UNIVERSIDADE ESTADUAL DE CAMPINAS
Instituto de Matemática, Estatística
e Computação Científica
Departamento de Matemática Aplicada

Acelerando o método de Levenberg-Marquardt para a minimização da soma de quadrados de funções com restrições em caixa

Tese de Doutorado

Luiz Antônio da Silva Medeiros

Orientador: Prof. Dr. Francisco de Assis Magalhães Gomes Neto

Co-orientador: Prof. Dr. José Mario Martínez

Campinas
Outubro de 2008

ACELERANDO O MÉTODO DE LEVENBERG-MARQUARDT PARA A MINIMIZAÇÃO DA SOMA DE QUADRADOS DE FUNÇÕES COM RESTRIÇÕES DE CAIXA

Este exemplar corresponde à redação final da tese devidamente corrigida e defendida por Luiz Antônio da Silva Medeiros e aprovada pela comissão julgadora.

Campinas, 8 de outubro de 2008.

Prof. Dr. Francisco A. M. Gomes
Orientador.

Prof. Dr. José Mario Martínez
Co-orientador

Banca Examinadora:

Profª. Maria de Los Angeles Gonzáles Lima (Univ. Simón Bolívar, Venezuela)
Prof. Ernesto Julián Goldberg Birgin (IME–USP)
Profª. Márcia A. Gomes Ruggiero (IMECC–UNICAMP)
Profª. Sandra Augusta Santos (IMECC–UNICAMP)
Prof. Francisco A. M. Gomes (IMECC–UNICAMP)

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

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP
Bibliotecária: Maria Júlia Milani Rodrigues – CRB8a 211 6

Medeiros, Luiz Antônio da Silva

M467a Acelerando o método de Levenberg-Marquardt para minimização de soma de quadrados de funções com restrições em caixa / Luiz Antônio da Silva Medeiros -- Campinas, [S.P. :s.n.], 2008.

Orientador : Francisco de Assis Magalhães Gomes Neto ; José Mario Martínez

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

1. Mínimos quadrados. 2. Otimização matemática. 3. Levenberg-Marquardt. 4. Método do gradiente projetado. I. Gomes Neto, Francisco de Assis Magalhães. II. Martínez, José Mario. III. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. IV. Título.

Título em inglês: Accelerating the Levenberg-Marquardt method for the minimization of the square of functions with box constraints.

Palavras-chave em inglês (Keywords): 1. Least squares. 2. Mathematical optimization. 3. Levenberg-Marquardt. 4. Project gradient methods.

Área de concentração: Matemática Aplicada

Titulação: Doutor em Matemática Aplicada

Banca examinadora:

Prof. Dr. Francisco de Assis Magalhães Gomes Neto (IMECC-UNICAMP)
Prof. Dr. Ernesto J. Goldberg Birgin (IME-USP)
Profa. Dra. Maria de Los Angeles González Lima (Univ. Simón Bolivar)
Profa. Dra. Marcia Aparecida Gomes Ruggiero (IMECC-UNICAMP)
Profa. Dra. Sandra Augusta Santos (IMECC-UNICAMP)

Data da defesa: 08/10/2008

Programa de pós-graduação: Doutorado em Matemática Aplicada

Tese de Doutorado defendida em 08 de outubro de 2008 e aprovada

Pela Banca Examinadora composta pelos Profs. Drs.

Prof(a). Dr(a). FRANCISCO DE ASSIS MAGALHÃES GOMES NETO

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

Prof(a). Dr(a). MARIA DE LOS ANGELES GONZÁLES LIMA

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

Prof(a). Dr(a). ERNESTO JULIÁN GOLDBERG BIRGIN

Se você não mudar a direção,
terminará exatamente onde partiu.

Antigo provérbio Chinês

Agradecimentos

Dedico esta consquista a todos que me acompanharam e participaram direta ou indiretamente ao longo de todo o tempo da pesquisa.

À Deus por me dar forças e suprimento para lutar pela concretização desta vitória, e pela tranquilidade que me proporcionou, mesmo na adversidade, a concentração em prol desta realização.

À minha familia, em especial aos meus pais, meus irmãos, minha esposa e minha filha pelo apoio, incentivo e sobretudo o *amor* que me fez suportar os espinhos ao logo desta estrada.

À minha tia Ridênia Noberto Maia e família, pelo apoio dado ainda bem antes deste sonho ser lapidado. Em sua residência em Fortaleza-CE, fui muito bem acolhido e lá começei a esboçar os rascunhos para esta jornada. Sem esta hospitalidade, reconhecida e valorizada, dificilmente teria conseguido o Mestrado em Matemática, fortalecendo as bases para o Doutorado.

Aos meus amigos e companheiros, Paulo César, Odair, Kelly, Amauri, Marcelo, Ângelo, Sueli, Júlia Benavides, Nir Cohen, Licia, Marcelo Santos, Luziane, Benaia, Valéria e tantos outros que dividiram comigo os bons e maus momentos ao longo destes anos, apoiando de todas as formas para tornar o ambiente familiar e por outro lado, propício para os estudos.

Agradeço aos meus professores. Em especial, a Francisco Gomes (Chico) e a José Mario Martínez, que dedicaram, com paciência e astúcia, boas horas de seu valoroso tempo para me ensinar e me repassar os "detalhes" desse conhecimento construído e elaborado, que só a experiência pode revelar.

Por fim, e com muito respeito, quero agradecer a Universidade Regional do Cariri (URCA). Destacando a luta do Departamento de Matemática para a qualificação de seus professores, com o objetivo maior de poder contribuir com mais eficiência e qualidade, através do ensino e da pesquisa, para o desenvolvimento da Região do Cariri e da própria Matemática.

A todos vocês deixo o meu mais profundo agradecimento. E dedico a minha felicidade e esta conquista a cada um. Obrigado.

RESUMO

Neste trabalho, apresentamos um algoritmo iterativo para a minimização de somas de quadrados de funções suaves, com restrições de caixa. O algoritmo é fortemente inspirado no trabalho de Birgin e Martínez [4]. A diferença principal está na escolha da direção de busca e na introdução de uma nova técnica de aceleração, usada para atualizar o passo. A cada iteração, definimos uma face ativa e resolvemos, nessa face, um subproblema quadrático irrestrito através do método Levenberg-Marquardt (ver [26], [28] e [33]), obtendo uma direção de descida e uma aproximação x^+ para a solução do problema. Ainda usando apenas as variáveis livres, tentamos acelerar o método definindo uma nova aproximação x^a como combinação linear das últimas $p-1$ aproximações da solução e do vetor x^+. Os coeficientes desta combinação linear são calculados convenientemente através da resolução de um problema de Quadrados Mínimos com uma restrição de igualdade. O subproblema que determina o passo acelerado leva em conta as informações sobre a função objetivo nessas p soluções aproximadas. Como em [4], executamos uma busca linear ao longo da direção e usamos técnicas de projeção para adicionar novas restrições. Para deixar a face ativa, usamos a direção do gradiente espectral projetado [5]. Experimentos númericos são apresentados para confirmar a eficiência e robustez do novo algoritmo.

Palavras chave: minimização de somas de quadrados; método de Levenberg-Marquardt; métodos de restrições ativas; aceleração.

ABSTRACT

In this work, we present an active set algorithm for minimizing the sum of squares of smooth functions, with box constraints. The algorithm is highly inspired in the work of Birgin and Martínez [4]. The differences are concentrated on the chosen search direction and on the use of an acceleration technique to update the step. At each iteration, we define an active face and solve an unconstrained quadratic subproblem using the Levenberg-Marquardt method (see [26], [28] and [33]), obtaining a descent direction and an approximate solution x^+. Using only the free variables, we try to accelerate the method defining a new solution x^a as a linear combination of the last $p-1$ approximate solutions together with x^+. The coefficients of this linear combination are conveniently computed solving a constrained least squares problem that takes into account the objective function values of these p approximate solutions. Like in [4], we compute a line search and use projection techniques to add new constraints to the active set. The spectral projected gradient [5] is used to leave the current active face. Numerical experiments confirm that the algorithm is both efficient and robust.

Key words: sum of squares optimization; Levenberg-Marquardt method; active set methods, acceleration.

Sumário

Introdução **1**

1 Métodos para a resolução de problemas não lineares de quadrados mínimos **3**

1.1 Método de Newton 6

1.2 Método de Levenberg-Marquardt irrestrito 9

1.3 Escolha do parâmetro de Levenberg - Marquardt 16

2 Técnicas de aceleração **25**

2.1 DIIS: *Direct Inversion of the Iterative Subspace* 26

2.2 MDIIS: *Modified Direct Inversion of the Iterative Subspace* 27

3 Algoritmo para a minimização de somas de quadrados de funções em caixas **35**

3.1 Definições 36

3.2 Algoritmo interno à face 40

3.3 Implementação do algoritmo interno à face 53

3.4 Algoritmo principal 56

4 Experimentos numéricos **63**

4.1 Grupo de experimentos 1 63

4.2 Grupo de experimentos 2 66

4.2.1 O problema das esferas [E1] 66

4.2.2 O problema do Aeroporto [E2] 68

4.2.3 O problema do empacotamento de cilindros [E3] 69

4.3 Análise do desempenho dos algoritmos 70

4.3.1 Resultados para o grupo de experimentos 1 71

4.3.2 Resultados para o grupo de experimentos 2 77

5 Conclusão **81**

A O método de Levenberg-Marquardt e regiões de confiança **83**
A.1 Analisando o *caso difícil* 88

B Teorema da projeção **93**

C Perfil de desempenho **99**

Bibliografia **101**

Introdução

O problema de encontrar um minimizador de uma função, quase sempre não linear, suave (de classe $\mathcal{C}^2$) e com restrições de canalização é objeto de estudo de muitos pesquisadores, não apenas pela busca intensa do conhecimento científico, mas principalmente pela grande aplicabilidade desses métodos de otimização em aréas como tecnologia, economia, saúde etc, que estão em constante desenvolvimento, necessitando cada vez mais de novos e poderosos métodos para resolver problemas mais complexos.

Em [4], Birgin e Martínez apresentaram um algoritmo (GENCAN) para a minimização de uma função não linear com restrições de canalização que adota técnicas baseadas em conjuntos de restrições ativas, exigindo que, na iteração corrente, tais restrições sejam dadas pelas componentes da aproximação atual que coincidem com os respectivos limitantes inferior ou superior que definem a caixa. O algoritmo que apresentamos para minimização de funções suaves consistindo de soma de quadrados de funções com restrições de canalização é fortemente inspirado no algoritmo GENCAN, tendo por diferença fundamental a inclusão de uma nova técnica de aceleração para a correção da direção de busca dentro das faces e a escolha da direção Levenberg-Marquardt no lugar da direção de Newton truncada adotada em GENCAN.

Quando a função objetivo é uma soma de quadrados de funções, a direção de busca proposta por Lenvenberg-Marquardt, originalmente associada à resolução de problemas de quadrados mínimos (ver Levenberg [26] e Marquardt [28]), costuma ter um desempenho melhor do que a direção de Newton truncada adotada em GENCAN [4]. Entretanto, como o método Levenberg-Marquardt, ou simplesmente método LM, pode necessitar de um esforço computacional maior, já que resolve a cada iteração vários sistemas lineares para obter uma direção de descida suficiente, surge a necessidade de se desenvolver técnicas que possam acelerar este método.

