura 4.23: Gráficos de Tempo e Iterações Totais por Estratégia para o Problema 11.

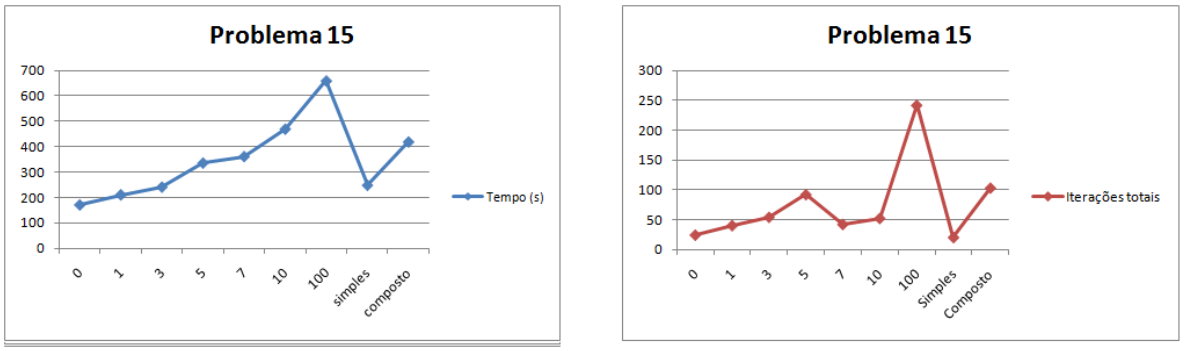

Figura 4.24: Gráficos de Tempo e Iterações Totais por Estratégia para o Problema 15.

Capítulo 5

Conclusão

O uso do algoritmo para sistemas não lineares, apresentado no Capítulo 3, na fase de restauração do método de Restauração Inexata se mostrou eficiente e por isso é uma boa alternativa a outros métodos como SGRA (*sequential gradient restoration Algorithm* - Referências [20] e [21]), para realizar a fase de restauração da viabilidade. Isso se deve ao fato do algoritmo para sistemas não lineares utilizar a estratégia de região de confiança, que por sua vez utiliza a matriz afim escala baseada nas condições de otimalidade do problema de minimizar a norma-2 do sistema não linear sujeito a restrições de canalização para reescalar o problema. Por permitir passos maiores em direção à solução, esta estratégia dá maior rapidez e estabilidade ao método.

Apesar do sucesso nos resultados, eles foram conseguidos através do ajuste de vários parâmetros do algoritmo de Restauração Inexata, principalmente com relação ao passo espectral. O cálculo deste valor foi determinante para a convergência dos testes e o seu valor inicial se mostrou ainda mais relevante sobre este aspecto. Por isso seria importante encontrar uma generalização para o valor inicial de η_k. A proposta apresentada no Capítulo 4 se mostrou eficiente para este conjunto de problemas teste, mas não sabemos como se comportaria para outros problemas. De qualquer forma, o uso de η_k ainda se mantém como uma alternativa atraente com relação ao cálculo da matriz Hessiana, devido ao seu baixo custo computacional.

A ideia de repetir os passos tangentes enquanto as condições de restauração permanecem satisfeitas diminuiu o número de iterações, mas não necessariamente o tempo de execução, do Algoritmo 2.1, conforme foi mostrado nos gráficos do Capítulo 4. Não temos como determinar um número ideal para a variável k_{trial}^{max} que limita o número de repetições, no entanto a estratégia de usar o arredondamento de η_k nos dá uma boa aproximação inicial para k_{trial}^{max}. Além disso, a estratégia de arredondamento do parâmetro espectral deu robustez ao Algoritmo 2.1. Todos os problemas teste conver-

giram para uma solução quando esta estratégia foi usada, inclusive o Problema 5, que teve sua execução interrompida para valores fixos de k_{trial}^{max}, devido ao rápido decréscimo do parâmetro θ.

Portanto o método de Restauração Inexata, apesar da grande sensibilidade com a variação de seus parâmetros, mostrou-se eficiente, com o uso do método para resolver sistemas não lineares com restrições de canalização na fase de restauração.

Capítulo 6

Apêndice dos Problemas Teste

Apresentamos aqui um apêndice com os problemas teste em sua forma original, conforme pode se verificar em [15], [11] e [14].

6.1 Problemas Comuns aos Algoritmos 2.1 e 3.3

6.1.1 Problema 1:

Fonte: [15] TP46.

$$\begin{array}{ll} \min & (x_1 - x_2)^2 + (x_3 - 1)^2 + (x_4 - 1)^4 + (x_5 - 1)^6 \\ \text{s. a} & x_1^2 x_4 + \sin(x_4 - x_5) - 1 = 0 \\ & -x_2 + x_3^4 x_4^2 - 2 = 0 \\ & -\infty \leq x \leq \infty \end{array} \tag{6.1}$$

ponto inicial,

$$x^{0^T} = \left(0.5\sqrt{2}, 1.7500, 0.5000, 2.0000, 2.0000\right),$$

solução,

$$x^{*^T} = (1, 1, 1, 1, 1)\,, \;\; f(x^*) = 0.$$

6.2 Problemas Exclusivos do Algoritmo 2.1

6.2.1 Problema 2:

Fonte: [15] TP53.

$$\begin{array}{ll} \min & (x_1 - x_2)^2 + (x_2 + x_3 - 2)^2 + (x_4 - 1)^2 + (x_5 - 1)^2 \\ \text{s. a} & x_1 + 3x_2 = 0 \\ & x_3 + x_4 - 2x_5 = 0 \\ & x_2 - x_5 = 0 \\ & -10 \leq x \leq -1 \end{array} \tag{6.2}$$

ponto inicial,

$$x^{0^T} = (1, 1, 1, 1, 1)\,,$$

solução,

$$x^{*^T} = (-33/43, 11/43, 27/43, -5/43, 11/43)\,,\ f(x^*) = 176/43.$$

6.2.2 Problema 3:

Fonte: [15] TP56.

$$\begin{array}{ll} \min & -x_1 x_2 x_3 \\ \text{s. a} & x_1 - 4.2\sin(x_4)^2 = 0 \\ & x_2 - 4.2\sin(x_5)^2 = 0 \\ & x_3 - 4.2\sin(x_4)^2 = 0 \\ & x_1 + 2x_2 + x_3 - 7.2\sin(x_4)^2 = 0 \\ & -\infty \leq x \leq \infty \end{array} \tag{6.3}$$

ponto inicial,

$$\begin{aligned} a &= \sin(\sqrt{(}1/4.2)); \\ b &= \sin(\sqrt{(}5/7.2)); \\ x^{0^T} &= (1, 1, 1, a, a, a, b)\,, \end{aligned}$$

solução,

$$x^{*^T} = \left(2.4, 1.2, 1.2, \sin\sqrt{4/7}, 1\sin\sqrt{2/7}, 2\tan 1\right),\ f(x^*) = -3.456.$$

6.2.3 Problema 4:

Fonte: [15] TP63.

$$\begin{array}{ll} \min & 10^3 - x_1^2 - 2x_2 - x_3^2 - x_1x_2 - x_1x_3 \\ \text{s. a} & 8x_1 + 14x_2 + 7x_3 - 56 = 0 \\ & x_1^2 + x_2^2 + x_3^2 - 25 = 0 \\ & 0 \leq x \leq \infty \end{array} \tag{6.4}$$

ponto inicial,

$$x^{0^T} = 2\,(1, 1, 1)\,,$$

solução,

$$x^{*^T} = (3.51211841492, 0.216988174172, 3.55217403459)\,,\ f(x^*) = 961.715172127.$$

6.2.4 Problema 5:

Fonte: [15] TP75.