Quase todas as técnicas propostas na literatura para acelerar o método LM se baseiam nas técnicas tradicionais de busca linear e na escolha de uma matriz de escalamento para melhorar o número de condição da matriz Hessiana (ver [40]) ou controlar o tamanho do passo para que a nova aproximação seja interior à caixa, como em Coleman [12]. Outras

estratégias consideradas para acelerar o método incluem uma reformulação da função objetivo, como em [44], e a utilização de técnicas de decomposições matriciais (ver [47] e [8]) para resolver o sistema de Levenberg-Marquardt.

Em [37], Pulay apresentou uma proposta de aceleração do algoritmo LM através de uma correção na direção de busca. Esta técnica, denominada **DIIS**, foi desenvolvida por volta de 1980 e é empregada até hoje, principalmente na área da *química molecular* e em *otimização geométrica* (ver, por exemplo, Farkas-Schlegel [20]). Ela considera que uma melhor aproximação da solução final pode ser construída como combinação linear das últimas p aproximações, supondo que os coeficientes desta combinação minimizam, no sentido dos quadrados mínimos, o vetor residual formado pela combinação dos passos anteriormente calculados. Existem também variantes dessa estratégia que usam as informações dos gradientes correspondentes às p aproximações consideradas em lugar dos respectivos passos corretores (vide Csaszar-Pulay [13] e Farkas-Schlegel [19]).

A nossa proposta de aceleração é inspirada na técnica *DIIS*. Entretanto, corrigimos a direção de busca usando um número limitado de valores funcionais computados em passos anteriores e na iteração corrente. Esta é uma forma barata de se calcular os coeficientes da combinação linear que estima a melhor aproximação. Além disso, se a solução ótima x^* do problema de minimização pertencer ao espaço gerado por estas p aproximações (subespaço iterativo) e as componentes da função objetivo forem lineares, o passo dado no processo de aceleração também será ótimo, no sentido de apontar para x^*.

Além de escolhermos a direção Levenberg-Marquardt com aceleração, propomos um algoritmo híbrido, no qual substituímos a condição de Armijo dentro das faces, usada em Birgin- Martínez [4], por uma condição de decréscimo suficiente baseada na maneira pela qual o modelo local quadrático aproxima a função objetivo numa vizinhança da aproximação corrente, mantendo a condição de Armijo para o teste de busca linear.

O trabalho está organizado da seguinte forma: no primeiro capítulo fazemos uma revisão dos métodos tradicionais para a resolução de problemas de quadrados mínimos e justificamos a nossa escolha pela direção de Levenberg-Marquardt. No capítulo seguinte, introduzimos as técnicas de aceleração para tentar corrigir a direção de busca. No capítulo 3, apresentamos o novo algoritmo para a minimização de somas de quadrados de funções suaves com restrições de canalização. Vários testes numéricos são apresentados no capítulo 4, para comprovar a eficiência e robustez do novo algoritmo. Por fim, nos apêndices, elucidamos alguns fatos apresentados ao longo do texto e que deram suporte à teoria desenvolvida.

Capítulo 1

Métodos para a resolução de problemas não lineares de quadrados mínimos

Considere o problema de minimizar uma função f real, consistindo em uma soma de quadrados de funções $\overline{f}_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, 2, \ldots, m$, reais, suaves e não-lineares. Isto é, considere o problema

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} \sum_{j=1}^{m} \overline{f}_j^2(x), \tag{1.1}$$

ou simplesmente,

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} ||F(x)||^2, \tag{1.2}$$

onde $F(x) = (\overline{f}_1(x), \ldots, \overline{f}_m(x))$ é uma função vetorial definida em $\mathbb{R}^n$ com valores em $\mathbb{R}^m$ e $|| \cdot ||$ denota a norma Euclidiana.

Na medida em que as dimensões m e n aumentam também se aumenta a dificuldade de tratar com o problema (1.1). Quando essas dimensões são muito grandes, faz-se necessário o uso de métodos iterativos para estimar uma aproximação da solução para o problema citado.

Um método iterativo consiste em gerar uma seqüência $\{x_k\}_{k \in \mathbb{N}}$ de aproximações da solução x^* do problema (1.1), tal que:

(i) se para algum k, x_k for a própria solução x^*, o método pára e fornece x_k como solução;

(ii) caso contrário, o método gera uma seqüência infinita de aproximações para x^*.

Ao longo do texto denotaremos por x_k uma aproximação da solução do problema (1.1) numa iteração k e por f_k, g_k, J_k e H_k sendo respectivamente o valor da função objetivo, o gradiente de f, a matriz Jacobiana de f e a matriz Hessiana de f, todos calculados na aproximação x_k.

Dentre os métodos iterativos mais empregados para resolver o problema (1.1), destacam-se o método Gauss-Newton, o método Levenberg-Marquardt, o método de Newton e os métodos quasi-Newton (vide [34, 6]).

O método de Newton é bem determinado se, a cada iteração k, a matriz Hessiana de f estiver disponível e for definida positiva em x_k. Neste caso, obtém-se taxa de convergência quadrática se a aproximação inicial está suficientemente próxima da solução. Porém, este método possui algumas desvantagens. Primeiramente, é possível que a seqüência x_k gerada não convirja em virtude da direção proveniente desse método ser ortogonal ao gradiente, não permitindo, portanto, a redução do valor da função objetivo. Em segundo lugar, a matriz Hessiana em alguma aproximação x_k pode estar próxima de ser singular e, assim, a solução esperada pode não ser obtida. Além disso, o vetor deslocamento $d_k = x_{k+1} - x_k$ obtido pelo método de Newton pode apresentar uma norma razoávelmente grande tornando os seus valores inadmissíveis ou pode não ser uma direção de descida, no caso em que a Hessiana não é definida positiva. Finalmente, o cálculo completo da matriz Hessiana pode ter um custo computacional muito alto.

Para contornar a última dificuldade, alguns métodos tomam aproximações da matriz Hessiana que sejam mais fáceis de calcular. É o caso do método Gauss-Newton para a minimização de somas de quadrados de funções, que só difere do método de Newton no sentido de que a aproximação da matriz Hessiana é considerada desprezando os termos que contenham as derivadas parciais segundas das funções cuja soma de seus quadrados definem a função objetivo, sendo, portanto, definida pelo produto da transposta da Jacobiana da função objetivo na aproximação corrente por ela mesma. Porém, o método Gauss-Newton não corrige as outras dificuldades já comentadas.

Em busca de controlar o tamanho do passo, surgem os chamados modelos de regiões de confiança, em que o passo corretor d_k para o novo iterando é obtido minimizando um modelo local para a função objetivo f, adicionando a restrição de que a norma da direção d_k seja menor ou igual a um número real $\Delta_k > 0$ dado.

O parâmetro Δ_k desta restrição é expandido ou contraído com base na capacidade do modelo local em prever o comportamento da função objetivo numa vizinhaça de x_k. Desta forma, é possível controlar a iteração, de modo que a convergência seja forçada, a partir de qualquer ponto, admitindo condições razoáveis sobre a função objetivo f. Entretanto, quando m é suficientemente maior do que n, o subproblema quadrático pode exigir um esforço computacional muito grande, comprometendo o desempenho do

algoritmo.

Em 1963, D. Marquardt [28], adotando as idéias de K. Levenberg [26], propôs um método que passa continuamente do método de máxima descida para o método de Gauss-Newton. Isto possibilita que o algoritmo defina sempre uma direção de descida e que, próximo da solução, obtenha a taxa de convergência quadrática derivada do método de Newton.

O algoritmo original do método proposto por Marquardt é bastante simples. Considere que x_k é a aproximação corrente da solução do problema (1.1), g_k o gradiente de f e B_k uma aproximação da matriz Hessiana de f, ambos calculados em x_k. Inicia-se pela resolução do sistema

$$(B_k + \lambda_k I)d_k = -g_k, \tag{1.3}$$

para algum parâmetro λ_k, não negativo, de modo que a matriz $(B_k + \lambda_k I)$ seja definida positiva. Em seguida, define-se como nova aproximação da solução, o vetor

$$x_{k+1} = x_k + d_k,$$

e compara-se o valor da função no novo ponto com o valor atual. Se a nova aproximação define um valor funcional menor, adota-se um parâmetro λ_{k+1} menor que λ_k, para que o método se aproxime do método de Newton próximo da solução. Caso contrário, aumenta-se esse parâmetro, aproximando a direção de busca da direção de máxima descida, garantindo assim o progresso do método.

Muitos autores associam o método Levenberg-Marquardt aos métodos de regiões de confiança. Isto ocorre porque o sistema (1.3) pode ser visto como parte das condições KKT do problema

$$\begin{array}{ll} \min\limits_{d \in \mathbb{R}^n} & q_k(d) = f_k + g_k^T d + \dfrac{1}{2} d^T B_k d, \\ ||d|| \leq \Delta_k & \end{array} \tag{1.4}$$

onde $\Delta_k > 0$ é o raio da região de confiança na k-ésima iteração.

Segundo Gay [23] e Sorensen [43], o problema de encontrar uma direção de busca associada ao probema (1.4) foi inicialmente discutido por Goldfeld, Quandt e Trotter [24] em conexão com o problema de minimização mais geral, no qual a função objetivo é uma função não linear suave qualquer, restrita ao caso em que $B_k + \lambda_k I$ é definida positiva, onde λ_k é uma estimativa superior do menor autovalor da matriz B_k. Para esse problema, Hebden [25] apresentou um algoritmo que calculava uma boa aproximação para d_k quando a matriz $B + \lambda_k I$ era suficientemente definida positiva. Moré [30] refinou o algoritmo de Hebden para somas de quadrados de funções. Sorensen [43] apresentou

um trabalho fortemente influenciado pelo trabalho de Moré [30], porém um pouco mais teórico. Gay [23] e Moré e Sorensen [32] consideraram o caso em que a matriz B_k é singular e $||(B_k + \lambda I)^{-1} g_k|| < \Delta_k,\ \forall \lambda \geq 0$ conhecido como *caso difícil*, e mostraram maneiras eficientes de tratar este problema para se obter uma solução razoável. Porém, se o número de variáveis é grande, o custo computacional pode ser significativo. Mais recentemente, Nocedal e Yuan [35] mesclaram o algoritmo de Moré [30] com uma busca curvilínea, obtendo também bons resultados.

Segundo Nocedal e Yuan (ver [35], página 2), métodos de regiões de confiança são muito eficientes quando aplicados a problemas de pequeno porte. Entretanto, se o número de variáveis é grande, resolver o subproblema (1.4) pode ser ser muito dispendioso. Primeiro, porque exige a solução de um ou mais sistemas lineares como em (1.3). Segundo, porque é possível que não exista λ positivo tal que $||(B + \lambda I)^{-1} g|| = \Delta$ quando B é singular (*caso difícil*). Por estes motivos, adotamos em nosso algoritmo a abordagem clássica do método Levenberg-Marquardt.

Apesar das boas propriedades de convergência do método Levenberg-Marquardt, ele ainda pode fornecer um ponto de sela como solução. Neste trabalho apresentamos uma técnica que, além de acelerar este método, tenta escapar de um ponto crítico que não seja minimizador local para E.

Na próxima seção, fazemos uma breve descrição do método de Newton, ressaltando as desvantagens supra citadas. Na seções 1.2 e 1.3 introduzimos o algoritmo Levenberg-Marquardt irrestrito.