$$\begin{array}{ll} \min & 3x_1 - 10^{-6}x_1^3 + 2x_2 + \dfrac{2 \times 10^{-6}}{3x_2^3} \\ \text{s. a} & 10^3\,[\sin(-x_3 - 0.25)] + \sin(-x_4 - 0.25) + 894.8 - x_1 = 0 \\ & 10^3\,[\sin(x_3 - 0.25)] + \sin(x_3 - x_4 - 0.25) + 894.8 - x_2 = 0 \\ & 10^3\,[\sin(x_4 - 0.25)] + \sin(x_4 - x_3 - 0.25) + 1.2948 \times 10^3 = 0 \\ & 0 \leq x \leq 1200 \\ & 0 \leq x \leq 1200 \\ & -0.48 \leq x \leq 0.48 \\ & -0.48 \leq x \leq 0.48 \end{array} \tag{6.5}$$

ponto inicial,

$$x^{0^T} = (0, 0, 0, 0)\,,$$

solução,

$$x^{*^T} = (776.159220293, 0.925, 0.19493919, 0.0511087936804,)\,,\ f(x^*) = 5174.41288686.$$

6.2.5 Problema 6:

Fonte: [15] TP77.

$$\begin{array}{ll} \min & (x_1-1)^2+(x_1-x_2)^2+(x_3-1)^2+(x_4-1)^4+(x_5-1)^6 \\ \text{s. a} & x_1^2x_4+\sin(x_4-x_5)-2\sqrt{(}2)=0 \\ & x_2+x_3^4x_4^2-8-\sqrt{(}2)=0 \\ & -\infty \leq x \leq \infty \end{array} \tag{6.6}$$

ponto inicial,

$$x^{0^T} = 5\,(1,1,1,1,1)\,,$$

solução,

$$x^{*^T} = (1.16617219726, 1.18211138813, 1.38025704044, 1.38025704044, 1.50603627961, 0.610920257517)\,,$$

$$f(x^*) = 0.241505128786.$$

6.2.6 Problema 7:

Fonte: [15] TP79.

$$\begin{array}{ll} \min & (x_1-1)^2+(x_1-x_2)^2+(x_3-x_3)^2+(x_3-x_4)^4+(x_4-x_5)^4 \\ \text{s. a} & x_1+x_2^2+x_3^3-2-3\sqrt{(}2)=0 \\ & x_2-x_3^2+x_4^2+2-\sqrt{(}2)=0 \\ & x_1x_5-2=0 \\ & -\infty \leq x \leq \infty \end{array} \tag{6.7}$$

ponto inicial,

$$x^{0^T} = 2\,(1,1,1,1,1)\,,$$

solução,

$$x^{*^T} = (1.19112745626, 1.36260316492, 1.47281793150, 1.63501661894, 1.67908143619)\,,$$

$$f(x^*) = 0.0787768208538.$$

6.2.7 Problema 8:

Fonte: [15] TP81.

$$\begin{aligned} \min \quad & e^{(x_1x_2x_3x_4x_5)} - \frac{1}{2}\left(x_1^3 + x_2^3 + 1\right)^2 \\ \text{s. a} \quad & x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0 \\ & x_2x_3 - 5x_4x_5 = 0 \\ & x_1^3 + x_2^3 + 1 = 0 \\ & -2.3 \leq x_1 \leq 2.3 \\ & -2.3 \leq x_2 \leq 2.3 \\ & -3.2 \leq x_3 \leq 3.2 \\ & -3.2 \leq x_4 \leq 3.2 \\ & -3.2 \leq x_5 \leq 3.2 \end{aligned} \tag{6.8}$$

ponto inicial,

$$x^{0^T} = (-2, 2, 2, -1, -1)\,,$$

solução,

$$x^{*^T} = (-1.71714240091, 1.59570833592, 1.82724792592, -0.763647440817, -0.763638975604)\,,$$

$$f(x^*) = 0.0539498477749.$$

6.2.8 Problema 9:

Fonte: [15] TP87.

$A = 131.078;$
$B = 1.48477;$
$C = 0.90798;$
$D = cos(1.47588);$
$E = sin(1.47588);$

$$F_1 = \begin{cases} 30x_1 & \text{se } x_1 > 300 \\ 31x_1 & \text{caso contrário} \end{cases}$$

$$F_2 = \begin{cases} 28x_2 & \text{se } x_1 > 100 \\ 31x_2 & \text{caso contrário} \end{cases}$$

$$
\begin{array}{ll}
\min & F_1 + F_2 \\
\text{s. a} & -x_1 + 300 - x_3 x_4 / A cos(B - x_6) + C x_3^2 / AD = 0 \\
& -x_2 - x_3 x_4 / A cos(B + x_6) + C x_4^2 / AD \\
& x_5 - x_3 x_4 / A sin(B + x_6) + C x_4^2 / AE = 0 \\
& 0 \leq x_1 \leq 400 \\
& 0 \leq x_2 \leq 1000 \\
& 340 \leq x_3 \leq 420 \\
& 340 \leq x_4 \leq 420 \\
& -1000 \leq x_5 \leq 1000 \\
& 0 \leq x_5 \leq 0.5236
\end{array} \tag{6.9}
$$

ponto inicial,

$$x^{0^T} = 1 \times 10^3 \left(0.3900, 0.9000, 0.4195, 0.3405, 0.1982, 0.0005\right),$$

solução,

$$x^{*^T} = (107.811937779, 196.318606955, 373.830728516, 420.0, 21.3071293896, 0.153291953422),$$

$$f(x^*) = 8927.59773493.$$

6.2.9 Problema 10:

Fonte: [15] TP107.

$V1 = 48.4/50.176;$
$C = V1 sin(0.25);$
$D = V1 cos(0.25);$
$Y1 = sin(x_8);$
$Y2 = cos(x_8);$
$Y3 = sin(x_9);$
$Y4 = cos(x_9);$
$Y5 = sin(x_8 - x_9);$
$Y6 = cos(x_8 - x_9);$

$$
\begin{aligned}
\min \quad & 3 \times 10^3 x_1 + 10^3 x_1^3 + 2 \times 10^3 x_2 + (2 \times 10^3 x_2^3)/3 \qquad (6.10)\\
\text{s. a} \quad & 0.4 - x_1 + 2Cx_5^2 + x_5x_6(-DY1 - CY2) + x_5x_7(-DY3 - CY4) = 0\\
& 0.4 - x_2 + 2Cx_6^2 + x_5x_6(DY1 - CY2) + x_6x_7(DY5 - CY6) = 0\\
& 0.8 + 2Cx_7^2 + x_5x_7(DY3 - CY4) + x_6x_7(-DY5 - CY6) = 0\\
& 0.2 - x_3 + 2Dx_5^2 - x_5x_6(-CY1 + DY2) - x_5x_7(-CY3 + DY4) = 0\\
& 0.2 - x_4 + 2Dx_6^2 - x_5x_6(CY1 + DY2) - x_6x_7(CY5 + DY6) = 0\\
& -0.337 + 2Dx_7^2 - x_5x_7(CY3 + DY4) - x_6x_7(-CY5 + DY6) = 0\\
& 0 \leq x_1 \leq \infty\\
& 0 \leq x_2 \leq \infty\\
& -\infty \leq x_3 \leq \infty\\
& -\infty \leq x_4 \leq \infty\\
& 0.90909 \leq x_5 \leq 1.0909\\
& 0.90909 \leq x_6 \leq 1.0909\\
& 0.90909 \leq x_7 \leq 1.0909\\
& -\infty \leq x_8 \leq \infty\\
& -\infty \leq x_9 \leq \infty
\end{aligned}
$$

ponto inicial,

$$x^{0^T} = (-2, 2, 2, -1, -1)\,,$$

solução,

$$x^{*^T} = (0.667009506909, 1.02238816675, 0.228287932605, 0.184821729352, 1.09090000001,$$

$$1.09090000001, 1.06903593236, 0.106612642267, -0.338786658776)\,,$$

$$f(x^*) = 5055.01180339.$$

6.2.10 Problema 11:

Fonte: [15] TP111.