1.1 Método de Newton

Considere o problema de minimização

$$\min_{x \in \mathbb{R}^n} \hat{f}(x), \tag{1.5}$$

onde $\hat{f}$ é uma função duas vezes continuamente diferenciável.

O método de Newton é um processo iterativo para a obtenção de uma solução aproximada de (1.5). A cada iteração k, o método considera como nova aproximação da solução do problema o vetor

$$x_{k+1} = x_k + d_k, \tag{1.6}$$

onde o passo corretor d_k é solução do subproblema quadrático

$$\min_{d \in \mathbb{R}^n} q_k(d) = \hat{f}_k + g_k^T d + \frac{1}{2} d^T H_k d \tag{1.7}$$

e q_k é a aproximação quadrática obtida pela expansão em série de Taylor truncada da função $\hat{f}$ em x_k.

A condição necessária de primeira ordem para o problema (1.7) é dada pelo sistema

$$H_k d = -g_k. \tag{1.8}$$

Assim sendo, as iterações do método de Newton podem ser definidas pela fórmula

$$x_{k+1} = x_k - H_k^{-1} g_k, \tag{1.9}$$

quando a matriz H_k é invertível.

A condição necessária de segunda ordem de (1.7) exige apenas que H_k seja semi-definida positiva. Neste caso, o problema (1.7) pode ter vários minimizadores locais. Se H_k for definida positiva, o problema (1.7) possui um único minimizador global. Este último caso assegura que as iterações de Newton estão bem definidas, no sentido de que a seqüência gerada é única e que d_k é uma direção de descida da função $\hat{f}$.

No caso em que H_k não é definida positiva, a solução de (1.8) pode ser ortogonal ao gradiente g_k ou ainda pode apontar para um maximizador local de (1.7). Estas situações sugerem que d_k não é uma direção de descida para a função $\hat{f}$. Mesmo quando a matriz H_k é definida positiva, a direção d_k, apesar de apontar para um minimizador global para q_k, pode ter os seus valores tão grandes que seja preciso executar uma busca linear nesta direção para obter um vetor que reduza o valor da função objetivo $\hat{f}$. No caso de q_k ter minimizadores locais, também pode ser necessária um busca linear na direção d_k. Entretanto, quando d_k é ortogonal ao gradiente g_k, o método falha em progredir, pois nenhuma busca linear na direção d_k surtirá efeito para se obter uma estimativa inferior para o valor da função objetivo.

Outra desvantagem do método de Newton é que ele exige o cálculo explícito das derivadas primeiras e segundas da função $\hat{f}$ em cada x_k. Neste sentido, quando a função $\hat{f}$ é a soma de quadrados de funções, como em (1.1), o método Gauss-Newton é mais apropriado, já que emprega apenas as derivadas primeiras.

Sob hipóteses razoáveis sobre a matriz Hessiana na solução do problema, o método de Newton gera uma seqüência $\{x_k\}_{k\in\mathbb{N}}$ quadraticamente convergente a uma solução x_* do problema (1.1), se a aproximação inicial x_0 está suficientemente próxima da solução desejada. O seguinte teorema comprova esta asserção.

Teorema 1.1 *Seja $\rho > 0$ um número real dado. Se $\hat{f}$ é uma função continuamente diferenciável tal que a matriz Hessiana H de $\hat{f}$ satisfaz a condição de Lipschitz*

$$||H(x) - H(y)|| \leq \rho ||x - y||$$

numa vizinhança do minimizador local x_* de $\hat{f}$, se x_k está suficientemente próximo de x_* para alguma iteração k, e se a matriz Hessiana H_* de $\hat{f}$ em x_* é definida positiva, então o método de Newton está bem definido e converge quadraticamente a x_*.

Prova Ver, por exemplo, Fletcher [21], pg. 46.

Apresentamos abaixo algumas situações em que o método de Newton pode falhar. Ao longo dos exemplos, usamos a notação $x_{(k)}$ para denotar o vetor x na iteração k, com o propósito de evitar conflito com a notação adotada para a k-ésima componente de x, que é representada por x_k.

Exemplo 1.1 (Powell) $\hat{f}(x) = x_1^4 + x_1x_2 + (1+x_2)^2$ com $x_{(0)} = (0,0)$.

Sejam g_0 e H_0 o gradiente e a Hessiana de $\hat{f}$ calculadas em $x_{(0)}$. O método de Newton não pode ser aplicado satisfatoriamente para este exemplo, pois a matriz Hessiana

$$H_0 = \begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix}$$

é indefinida. Note ainda que qualquer busca linear ao longo de $d = H_0^{-1}g_0 = (-2,0)^T$ altera apenas a primeira componente de $x_{(0)}$. Assim, $x_{(1)} = (h,0)^T$ para algum $h \in \mathbb{R}$ e $\hat{f}(x_{(1)}) = h^4 + 1 \geq 1, \quad \forall h \in \mathbb{R}$. Logo, o minimizador na busca linear é $h = 0$, isto é, $x_{(1)} = x_{(0)}$, e o algoritmo falha em progredir. Esta dificuldade surge porque $g_0^T d = 0$, e as direções d e $-d$ não são de descida. Note, porém, que a função possui um minimizador bem definido, e não é difícil reduzir o valor da função. Por exemplo, tomando d como solução do sistema

$$(H_0 + \lambda I)d = -g_0$$

e usando $\lambda = 2$, obtemos $x_{(1)} = \frac{1}{2}(2,-4)^T$ e $\hat{f}(x_{(1)}) = \frac{65}{49^2} << 1 = \hat{f}(x_{(0)})$.

Exemplo 1.2 $\hat{f}(x) = ax_1^3 + 2x_1^2 + 2x_2^2 + bx_2^3 + 5x_1x_2 - x_1 + x_2 + 10$ com $x_{(0)} = (0,0)$ e $b - a > -3$.

Novamente, o método de Newton não pode ser aplicado satisfatoriamente para este exemplo, pois a matriz Hessiana

$$H_0 = \begin{bmatrix} 4 & 5 \\ 5 & 4 \end{bmatrix}$$

é indefinida. Resolvendo o sistema de Newton, $H_0 d = -g_0$, obtemos $d = (-1,1)^T = g_0$. Isto é, d está na direção oposta a direção de máxima descida. Note que, tomando $-d$ no lugar de d, conseguimos reduzir o valor da função.

Exemplo 1.3 $\hat{f}(x) = ax_1^3 + 2x_1^2 + 2x_2^2 + bx_2^3 + 3x_1x_2 + x_1 - x_2 + 10$ com $x_{(0)} = (0,0)$ e $b - a > -1$.

Observe que a matriz Hessiana inicial

$$H_0 = \begin{bmatrix} 4 & 3 \\ 3 & 4 \end{bmatrix}$$

é positiva definida e que $d = -H_0^{-1}g_0 = -g_0$ é a direção de máxima descida. Porém, por um cálculo simples, e usando o fato de que $b - a > -1$, verifica-se que $\hat{f}(x_{(1)}) > \hat{f}(x_{(0)})$. Neste caso, deveríamos usar uma técnica de redução do passo para encontrar um $\alpha < 1$ tal que o vetor $x_{(1)}(\alpha) = x_{(0)} + \alpha d$ tenha um valor funcional menor do que $\hat{f}(x_{(0)})$. O método de Newton puro, para este exemplo, falha em progredir porque o sistema de Newton tem como solução um vetor d relativamente grande para o problema.

Exemplo 1.4 $\hat{f}(x) = \frac{\sqrt{2}}{2}(e^{x_1x_2} + arctg(x_2))^2$ com $x_{(0)} = (1,0)$.

Verifica-se facilmente, pelos autovalores, que a matriz Hessiana em $x_{(0)}$, dada por

$$H_0 = \begin{bmatrix} 0 & \sqrt{2} \\ \sqrt{2} & 5\sqrt{2} \end{bmatrix},$$

é semidefinida positiva. Resolvendo o sistema de Newton, obtém-se $d = -H_0^{-1}g_0 = (-1,0)^T$, que é ortogonal ao gradiente inicial g_0. Novamente, o método de Newton é incapaz de encontrar um vetor que torne o valor da função menor do que a aproximação inicial, mesmo executando uma busca linear ao longo de d. Com efeito, a direção encontrada só altera a primeira coordenada, e

$$\hat{f}(x_1, 0) = \hat{f}(1,0) = \frac{\sqrt{2}}{2}, \quad \forall x_1 \in \mathbb{R}^n.$$

Observe que qualquer ponto $w = (\overline{x}_1, \overline{x}_2)$ tal que $e^{\overline{x}_1\overline{x}_2} = -arctg(\overline{x}_2)$ é solução global do problema.

1.2 Método de Levenberg-Marquardt irrestrito

Nesta seção, vamos apresentar o método de Levenberg-Marquardt, que abreviamos por *método LM*, para a minimização de funções que consistem em somas de quadrados, como no problema (1.1).

As derivadas parciais de primeira ordem da função f do problema (1.1) são dadas por

$$\frac{\partial f}{\partial x_i} = \sum_{k=1}^{m} \overline{f}_k(x) \frac{\partial \overline{f}_k(x)}{\partial x_i}, \quad \text{para cada } i = 1, 2, \ldots, n. \tag{1.10}$$

De (1.10), obtemos as derivadas parciais de segunda ordem de $f(x)$, que são dadas por

$$\frac{\partial^2 f}{\partial x_i \partial x_j}(x) = \sum_{k=1}^{m} [\frac{\partial \overline{f}_k(x)}{\partial x_i} \frac{\partial \overline{f}_k(x)}{\partial x_j} + \overline{f}_k(x) \frac{\partial^2 \overline{f}_k}{\partial x_i \partial x_j}(x)], \tag{1.11}$$

onde $i, j = 1, 2, \ldots, n$.

Desprezando os termos da segunda parcela de (1.11), teremos uma aproximação das derivadas parciais segundas de f, a saber,

$$\frac{\partial^2 f}{\partial x_i \partial x_j}(x) \cong \sum_{k=1}^{m} \frac{\partial \overline{f}_k(x)}{\partial x_i} \frac{\partial \overline{f}_k(x)}{\partial x_j}. \tag{1.12}$$

Podemos relacionar (1.10) e (1.12) usando a matriz

$$J(x) = \begin{bmatrix} \frac{\partial \overline{f}_1}{\partial x_1}(x) & \frac{\partial \overline{f}_1}{\partial x_2}(x) & \ldots & \frac{\partial \overline{f}_1}{\partial x_n}(x) \\ \frac{\partial \overline{f}_2}{\partial x_1}(x) & \frac{\partial \overline{f}_2}{\partial x_2}(x) & \ldots & \frac{\partial \overline{f}_2}{\partial x_n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \overline{f}_m}{\partial x_1}(x) & \frac{\partial \overline{f}_m}{\partial x_2}(x) & \ldots & \frac{\partial \overline{f}_m}{\partial x_n}(x) \end{bmatrix}. \tag{1.13}$$

Com efeito, considerando o gradiente de (1.1), segue que

$$\nabla f(x) = J(x)^T F(x), \text{ onde } F(x) = (\overline{f}_1(x), \overline{f}_2(x), \ldots, \overline{f}_m(x))^T. \tag{1.14}$$

A Hessiana aproximada, definida a partir de (1.12), pode ser escrita como

$$\nabla^2 f(x) \cong J(x)^T J(x). \tag{1.15}$$

Substituindo (1.14) e (1.15) nas equações de Newton (1.9), obtemos

$$x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k) \cong x_k - (J(x_k)^T J(x_k))^{-1} J(x_k) F(x_k). \tag{1.16}$$

Por simplicidade de notação, sempre que não houver possibilidade de confusão, usaremos $F_k = F(x_k)$, $J_k = J(x_k)$, $g_k = \nabla f(x_k)$, e $B_k = J(x_k)^T J(x_k)$. Desta forma, podemos reescrever a equação (1.16) como

$$J_k^T J_k d_k = -J_k^T F_k, \tag{1.17}$$

onde $d_k = x_{k+1} - x_k$. Esse sistema pode ser apresentado na forma

$$B_k d_k = -g_k. \tag{1.18}$$

Na seção anterior, comentamos que, caso não sejam tomadas medidas adicionais, um método no qual d_k é obtida a partir de (1.18) pode apresentar alguns problemas. Primeiramente, é possível que o método não convirja, no sentido de que o vetor deslocamento d_k pode ser ortogonal ao gradiente g_k, o que faz com que o algoritmo não obtenha sucesso mesmo se uma busca linear for executada, como nos Exemplos 1.1 e 1.4. Além disso, d_k pode ser uma direção de descida viável, porém sua norma pode ser grande demais, como no Exemplo 1.3. Finalmente, a matriz Hessiana pode ser indefinida e a direção obtida poderá apontar para um maximizador local como no exemplo 1.2.

O método de Levenberg-Marquardt é uma forma de garantir um decréscimo no valor da função objetivo. Isto é feito somando um termo adicional, $\lambda_k \geq 0$, à diagonal da aproximação da Hessiana B_k, de modo que a matriz $B_k + \lambda_k I$ seja definida positiva. Obtém-se, assim, um novo sistema,

$$(B_k + \lambda_k I)d_k = -g_k, \tag{1.19}$$

onde I é a matriz identidade de ordem n e λ_k é um coeficiente que torna a matriz dos coeficientes do sistema (1.19) definida positiva e sua solução d_k uma direção de descida a partir de x_k. O escalar λ_k, determinado a cada iteração, é chamado de parâmetro de Levenberg-Marquardt, ou abreviadamente de parâmetro LM.

Observe que, quando $\lambda_k = 0$, o método LM se reduz ao método Gauss-Newton. Neste caso, se a matriz B_k é uma boa aproximação da matriz Hessiana $\nabla^2 f_k$, o método terá taxa de convergência quadrática partindo de um ponto próximo da solução. Se λ_k tender ao infinito, d_k tenderá à direção de máxima descida, porém a sua norma tenderá a zero.

Com efeito, os teoremas abaixo, apresentados por Marquardt [28], comprovam o que foi dito.

Teorema 1.2 *Seja $d(\lambda)$ a solução de (1.19) para um dado valor de λ. Então $||d(\lambda)||^2$ é uma função contínua decrescente de λ, tal que $||d(\lambda)|| \to 0$ se $\lambda \to +\infty$.*

Prova: Ver Marquardt [28].

Teorema 1.3 *Seja γ o ângulo entre $d(0)$ (direção* Gauss-Newton*) e $-g$ (direção de máxima descida). Então, γ é uma função contínua monótona decrescente de λ, tal que $\gamma \to 0$ se $\lambda \to +\infty$. Como $-g$ é independente de λ, segue que d rotaciona na direção $-g$ quando $\lambda \to +\infty$.*

Prova: Ver Marquardt [28].

Na prática, utilizamos λ_{k-1} (usado para tornar definida positiva a matriz $B_{k-1} + \lambda_{k-1} I$) para estimar o valor desse parâmetro na iteração k, aumentando-o se necessário.

Os resultados acima sugerem que, quando a direção de Levenberg-Marquardt provoca um decréscimo suficiente no valor da função objetivo, devemos diminuir o parâmetro LM de modo a forçar o método a se comportar como o método de Gauss-Newton próximo de uma solução local, esperando assim obter taxa de convergência quadrática. Caso não se obtenha um decréscimo suficiente, aumentamos o valor de λ_k, forçando o algoritmo a nos fornecer uma direção mais próxima da direção de máxima descida.

Verificamos se a matriz $B_k + \lambda_k I$ é definida positiva tentando decompô-la usando a fatoração de Cholesky. Se o algoritmo da fatoração de Cholesky é bem sucedido, ou seja, se encontramos o fator de Cholesky, então a matriz é definida positiva e usamos sua decomposição para resolver o sistema (1.19) através de dois sistemas triangulares. Caso contrário, aumentamos o valor de λ_k e tentamos uma nova fatoração.

No algoritmo de Levenberg-Marquardt, uma vez obtido o vetor d_k, a nova aproximação da solução do problema (1.1) é definida como $x_{k+1} = x_k + d_k$. Como nos métodos de regiões de confiança, para aceitar esse ponto, avaliamos quão bem o modelo quadrático local prevê o comportamento da função objetivo. Para tanto, definimos a redução real da função objetivo como $ared_k = f_k - f_{k+1}$, e a redução prevista pelo modelo quadrático como $pred_k = q_k(0) - q_k(d_k)$. Se a razão $\rho_k = ared_k/pred_k$ for superior a uma constante η, aceitamos a nova aproximação x_{k+1}. Caso contrário, descartamos a direção encontrada, aumentamos o parâmetro LM e definimos um novo passo.

Para evitar problemas numéricos no cálculo de ρ_k seguimos as idéias de Moré [30]. Assim, observando que

$$pred_k = q_k(0) - q_k(d_k) = \lambda_k ||d_k||^2 + \frac{1}{2}||J_k d_k||^2, \tag{1.20}$$

obtemos

$$\begin{aligned} \rho_k &= \frac{ared_k}{pred_k} = \frac{f_k - f_{k+1}}{\lambda_k ||d_k||^2 + \frac{1}{2}||J_k d_k||^2} = \frac{\frac{1}{2}||F_k||^2 - \frac{1}{2}||F_{k+1}||^2}{\lambda_k ||d_k||^2 + \frac{1}{2}||J_k d_k||^2} \\ &= \frac{1 - (\frac{||F_{k+1}||}{||F_k||})^2}{(\frac{||J_k d_k||}{||F_k||})^2 + 2(\sqrt{\lambda_k}\frac{||d_k||}{||F_k||})^2}. \end{aligned} \tag{1.21}$$

Por outro lado,

$$\begin{aligned} ||F_k||^2 - ||F_k + J_k d_k||^2 &= -2d_k^T J_k^T F_k - ||J_k d_k||^2 = -2d_k^T g_k - ||J_k d_k||^2 \\ &= 2d_k^T (B_k + \lambda_k I) d_k - ||J_k d_k||^2 \\ &= 2\lambda_k ||d_k||^2 + ||J_k d_k||^2, \end{aligned}$$

de modo que

$$||J_k d_k|| \leq ||F_k|| \text{ e } \sqrt{\lambda}||d_k|| \leq ||F_k||. \tag{1.22}$$

De (1.22), concluímos que o cálculo do denominador em (1.21) nunca gerará erros numéricos, uma vez que $||F_k||$ só é empregada no cálculo de 1.21 se x_k não está suficientemente próximo à uma solução e do denominador de 1.21 estar bem definido.

Usamos comos critérios de parada a relação $||F_k|| \leq \varepsilon_1$ para alguma aproximação x_k e a condição auxiliar $pred_k \leq \varepsilon_2$, onde $\varepsilon_1, \varepsilon_2 > 0$ são precisões fornecidas. Em geral, quando uma dessas condições é satisfeita, estamos muito perto da solução.

Ainda é possível associar o método de Levenberg-Marquardt às técnicas de regiões de confiança. Com efeito, nos métodos de regiões de confiança, monitoramos o raio Δ_k, enquanto no método LM, monitoramos o parâmetro λ_k. Além disso, se $\lambda_k \geq 0$ tal que a matriz $B_k + \lambda_k I$ é definida positiva, podemos definir Δ_k como

$$\Delta_k = || - (B_k + \lambda_k I)^{-1} g_k||. \tag{1.23}$$

Com isso, prova-se que $d_k = -(B_k + \lambda_k I)^{-1} g_k$ é solução do problema

$$\min_{||w|| \leq \Delta_k} q_k(d). \tag{1.24}$$

Portanto, podemos pensar que o método Levenberg-Marquardt é um método de região de confiança no qual o raio Δ_k é dado implicitamente em função de λ_k pela equação (1.23). A análise de convergência da maioria dos algoritmos do tipo Levenberg-Marquardt se baseia neste fato.

Os dois lemas seguintes relacionam o método Levenberg-Marquardt ao método de região de confiança e, ao mesmo tempo, fornecem uma estrutura para implementações numéricas. As demonstrações dos lemas, encontradas em Sorensen [43], são apresentadas de forma detalhada abaixo.

Lema 1.1 *Se d_* é uma solução do problema*

$$\min_{||d|| \leq \Delta} q(d) = g^T d + \frac{1}{2} d^T B d, \tag{1.25}$$

então d_* é uma solução da forma

$$(B + \lambda I) d_* = -g, \tag{1.26}$$

com $B + \lambda I$ semi-positiva definida, $\lambda \geq 0$ e

$$\lambda(||d_*|| - \Delta) = 0. \tag{1.27}$$

Prova Das condições KKT, segue que d_* deve resolver uma equação da forma (1.26), com condições de complementaridade (1.27). Resta mostrar que a matriz $B + \lambda I$ é

semidefinida positiva. Para isto, vamos supor inicialmente que $d_* \neq 0$. Como d_* resolve (1.26), também resolve o problema

$$\min_{||d||=||d_*||} q(d) = g^T d + \frac{1}{2} d^T B d.$$

Assim, $q(d) \geq q(d_*),\ \forall d$ tal que $||d|| = ||d_*||$. Ou seja,

$$g^T d + \frac{1}{2} d^T B d \geq g^T d_* + \frac{1}{2} d_*^T B d_*.$$

Uma vez que $g^T s = -s^T (B + \lambda I) d_*$ qualquer que seja s, a última desigualdade pode ser reescrita como

$$-d^T (B + \lambda I) d_* + \frac{1}{2} d^T B d \geq -d_*^T (B + \lambda I) d_* + \frac{1}{2} d_*^T B d_*. \tag{1.28}$$

Adicionando e subtraindo $\frac{\lambda}{2} d^T d$ ao lado esquerdo de (1.28) e reordenando os termos desta inequação, obtemos

$$\frac{1}{2}(d_* - d)^T (B + \lambda I)(d_* - d) \geq \frac{\lambda}{2}(||d||^2 - ||d_*||^2) = 0,\ \forall d\ :\ ||d|| = ||d_*||. \tag{1.29}$$

Como $d_* \neq 0$ e d é arbitrário, a desigualdade (1.29) mostra que $B + \lambda I$ é semidefinida positiva.

Se, por outro lado, $d_* = 0$, segue de (1.26) que $g = 0$ e d_* resolve o problema irrestrito

$$\min_{d \in \mathbb{R}^n} q(d),$$

com valor ótimo $q_* = q(d_*) = 0$. Uma vez que, $d^T B d = 2q(d) \geq 2q_* = 0$ qualquer que seja $d \in \mathbb{R}^n$, concluímos que B é semidefinida positiva.■

Lema 1.2 *Sejam $\lambda \in \mathbb{R}$, $d_* \in \mathbb{R}^n$ satisfazendo* (1.26) *com $B + \lambda I$ semidefinida positiva.*

(i) *Se $\lambda = 0$ e $||d_*|| < \Delta$, então d_* resolve* (1.25).