$T = 0$;
$C = [-6.089; -17.164; -34.054; -5.914; -24.721; -14.986; -24.1; -10.708; -26.662; -22.179]$;

$$\begin{aligned} \min \quad & \sum_{i=1}^{10} e^{x_i}\left(C(i)+x_i-\log\left(\sum_{i=1}^{10} e^{x_i}\right)\right) \\ \text{s. a} \quad & e^{x_1}+2e^{x_2}+2e^{x_3}+e^{x_6}+e^{x_1 0}-2;=0 \\ & e^{x_4}+2e^{x_5}+e^{x_6}+e^{x_7}-1=0 \\ & e^{x_3}+e^{x_7}+e^{x_8}+2e^{x_9}+e^{x_{10}}-1=0 \\ & -1\leq x\leq 100 \end{aligned} \tag{6.11}$$

ponto inicial,

$$x^{0^T}=(0.8000,0.8000,0.2000,0.2000,1.0454,1.0454,1.0454,0,0)\,,$$

solução,

$$x^{*^T}=-\,(3.20121253241,1.91205959435,0.244441308369,6.53748856532,0.723152425984,$$
$$7.26773826993,3.59671064233,4.01776873216,3.28746169619,2.3355818305)\,,$$
$$f(x^*)=-47.7610902637.$$

6.3 Problemas Exclusivos do Algoritmo 3.3

6.3.1 Problema 12:

Fonte: [8] Problema 2.

$n=300$;

Para i de 1 a $n/2$:

$$F_i(x)=\sqrt{i}\left(x_i+x_{n/2+i}-i\right)=0. \tag{6.12}$$

6.3.2 Problema 13:

Fonte: [8] Problema 4.

$n=300$;

Para i de 1 a $n/2$:

$$F_i(x)=(x_i+x_{n/2+i})^2-i=0. \tag{6.13}$$

6.3.3 Problema 14

É definido em [11] da seguinte maneira:

$$F(x_1, x_2) = x_2 - \frac{1}{100}x_1, \text{ com } -\infty \leq x_1 \leq \infty \text{ e } x_2 \geq 0. \tag{6.14}$$

6.4 Problemas Exclusivos do Algoritmo 2.1

6.4.1 Problema 15:

Problema de otimização global restrita retirado da referência [14]:

$$\begin{aligned} \min \quad & g(x) = -\left(\sqrt{n}\right)^n \prod_{i=1}^{n} x_i \\ s.a \quad & \sum_{i=1}^{n} x_i^2 - 1 = 0 \\ & 0 \leq x_i \leq 1 \\ & \text{para } i = 1, \ldots, n, \end{aligned} \tag{6.15}$$

ponto inicial,

$$x^{0^T} = \text{números aletórios entre 0 e 1}$$

solução,

$$x^* = \frac{1}{\sqrt{n}}(1, \ldots, 1), \ g(x^*) = 1.$$

Como o produtório é uma função muito sensível a variações e o fator $(\sqrt{n})^n$ cresce rapidamente, reformulamos a função objetivo da seguinte forma: $g(x)$ por $f(x) = -log(g(x))$. Ou seja,

$$f(x) = -\left(nlog(\sqrt{n}) + \sum_{i=1}^{n} log(x_i) \right).$$

A mudança nos dá uma função mais bem comportada, pois evita o rápido crescimento do termo $(\sqrt{n})^n$. O valor da função objetivo em x^* será $f(x^*) = 0$.

Referências Bibliográficas

[1] Barzilai, J. & Borwein, J. M., *Two point step size gradient method*, IEEE Trans. Automat. Control, Vol 21, pp. 141-148, 1976.

[2] Bazaraa, M. S., Sherali, H D. & Shetty, C. M., Nonlinear Programming Theroy and Algorithms, 2^a Edição, 1993.

[3] Bellavia, S., Macconi, M. & Morini, B., *An affine scaling trust-region approach to bound-constrained nonlinear systems*, Applied Numerical Mathematics, Vol. 44, pp. 257-280, 2003.

[4] Birgin, E. G., Martínez, J. M. & Raydan, M., *Nonmonotone spectral projected gradient methods on convex sets*, SIAM J. Optim., Vol. 10, 4, pp. 1196-1211, 2000.