(ii) *d_* resolve o problema*

$$\min_{||d|| \leq ||d_*||} q(d) = g^T d + \frac{1}{2} d^T B d.$$

(iii) *Se $\lambda \geq 0$ e $||d_*|| = \Delta$, então d_* resolve* (1.25).

Além disso, se $B + \lambda I$ é definida positiva, d_* é a única solução de (1.25).

Prova: Como $(B+\lambda I)d_* = -g$, então,

$$\left\{\begin{array}{rcl} d_*^T(B+\lambda I)d_* &=& -g^T d_* \\ d^T(B+\lambda I)d_* &=& -g^T d \end{array}\right. \Rightarrow \left\{\begin{array}{rcl} g^T d_* + d_*^T(B+\lambda I)d_* &=& 0, \\ g^T d + d^T(B+\lambda I)d_* &=& 0. \end{array}\right. \tag{1.30}$$

Uma vez que $B+\lambda I$ é semidefinida positiva, temos

$$(d-d_*)^T(B+\lambda I)(d-d_*) \geq 0,\ \forall d \in \mathbb{R}^n. \tag{1.31}$$

Desenvolvendo a última desigualdade, obtemos

$$d_*^T(B+\lambda I)d_* + d^T(B+\lambda I)d \geq 2d^T(B+\lambda I)d_*,$$

ou ainda

$$d^T(B+\lambda I)d_* \leq \frac{1}{2}d_*^T(B+\lambda I)d_* + \frac{1}{2}d^T(B+\lambda I)d. \tag{1.32}$$

De (1.30) e (1.32), temos

$$\begin{array}{rcl} g^T d_* + d_*^T(B+\lambda I)d_* &=& g^T d + d^T(B+\lambda I)d_* \\ &\leq& g^T d + \frac{1}{2}d_*^T(B+\lambda I)d_* + \frac{1}{2}d^T(B+\lambda I)d. \end{array}$$

Então,

$$g^T d_* + d_*^T(B+\lambda I)d_* - \frac{1}{2}d_*^T(B+\lambda I)d_* \ \leq\ g^T d + \frac{1}{2}d^T(B+\lambda I)d,\ \forall d \in \mathbb{R}^n.$$

Logo,

$$g^T d_* + \frac{1}{2}d_*(B+\lambda I)d_* \leq g^T d + \frac{1}{2}d(B+\lambda I)d,\ \forall d \in \mathbb{R}^n. \tag{1.33}$$

De (1.33), obtemos

$$q(d) \geq q(d_*) + \frac{\lambda}{2}(||d_*||^2 - ||d||^2),\ \forall d \in \mathbb{R}^n. \tag{1.34}$$

As afirmações $(i), (ii)$ e (iii) seguem diretamente de (1.34). A unicidade segue de (1.33) porque temos desigualdade estrita quando $B+\lambda I$ é definida positiva e $d_* \neq d$.■

A maior dificuldade para se obter uma solução de (1.25) ocorre quando B_k não é definida positiva e (1.25) não tem solução na fronteira de $\Omega_k = \{d : ||d|| \leq \Delta_k\}$. Com efeito, não é difícil ver que (1.25) não tem solução w com $||w|| = \Delta$ se, e somente se, B não é positiva definida e $||-B_k^{-1}g|| < \Delta_k$.

Agora, vamos supor que (1.25) tem uma solução sobre a fronteira de Ω. Se g não é perpendicular ao autoespaço

$$S_1 = \{z : Bz = \lambda_1 z, z \neq 0\},$$

onde λ_1 é o menor autovalor de B, então a equação não linear $||w(\lambda)|| = \Delta$, onde

$$w(\lambda) = -(B + \lambda I)^{-1} g,$$

tem uma solução $\lambda \geq 0$ no intervalo $(-\lambda_1, +\infty)$. Caso contrário, isto é, se g é perpendicular a S_1, então não garantimos a existência de uma solução para a equação $||w(\lambda)|| = \Delta$. Este caso é conhecido como *caso difícil* ou *hard case* (ver Gay [23], Moré e Sorensen [32] ou Sorensen [43]).

Neste caso, uma solução do problema (1.25) pode ser obtida resolvendo o sistema

$$(B - \lambda_1 I)d = -g$$

para $||d|| \leq \Delta$ e determinando um autovetor $z \in S_1$. Então, definimos

$$w = d + \tau z,$$

onde τ é um parâmetro real tal que $||d + \tau z|| = \Delta$. O Lema 1.2 garante que w é solução de (1.25).

Gay[23], Sorensen[43] e Moré e Sorensen[32] propuseram técnicas para encontrar o autovetor $z \in S_1$. Entretanto, elas exigem um cálculo computacional considerável, principalmente se $m >> n$ é muito grande. Para problemas de grande porte, uma solução eficiente foi apresentada por Rojas, Santos e Sorensen [39], que criaram um algoritmo que não exige o uso de fatorações, baseando-se apenas em produtos de matrizes por vetores.

O caso em que existe solução $\lambda \geq 0$ no intervalo $(-\lambda_1, +\infty)$ tal que $||w(\lambda)|| = \Delta$ é bastante fácil de ser tratado. Com efeito, todos os artigos que tratam deste problema consideram o subproblema de encontrar um zero para a função

$$\varphi(\lambda) = \frac{1}{||w(\lambda)||} - \frac{1}{\Delta}. \tag{1.35}$$

Na grande maioria dos artigos, aplica-se o método de Newton ao problema (1.35) para resolver a equação $\phi(\lambda) = 0$. Este método é muito eficiente na prática devido à concavidade da função φ.

No Apêndice A, apresentamos os cálculos para a obtenção de uma aproximação da solução do problema (1.35). Leitores interessados devem ler Moré[30], Gay[23], Sorensen[43] e Moré e Sorensen[32], que discutem este aspecto.

1.3 Escolha do parâmetro de Levenberg - Marquardt

Nesta seção, discutiremos as estratégias para a escolha e atualização do parâmetro λ_k, que define o algoritmo de Levenberg-Marquardt clássico apresentado na seção anterior.

Lembramos que, a cada iteração do algoritmo de Levenberg- Marquardt, fornecemos uma estimativa inicial λ_k desse parâmetro, que deve ser aumentado até que a matriz

$$B_k + \lambda_k I, \tag{1.36}$$

associada ao modelo quadrático local, seja definida positiva e uma redução significativa do valor da função objetivo seja atingida.

Mais precisamente, uma escolha admissível do parâmetro de Levenberg-Marquardt pode envolver várias tentativas de encontrar o fator de Cholesky da matriz (1.36), e quanto pior for a nossa estimativa inicial maior será o esforço computacional para obter uma melhor aproximação da solução do problema de otimização. Por outro lado, dificuldades numéricas também podem surgir quando a estimativa inicial é muito grande. De fato, quanto maior o parâmetro de Levenberg-Marquardt, menor é a norma da solução d que define a direção de busca na iteração atual, e o algoritmo pode parar mesmo estando longe da solução desejada.

Assim, a razão de convergência do algoritmo LM, bem como a sua eficiência, dependem diretamente da escolha da estimativa inicial do parâmetro de Levenberg-Marquardt em cada iteração.

Abaixo, descrevemos algumas possibilidades de escolha do parâmetro λ_k. Se B_k já é definida positiva, pode-se tomar $\lambda_k = 0$. Portanto, resta-nos considerar o caso em que é necessário encontrar um λ_k estritamente positivo.

Seguindo as idéias de Davies e Whitting [14], definiremos $\lambda_k = \frac{1}{w_k}$ para algum $w_k > 0$, de modo que, agora, nosso problema consiste em encontrar w_k. Além disso, omitiremos abaixo o índice k para facilitar a notação.

Considere a função

$$\phi(w) = f(x + d(w)), \tag{1.37}$$

onde

$$(B + \frac{1}{w}\Sigma)d(w) = -g, \tag{1.38}$$

com $\Sigma = diag(a_1, \ldots, a_n), \quad a_i > 0$, escolhida apropriadamente. A equação (1.38) é equivalente a

$$(wB + \Sigma)d(w) = -wg. \tag{1.39}$$

Para encontrar w, precisamos determinar o zero da expansão linear de $\phi(w)$ em torno de $w = 0$, isto é, o zero da função

$$\psi(w) = \phi(0) + \phi'(0)w. \tag{1.40}$$

Derivando $\phi(w)$ com repeito a w em (1.37), chegamos a

$$\phi'(w) = \nabla f(x + d(w))^T d'(w),$$

em que $d'(w)$ denota a derivada da função vetorial $d : \mathbb{R} \to \mathbb{R}^n$. Por outro lado, tomando $w = 0$ em (1.39), encontramos $d(0) = 0$. Logo, para $w = 0$, a última igualdade pode ser reescrita como,

$$\phi'(0) = \nabla f(x + d(0))^T d'(0) = \nabla f(x)^T d'(0) = g^T d'(0).$$

Derivando implicitamente a equação (1.39), obtemos

$$(wB + \Sigma)d'(w) = -g - Bd(w). \tag{1.41}$$

Assim, tomando $w = 0$ em (1.41), encontramos

$$\Sigma d'(0) = -g,$$

ou simplesmente

$$a_i d_i'(0) = -g_i, \quad i = 1, 2, \ldots, n. \tag{1.42}$$

A equação (1.42) impõe uma restrição para a escolha de $d_i'(0)$. Se $g_i = 0$, consideramos $d_i' = 0$ para qualquer valor de a_i.

Agora, escrevamos simplesmente

$$v = d'(0).$$

Assim, se $a_i > 0$, de (1.42), segue que

$$v_i = \frac{g_i}{a_i} = g_i a_i^{-1}.$$

O zero da função $\psi(w)$ é dado por

$$w = -\frac{\phi(0)}{\phi'(0)} = -\frac{f}{g^T v}. \tag{1.43}$$

Como $\lambda = 1/w$, segue que

$$\lambda = \frac{\phi'(0)}{\phi(0)} = -\frac{g^T v}{f}. \tag{1.44}$$

O parâmetro λ na equação (1.44) está vinculado à escolha da matriz escalamento Σ, que, por sua vez, é usada para determinar o vetor v. Abaixo, apresentamos algumas sugestões para o parâmetro LM, definidas a partir de (1.44), com uma escolha conveniente do vetor v.

$E1$ Levenberg [26].

$$\lambda_k = \frac{||g_k||^2}{f_k}, \quad \text{obtido com} \quad v_k = g_k.$$

$E2$ Levenberg modificado 1.

$$\lambda_k = ||g_k||^2, \quad \text{obtido com} \quad v_k = f_k g_k.$$

$E3$ Levenberg modificado 2.

$$\lambda_k = ||g_k||, \quad \text{obtido com} \quad v_k = \frac{f_k}{||g_k||} g_k.$$

$E4$ Fan e Yuan [18].

$$\lambda_k = ||F_k||^2 = 2f_k \quad \text{obtido com} \quad (v_k)_i = \begin{cases} \frac{2f_k}{p(g_k)_i} & \text{se } (g_k)_i \neq 0, \\ 0 & \text{se } (g_k)_i = 0, \end{cases}$$

onde p é o número de componentes não nulas do gradiente g_k.

$E5$ Yamashita e Fukushima [45].

$$\lambda_k = ||F_k|| = 2\sqrt{f_k} \quad \text{obtido com} \quad (v_k)_i = \begin{cases} \frac{2\sqrt{f_k}}{p(g_k)_i} & \text{se } (g_k)_i \neq 0, \\ 0 & \text{se } (g_k)_i = 0. \end{cases}$$

$E6$ Moré [30].

$$\lambda_k = g_k^T v_k, \tag{1.45}$$

sendo $(v_0)_i = \left\| \frac{\partial f(x_0)}{\partial x_i} \right\|$ (a norma da i-ésima coluna da matriz Jacobiana)
e $(v_k)_i = \max \left\{ (v_{k-1})_i, \ \left\| \frac{\partial f(x_k)}{\partial x_i} \right\| \right\}$.

A motivação para definir a estratégia $E6$ está associada à técnica para escalar a matriz Hessiana truncada B_k proposta por Moré [30]. Porém, o algoritmo proposto por Moré difere desta estratégia, uma vez que a escolha de λ_k dada por (1.45) altera uniformemente as coordenadas da direção d, enquanto o método de Moré modifica separadamente cada coordenada de d.

As estratégias $E2$ e $E3$ foram motivadas pelas escolhas adotadas em Fan e Yuan [18] e Yamashita e Fukushima [45]. Porém, em lugar de estimar o parâmetro LM pelo valor da função, consideramos a norma do gradiente da aproximação corrente. Esta escolha é razoável, uma vez que o valor da função na solução ótima x^* pode não ser nulo e, conseqüêntemente, o parãmetro LM também não se anulará numa vizinhança de x^*. Entretanto, em geral, o gradiente se anula em x^*, sugerindo ser possível atribuir um valor pequeno a λ_k próximo da solução.

As estratégias $E1-E6$ estimam, no ínicio de cada iteração, um valor para o parâmetro LM. Também existem estratégias que tentam aproveitar o valor do parâmetro calculado

em uma iteração imediatamente anterior para estimar o valor do novo parâmetro na iteração corrente. Este tipo de estratégia necessita de uma aproximação inicial λ_0 do parâmetro LM e leva em conta que, quanto mais próximo da solução, menor deverá ser o parâmetro de Levenberg-Marquardt. Isto é bastante natural, uma vez que, para valores muito pequenos de λ_k, o método de LM se comporta como o método de Gauss-Newton, herdando suas boas propriedades de convergência.

Considere o modelo local de f em x_k,

$$q_k(d) = f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T (B_k + \lambda_k I) d,$$

em que λ_k um número não negativo que assegura a positividade da matriz $B_k + \lambda_k I$.

Seja d_k o minimizador irrestrito de $q_k(d)$. Defina $x_{k+1}^t = x_k + d_k$ como a próxima estimativa do minimizador de f. Seja

$$ared_k = f(x_{k+1}^t) - f(x_k)$$

a redução provocada em f pela nova aproximação,

$$pred_k = q_k(0) - q_k(d_k)$$

a redução prevista pelo modelo local q_k e

$$\rho_k = ared_k / pred_k.$$

Dado λ_k, estima-se o próximo parâmetro λ_{k+1} a partir da razão ρ_k. Se ρ_k está próximo de 1, entendemos que o modelo local está aproximando bem a função f em torno de x_k. Assim, desejamos que λ_{k+1} seja menor que λ_k. Fazemos isto pré-multiplicando λ_k por um número real $\alpha,\ 0 < \alpha < 1$, e atribuímos o resultado à estimativa inicial de λ_{k+1}. Por outro lado, se ρ_k é muito pequeno ou não positivo, obtemos λ_{k+1} multiplicando λ_k por um número real $\beta > 1$, uma vez que pode-se garantir o decréscimo da função para algum λ_{k+1} suficientemente grande.

Marquardt [28] utiliza $\alpha = 1/\beta$, com $\beta = 10$. Outros autores adotam a mesma estratégia de Marquardt, mas empregam $\beta = 2$. Osborne [36] sugere $\alpha = 1/3$ e $\beta = 3/2$.

Madsen, Nielsen e Tingleff [27] afirmam que a escolha de α e β deve ser feita de modo a não permitir que a razão λ_{k+1}/λ_k varie muito de uma iteração para outra.

Nielsen [33] sugere um processo de atualização do parâmetro de Levenberg-Marquardt que procura evitar os saltos bruscos do escalar λ entre iterações consecutivas. Grandes oscilações do parâmetro de Levenberg-Marquardt podem exigir um esforço maior para a obtenção de um valor admissível. Tal processo tem apresentado bons resultados na prática, tornando-o competitivo em relação às outras escolhas. Supondo dadas as constantes reais $\eta > 0$, $\mu > 1$ e $\beta > 0$, e um inteiro positivo e ímpar p, o processo proposto por Nielsen pode ser descrito pelo seguinte esquema:

Se $\rho_k > \eta$, faça

$$\lambda_{k+1} = \lambda_k \max\left\{\frac{1}{\mu}, 1-(\beta-1)(2\rho_k-1)^p\right\}; \quad \nu = \beta. \tag{1.46}$$

Senão,

$$\lambda_k = \nu\lambda_k; \quad \nu = 2\nu. \tag{1.47}$$

Como se observa, se há uma série de iterações consecutivas nas quais a nova aproximação é rejeitada, isto é, $\rho_k \leq \eta$, o valor de λ_k aumenta rapidamente.

Definimos como a estratégia $E7$, o processo de atualização (1.46)-(1.47), considerando $\mu = 3, \beta = 2$ e $\nu = 2$. O processo de atualização dado por Marquardt [28] (com $\beta = 10$ e $\alpha = 1/\beta$) define a estratégia $E8$.

As duas últimas estratégias avaliadas têm por base o algoritmo abaixo.

Se $\rho_k > \eta$, faça

$$\lambda_{k+1} = \alpha\lambda_k; \quad \nu = \beta. \tag{1.48}$$

Senão,

$$\lambda_{k+1} = \nu\lambda_k; \quad \nu = 2\nu. \tag{1.49}$$

Neste algoritmo, o parâmetro ν tem como valor inicial β, que deve ser fornecido. Denominamos $E9$ a estratégia (1.48)-(1.49) com $\beta = 3/2$ e $\alpha = 1/3$. Por outro lado, definimos $E10$ tomando $\beta = 2$ e $\alpha = 1/2$. Esses valores foram inspirados em Osborne [36] e em Madsen, Nielsen e Tingleff [27], respectivamente.

Utilizamos os 35 problemas testes descritos em Moré, Garbow e Hillstrom [31], sem busca linear, para comparar as estratégias acima. Uma vez que 19 desses problemas têm dimensão variável, fornecemos na Tabela 1.1 as dimensões adotadas nesses casos.

Os valores iniciais do parâmetro de Levenberg que definem as estratégias $E7$ a $E10$ são descritos abaixo, onde ε_{rel}, ε_{abs} são valores definidos *a priori*. Em nosso testes, usamos $\varepsilon_{rel} = 10^{-10}$ e $\varepsilon_{abs} = 10^{-12}$.

$E7: \ \lambda_0 = \max\{\varepsilon_{abs} \max((B_0)_{ii}), \varepsilon_{rel}\}$

$E8: \ \lambda_0 = 0.01.$

$E9: \ \lambda_0 = 1.$

$E10: \ \lambda_0 = 1.$

problema	n	m
Jenrich e Sampson	2	10
Gulf	3	10
Biggs EXP6	6	13
Watson	9	31
Rosenbrock Extendida	200	200
Powell Extendida	200	200
Penalidade I	10	11
Penalidade II	10	20
Função de dimensão variável [1]	200	202
Trigonométrica	200	200
Brown quase-linear	200	200
Valores no bordo discreto	200	200
Equação integral	200	200
Broyden Tridiagonal	200	200
Broyden Banda	200	200
Linear - posto completo	200	200
Linear - posto um	200	200
Linear - posto um com colunas e linhas nulas	200	200
Chebyquad	80	80

Tabela 1.1: Dimensão (n) e número de funções coordenadas (m) usados nos problemas de dimensão variável.

Na Figura 1.1, comparamos o perfil de desempenho[2] das estratégias mencionadas, com exceção de $E2$ e $E4$, que se mostraram bastante inferiores às demais. Para estas estratégias, o parâmetro LM assumiu valores altos, piorando o desempenho do algoritmo.

Conforme descrito no Apêndice C, quanto mais alta a curva do perfil de desempenho, mais eficiente o algoritmo. Além disso, se o perfil de desempenho de algum programa consegue atingir o valor um no eixo vertical então esse programa é robusto, no sentido de resolver todos os problemas testados. Desta forma, verificamos que as quatro últimas estratágias foram as que apresentaram os melhores resultados. Destas, $E7$ foi a que consumiu o menor número de iterações para resolver a maioria dos problemas. Nenhuma das estratégias, entretanto, foi capaz de resolver o problema Gulf, sendo $E3$ a estratágia que obteve o menor valor funcional.

[2]O Perfil de Desempenho é um gráfico criado por Dolan e Moré [16] para comparar algoritmos. O leitor interessado deve se dirigir ao Apêndice C para maiores detalhes.

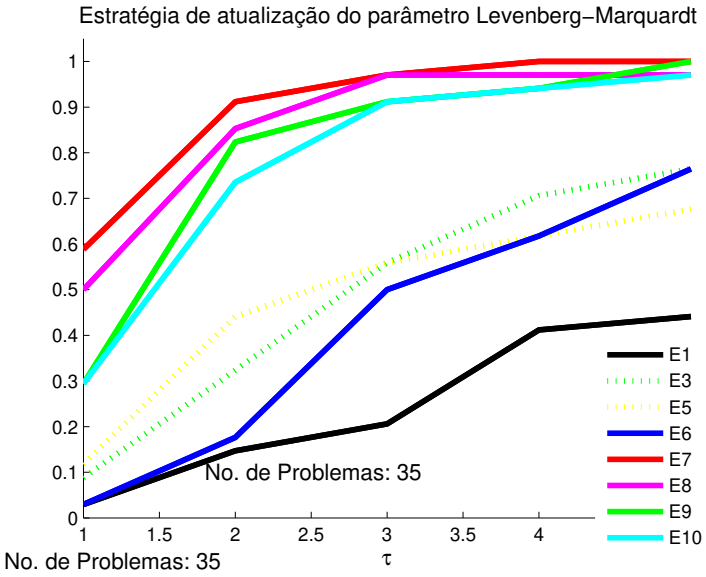

Figura 1.1: Perfis de desempenho das estratégias de atualização do parâmetro LM em função do número de iterações.

Apesar de usarmos problemas de pequeno porte para selecionar a estratégia de estimativa do parâmetro de Levenberg, testes preliminares comprovaram a eficiência da estratégia $E7$ para problemas com dimensão mais alta. Por isso, adotaremos essa estratégia daqui por diante.

Capítulo 2

Técnicas de aceleração

Apesar de ser globalmente convergente, o método de Levenberg-Marquardt apresenta algumas desvantagens. Dentre elas, podemos destacar a necessidade de se calcular e armazenar a matriz Hessiana truncada $B = J^T J$ a cada iteração. Se o número de linhas m da matriz Jacobiana J é grande, o método terá um custo computacional elevado.