[5] Birgin, E. G. & Martínez, J. M., *Local convergence of Inexact Restoration method and numerical experiments*, Journal of Optimization Theory and Applications 127 p. 229-247, 2005.

[6] Birgin, E. G., Martínez, J. M. & Raydan, M., *Algorithm 813: SPG - Software for convex constrained optimization*, ACM Trans. Math. Software, Vol. 27 , pp. 340-349, 2001.

[7] Coleman, T. F. & Li, Y., *An interior trust region approach for nonlinear minimization subject to bounds*, SIAM J. Optim. Vol. 6, pp. 418-445, 1996.

[8] Dan, N., Fukushima, M. & Yamashita, N., *Convergence properties of inexact Levenberg-Marquardt method under local error bound conditions*. Optimization Methods and Software, Vol. 17, pp. 605-626, 2002.

[9] Dennis, J. E. & Schnabel, R. B., Numerical methods for unconstrained optimization and nonlinear equations, Prentice Hall, Englewood Cliffs, NJ, 1983.

[10] Diniz-Ehrhardt, M. A., Gomes-Ruggiero, M. A., Martínez, J. M. & Santos S. A., *Augmented Lagrangian algorithms based on the spectral projected gradient method for solving nonlinear programming problems*, Journal of Optimization Theory and Applications, Vol. 123, 3, pp. 497-517, 2004.

[11] Francisco, J. B., Krejić, N. & Martínez, J. M., *An interior-point method for solving box-constrained underdetermined nonlinear systems*, Journal of Computational and Applied Mathematics, Vol. 177, pp. 67-88, 2005.

[12] Francisco, J. B. & Martínez, J. M., *Viabilidade em Programação Não-linear: Restauração e Aplicações* Tese de Doutorado, Brasil - Unicamp - Imecc, 2005.

[13] Gomes-Ruggiero, M. A., Martínez, J. M. & Santos, S. A. *Inexact Restoration and Spectral Projected Gradient Methods for Minimization with Nonconvex Constraints*, SIAM J. Sci. Comput., Vol. 31, No., pp. 1628-1652, 27 Fevereiro de 2009.

[14] Hedar, A., *Test Problems for Constrained Global Optimization*, http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page2613.htm

[15] Hock, W. & Schittkowski, K., Test Examples for Nonlinear Programming Codes, Springer Series Lectures Notes in Economics Mathematical Systems, 1981.

[16] Kanzow, C., Yamashita, N. & Fukushima, M., *Levenberg-Marquardt methods for constrained nonlinear equations with strong local convergence properties*, Journal of Computational and Applied Mathematics, Vol. 172, pp. 375-397, 2004.

[17] Lima, E. L., Álgebra linear, Coleção Matemática Universitária, IMPA 2004.

[18] Martínez, J. M. & Pilotta, E. A., *Inexact-Restoration Algorithm for Constrained Optimization*, Journal of Optimization Theory and Applications, Vol. 104, 1, pp. 135-163, 2000.

[19] Martínez, J. M. & Pilotta, E. A., *Inexact Restoration Methods for Nonlinear Programming: Advances and Perspectives*, Optimization and Control with Applications, Vol. 96, Part II, pp. 271-291, 2006.

[20] Miele, A., Levy, A. V. & Basapur, V. K., *Sequential Gradient Restoration Algorithm for Mathematical Programming Problems*, Journal of Optimization Theory and Applications, Vol. 7, pp. 450-472, 1971.

[21] Miele, A., Huang, H. Y. & Heideman, J. C., *Sequential Gradient Restoration Algorithm for the Minimization of Constrained Functions: Ordinary and Conjugate Gradient Version*, Journal of Optimization Theory and Applications, Vol. 4 pp. 213-246, 1969.

[22] Nocedal, J. & Wright, S. J., Numerical Optimization, Springer Series in Operation Researchs, 1999.

[23] Raydan, M., *The Barzilai & Borwein Gradient method for the large scale unconstrained minimization problem*, SIAM J. Optim., Vol. 7, pp. 26-33, Fevereiro 1997.

[24] Watkins, D. S., Fundamentals of Matrix Computations, 2^a Edição, Wiley 2002.