É interessante, portanto, buscar novas técnicas que acelerem o método LM, reduzindo o número de iterações e, conseqüentemente, o tempo gasto pelo algoritmo. Muitos pesquisadores têm estudado e proposto técnicas para acelerar o método LM. A maioria delas concentram-se nas técnicas tradicionais de busca linear e na escolha de uma matriz escalamento para melhorar o número de condição da matriz Hessiana (ver Sakamoto et al. [40]) ou controlar o tamanho do passo para que a nova aproximação seja interior à caixa (Coleman [12]). Outras estratégias consideradas para acelerar o método incluem uma reformulação da função objetivo (como proposto por Wilamowski, Chen Y e Malinowski [44]) e a utilização de técnicas de decomposições matriciais (ver Zhou e Si [47] e Chan e Szeto [8]) para resolver o sistema Levenberg-Marquardt.

Em 1980, Pulay [37] apresentou uma proposta que acelera o algoritmo LM através de uma correção na direção de busca. Esta técnica, denominada DIIS, é empregada principalmente na área da química molecular e em otimização geométrica (ver, por exemplo, Farkas e Schlegel [20]). Ela considera que uma melhor aproximação da solução final pode ser construída como combinação linear das últimas p aproximações, supondo que os coeficientes desta combinação minimizam, no sentido dos quadrados mínimos, o vetor residual formado pela combinação dos passos anteriormente calculados. Existem também variantes dessa estratégia, como as propostas por Csaszar e Pulay[13] e Farkas e Schlegel [19], que usam as informações dos gradientes correspondentes às p aproximações consideradas, em lugar dos respectivos passos corretores.

Neste capítulo, apresentamos uma proposta de aceleração inspirada na técnica DIIS.

De fato, corrigimos a direção de busca a partir de observações de um número limitado de valores funcionais computados em passos anteriores e na iteração corrente. Esta é uma forma barata de se calcular os coeficientes da combinação linear que estima a melhor aproximação. Além disso, se a solução ótima x^* do problema de minimização pertencer ao espaço gerado por estas p aproximações e as componentes da função objetivo forem lineares, o passo dado no processo de aceleração também será ótimo, no sentido de apontar para x^*.

Abaixo, apresentamos a técnica de aceleração proposta por Pulay e a nova proposta de aceleração. Veremos, ao final do capítulo, que estas técnicas podem aumentar o desempenho do algoritmo LM. Esses resultados positivos também são observados quando tentamos acelerar o algoritmo GENCAN com direção de Levenberg-Marquardt no interior das faces ativas, como comprovam os experimentos numéricos apresentados no Capítulo 4.

2.1 DIIS: *Direct Inversion of the Iterative Subspace*

O método $DIIS$ é uma técnica de aceleração introduzida por Pulay [37]. Este método propõe que se obtenha uma melhor aproximação x^a da solução de um problema de minimização a partir de $p+1$ vetores $x^i, i = 0, 1, 2, \ldots, p$, onde x_0 é uma aproximação inicial dada e as demais aproximações são calculadas por algum método iterativo. Define-se, assim,

$$x^a = \sum_{i=1}^{p} u_i x^i,$$

onde os coeficientes $u_i,\ i = 1, 2, \ldots, p$, são obtidos de forma que o vetor residual associado,

$$d^a = \sum_{i=1}^{p} u_i d^i = \sum_{i=1}^{p} u_i (x^i - x^{i-1})$$

seja o menor possível, no sentido dos quadrados mínimos. Exige-se também que

$$\sum_{i=1}^{p} u_i = 1.$$

Segundo Sherrill [42], a motivação para esta última exigência é o fato de que cada uma das soluções x^i pode ser escrita como soma da solução exata x^* e de um termo de erro e^i, ou seja,

$$x^i = x^* + e^i,\ i = 1, 2, \ldots, p.$$

Assim, a solução aproximada dada pelo método $DIIS$ é descrita por

$$\begin{aligned} x^a &= \sum_{i=1}^{p} u_i x^i = \sum_{i=1}^{p} u_i (x^* + e^i) \\ &= x^* \left[\sum_{i=1}^{p} u_i \right] + \sum_{i=1}^{p} u_i e^i = x^* + \sum_{i=1}^{p} u_i e^i. \end{aligned}$$

O objetivo é a minimização do erro efetivo, que é o segundo termo do lado direito da última equação. Na prática, como não se conhece os valores das componentes e^i, estas são substituídas pelas direções calculadas nas últimas p iterações.

Logo, supondo que estejamos na iteração k e que as direções $d^i,\ i = k-p+1, \ldots, k$ estejam disponíveis, precisamos resolver o problema

$$\begin{aligned} min \quad & \frac{1}{2} \left\| \sum_{i=1}^{p} u_i d^{k-p+i} \right\|^2 \\ suj.\ a \quad & \sum_{i=1}^{p} u_i = 1. \end{aligned} \tag{2.1}$$

Definindo a matriz simétrica $A_{p\times p} = [A_{ij}] = [(d^{k-p+i})^T d^{k-p+j}]$, as condições KKT do problema (2.1) são dadas pelo sistema

$$\begin{bmatrix} A^T A & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} u \\ \overline{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}, \tag{2.2}$$

onde $\mathbf{1} = [1\ 1\ \ldots\ 1]^T \in \mathbb{R}^p$, $\mathbf{0} = [0\ 0\ \ldots\ 0]^T \in \mathbb{R}^p$, u é solução de (2.1) e $\overline{\lambda}$ é o multiplicador de Lagrange associado à restrição $\sum_{i=1}^{p} u_i = 1$.

Como o sistema (2.2) tem dimensão $(p+1)\times(p+1)$ e p é razoavelmente pequeno, podemos empregar técnicas diretas para obter sua solução, tais como a decomposição QR da matriz dos coeficientes $\begin{bmatrix} A^T A & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix}$.

2.2 MDIIS: *Modified Direct Inversion of the Iterative Subspace*

Neste seção, propomos uma nova técnica de aceleração do algoritmo Levenberg-Marquardt para minimização de somas de quadrados baseada nas informações da função objetivo nas últimas p aproximações. A essa técnica demos o nome de MDIIS, acrônimo de *Modified Direct Inversion of the Iterative Subspace*, em alusão à estratégia de aceleração proposta por Pulay.

Mais especificamente, dadas p aproximações $x^1, x^2, \ldots, x^p$ da solução x^* do problema (1.2), definimos a aproximação x^a de x^* como combinação linear dessas aproximações, ou seja,

$$x^a = \sum_{i=1}^{p} u_i x^i. \tag{2.3}$$

Na prática, supondo que estamos na iteração k de nosso algoritmo principal e que $k \geq p$, calculamos o passo acelerado usando as aproximações $x^{k-p+i},\ i = 1, \ldots, p$ geradas nas últimas p iterações. Entretanto, nesta seção, omitiremos o termo $k - p$ dos índices para simplificar a notação.

Na iteração k, os coeficientes $u_1, u_2, \ldots, u_p$, da combinação linear (2.3) são obtidos resolvendo-se o problema de quadrados mínimos

$$\begin{aligned} min \quad & \frac{1}{2}\left\|\sum_{i=1}^{p} u_i F(x^i)\right\|^2 \\ suj.\ a \quad & \sum_{i=1}^{p} u_i = c_k. \end{aligned} \tag{2.4}$$

onde $F(x^j) = [f_1(x^j)\ f_2(x^j)\ \ldots\ f_m(x^j)]^T,\ j = 1, \ldots, p,$ e c_k é uma constante real, não nula, escolhida convenientemente.

A principal motivação desta abordagem reside no fato particular de que se F for linear e se a solução x^* puder ser escrita como combição linear do vetores $x^i, i = 1, \ldots, p$, estaremos resolvendo o problema (1.2) diretamente, a menos de uma busca linear, uma vez que os coeficientes do subproblema acelerado seriam a solução de

$$\begin{aligned} min \quad & \frac{1}{2}\|F(u_1 x^1 + u_2 x^2 + \cdots + u_p x^p)\|^2 \\ suj.\ a \quad & \sum_{i=1}^{p} u_i = c_k. \end{aligned} \tag{2.5}$$

Vamos analisar, agora, o papel da restrição do subproblema (2.4). Antes disso, entretanto, convém simplificar nossa notação. Para tanto, definamos a matriz $A = [F(x^1)\ F(x^2)\ \ldots\ F(x^p)]$ e o vetor $u = [u_1\ u_2\ \ldots\ u_p]^T$, composto pelas variáveis que desejamos encontrar. Neste caso, o problema (2.4) é equivalente a

$$\begin{aligned} min \quad & h(u) = \frac{1}{2}||Au||^2 \\ suj.\ a \quad & u^T \mathbf{1} = c_k. \end{aligned} \tag{2.6}$$

A solução do problema (2.6) é obtida resolvendo-se o sistema

$$\begin{bmatrix} A^T A & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} u \\ \overline{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ c_k \end{bmatrix}, \tag{2.7}$$

onde $\overline{\lambda}$ é o multiplicador de Lagrange associado à restrição $u^T\mathbf{1} = c_k$.

Se o problema (2.6) possui solução única para alguma constante c_k não nula, o Lema 2.1 abaixo indica que não adianta alterar o valor da constante c_k, uma vez que a solução é única a menos de um escalamento das variáveis.

Lema 2.1 *Seja u^c uma solução de (2.6) para $c_k = c \neq 0$ e seja h_c o valor ótimo associado. Então, para qualquer $\gamma \in \mathbb{R}$, podemos encontrar uma solução u^γ de (2.6) com $c_k = \gamma$ tal que*

$$\gamma u^c = c u^\gamma. \tag{2.8}$$

Além disso, se h_γ é o valor ótimo correspondente a u^γ, então

$$\gamma^2 h_c = c^2 h_\gamma. \tag{2.9}$$

Prova: O caso em que $\gamma = 0$ é trivial. Suponhamos que $\gamma \neq 0$. Definamos $u^\gamma = \frac{\gamma}{c} u^c$. Neste caso,

$$\sum_{i=1}^{p} u_i^\gamma = \frac{\gamma}{c} \sum_{i=1}^{p} u_i^c = \frac{\gamma}{c} c = \gamma.$$

Além disso,

$$h_d \equiv h(u^\gamma) = \frac{1}{2}||Au^\gamma||^2 = \left[\frac{\gamma}{c}\right]^2 \frac{1}{2}||Au^c||^2 = \left[\frac{\gamma}{c}\right]^2 h_c.$$

Por outro lado, u^γ é factível para (2.6) com $c_k = \gamma$. Se $\overline{u}^*$ é uma solução de (2.6) com $c_k = \gamma$, então $h(\overline{u}^*) \leq h_\gamma$, ou ainda

$$\frac{\gamma^2}{c^2} h_c \geq h(\overline{u}^*). \tag{2.10}$$

Agora, definamos $v = \frac{c}{\gamma}\overline{u}^*$. Neste caso, v é factível para (2.6) com $c_k = c$, de forma que $h(v) \geq h_c$. Mas, $h(v) = \frac{c^2}{\gamma^2} h(\overline{u}^*)$, de onde obtemos

$$h_c \leq \frac{c^2}{\gamma^2} h(\overline{u}^*). \tag{2.11}$$

De (2.10) e (2.11), segue que

$$h_\gamma = \frac{\gamma^2}{c^2} h_c = h(\overline{u}^*).$$

Assim, u^γ também é solução do problema convexo (2.6) com $c_k = \gamma$, já que tem o mesmo valor funcional que a solução $\overline{u}^*$.■

Corolário 2.1 *Se $\gamma = -c$ no lema anterior, então podemos encontrar $u^\gamma = -u^c$ com $h_c = h_\gamma$.*

Prova: A demonstração pode ser obtida trivialmente substituindo γ por $-c$ no Lema 2.1.■

O corolário abaixo indica que não vale a pena ampliar o conjunto factível, visto que a solução sempre pertencerá à sua fronteira.

Corolário 2.2 *A solução* $u^* = [u_1^* \; u_2^* \; \ldots \; u_p^*]^T$ *do problema*

$$\begin{aligned} min \quad & h(u) = \frac{1}{2}||Au||^2 \\ suj.\ a \quad & u^T\mathbf{1} \geq c_k. \end{aligned} \tag{2.12}$$

satisfaz exatamente a restrição $u^T\mathbf{1} = c_k$.

Prova: Sejam u^Q e u^R soluções de (2.6) e de (2.12), respectivamente. Vamos supor que $\sum_i^p u_i^R = \gamma > c$. Então, pelo Lema 2.1, podemos encontrar uma solução $\overline{u}^Q$ de (2.6) tal que

$$\overline{u}^Q = \frac{c}{\gamma}u^R \tag{2.13}$$

e

$$h(u^Q) = h(\overline{u}^Q) = \frac{c^2}{\gamma^2}h(u^R). \tag{2.14}$$

Mas $c < \gamma$, de modo que

$$h(u^Q) < h(u^R). \tag{2.15}$$

Como $\Omega_Q = \{u \; : \; \sum_i^p u_i = c\} \subset \Omega_R = \{u \; : \; \sum_i^p u_i \geq c\}$, segue que

$$h(u^R) \leq h(u^Q). \tag{2.16}$$

De (2.15) e (2.16), obtemos uma contradição. Assim, a solução de (2.12) satisfaz exatamente a condição $\sum_i^p u_i^R = c$.■

O Lema 2.1 e os Corolários 2.1 e 2.2 sugerem que a aproximação x^a será sempre bem determinada, a menos de um fator de multiplicação. Com efeito, se X é a matriz cuja j-ésima coluna é formada pelo vetor x^j, podemos escrever a solução do problema (2.6) com $c_k = c$ como

$$x^a = Xu^c.$$

Se $\overline{x}^a = Xu^\gamma$ é solução do problema (2.6) com $c_k = \gamma$, pelo Lema 2.1, podemos tomar $u^\gamma = \frac{c}{\gamma}u^c$, e teremos

$$\overline{x}^a = \frac{c}{\gamma}x^a.$$

Apesar de x^a estar bem determinado, o fator de multiplicação interfere na escolha da direção $d^a = x^a - x$ (em que x é a última aproximação aceita da solução x^*), como

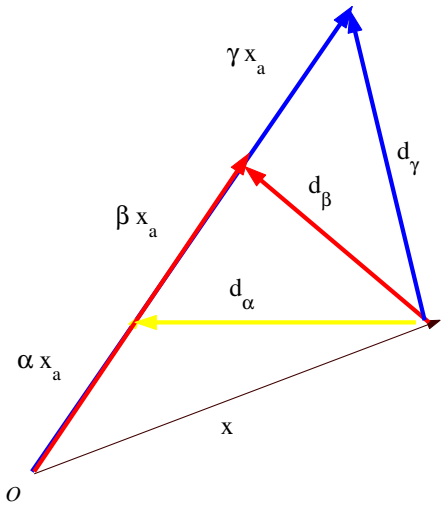

Figura 2.1: Como a escolha de c_k altera a direção acelerada $d^a = x^a - x$.

mostra a Figura 2.1. Em algoritmos que utilizam busca linear, a escolha da direção de busca é fundamental para obter uma boa taxa de convergência. Desta forma, a escolha da constante c_k, a cada iteração, é essencial no processo de aceleração proposto.

Nos testes que realizamos, a determinação do vetor u através de (2.6) com $c_k = 1$ teve um bom desempenho quando aplicada à reslução de problemas de minimização irrestrita na forma (1.2). Entretanto, resultados melhores foram obtidos por uma estratégia empírica bastante simples. Supondo que x_k é a aproximação da solução no início da k-ésima iteração e que d_k é a direção de Levenberg-Marquardt, admissível ou não, calculada nessa iteração, usamos

- $c_k = 1$, se $||x_k|| > ||d_k||$; e
- $c_k = 2$, caso contrário.

Para justificar a escolha acima, lembramos que $||x^a|| \leq c_k \max_{1 \leq i \leq p} ||x^i||$. Inspirados nessa relação, se a norma da direção de busca é menor do que a norma da aproximação corrente, tentamos estimar uma melhor aproximação x^a de modo que sua norma seja menor ou igual à maior norma entre as aproximações $x^1, \ldots, x^p$ consideradas. Por outro lado, se a norma de d_k for maior do que a norma de x_k, tentamos aumentar a norma da aproximação acelerada, aumentando a constante c_k.

O desempenho do algoritmo acelerado depende não só do critério de escolha da constante c_k, como também de p, o número de pontos usados para determinar o subespaço iterativo do processo de aceleração. Na prática, não é recomendado usar um valor de p muito alto, pois o aumento do esforço computacional não é compensado por uma

melhoria na qualidade da solução. Experimentos numéricos sugerem que o número de pontos considerados no processo de aceleração deve ser tomado como o mínimo entre n e p, tal que $3 \leq p \leq 12$. Esses valores não oneram de maneira excessiva o tempo de execução do algoritmo.

Terminamos este capítulo apresentando uma breve análise do desempenho prático das estratégias de aceleração. Em nossos testes, utilizamos os mesmos 35 problemas descritos em Moré, Garbow e Hillstrom [31], mencionados no Capítulo 1. Para tornar mais desafiador nosso experimento, aumentamos as dimensões dos 19 problemas com dimensão variável, conforme descrito na Tabela 2.1.

problema	n	m
Jenrich e Sampson	2	10
Gulf	3	10
Biggs EXP6	6	13
Watson	9	31
Rosenbrock Extendida	1000	1000
Powell Extendida	1000	1000
Penalidade I	200	201
Penalidade II	200	400
Função de dimensão variável [1]	200	202
Trigonométrica	1000	1000
Brown quase-linear	1000	1000
Valores no bordo discreto	1000	1000
Equação integral	1000	1000
Broyden Tridiagonal	1000	1000
Broyden Banda	1000	1000
Linear - posto completo	1000	1000
Linear - posto um	1000	1000
Linear - posto um com colunas e linhas nulas	1000	1000
Chebyquad	80	80

Tabela 2.1: Dimensão (n) e número (m) de funções coordenadas usados nos problemas de dimensão variável.

Neste experimento, utilizamos o algoritmo de Levenberg-Marquardt apresentado por Nielsen [33]. O passo acelerado é aceito em duas situações:

- se a solução aproximada x^{LM} proveniente do método de Levenberg-Marquardt provoca um decréscimo suficiente no valor da função objetivo, aceitamos x^a se $f(x^a) < f(x^{LM})$;

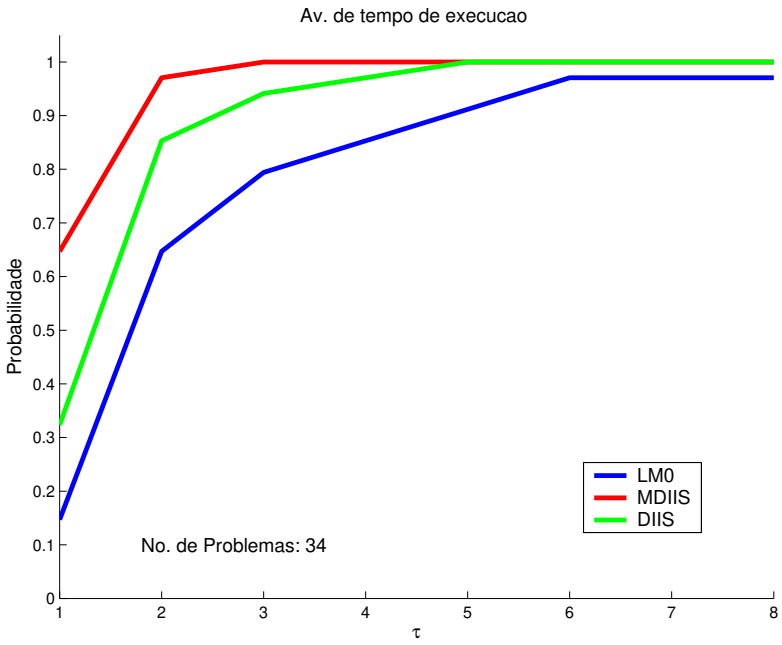

Figura 2.2: Perfil de desempenho para o Algoritmo com e sem aceleração, avaliando o tempo de execução

- se a solução aproximada x^{LM} é rejeitada, aceitamos x^a se este ponto satisfaz uma condição de decréscimo suficiente do tipo Armijo, usando como referência a aproximação corrente da solução.

Na Figura 2.2, usamos o gráfico do perfil de desempenho para comparar o tempo de execução do algoritmo de Levenberg-Marquardt sem aceleração (LMO) com o tempo consumido pelos algoritmos com direção LM acelerada pelo processo de Pulay (DIIS) e pela nova proposta de aceleração (MDIIS), fixando $p = 5$.

Como podemos ver, a nova proposta de aceleração é mais eficiente do que o algoritmo LM sem aceleração e do que o algoritmo que usa a estratégia DIIS. A aceleração também tornou o algoritmo mais robusto para este grupo de problemas. Apenas para um dos problemas, todos os algoritmos pararam por falta de decréscimo no valor da função objetivo, sendo DIIS o algoritmo que apresentou o menor valor funcional.

O resultado acima nos motiva a incluir nossa técnica de aceleração em GENCAN [4], com o objetivo de minimizar a soma de quadrados utilizando a direção de Levenberg-Marquardt nas faces ativas, em lugar da direção de Newton truncada.

Capítulo 3

Algoritmo para a minimização de somas de quadrados de funções em caixas

Nos capítulos e seções anteriores, apresentamos alguns dos ingredientes necessários para a definição do algoritmo principal desta tese, cujo objetivo é a minimização da soma de quadrados de funções suaves com restrições de canalização, problema que representamos na forma

$$\min_{x\in\Omega} f(x) = \frac{1}{2}\sum_{j=1}^{m} \overline{f}_j^2(x), \tag{3.1}$$

onde $\Omega = \{x \in \mathbb{R}^n : l \leq x \leq u\}$ é uma caixa compacta em $\mathbb{R}^n$, e as desigualdades entre vetores são entendidas componente a componente.

Nosso algoritmo é baseado em uma estratégia de restrições ativas similar ao método GENCAN, descrito em Birgin e Martínez [4]. Entretanto, duas características distinguem os algoritmos, ambas relacionadas ao processo de minimização dentro da face ativa. A primeira é que, em lugar da direção de Newton truncada, utilizamos a direção de Levenberg-Marquardt. A segunda é que empregamos um novo processo de aceleração para corrigir o passo LM.

Antes de apresentarmos o algoritmo, mostraremos, na próxima seção, como é definida e identificada uma face ativa em Ω, bem como descrevemos as entidades vetoriais, tais como gradientes e Hessianas, à ela associadas.

3.1 Definições

Seja $x \in \Omega$ a aproximação atual da solução x^* do problema (3.1). Definimos a função identificadora $\delta(x, \Omega) \equiv \delta : \{1, 2, \ldots, n\} \to \{-1, 0, 1\}$ como

$$\delta(i) = \begin{cases} -1, & \text{se} \quad x_i = l_i; \\ 0, & \text{se} \quad l_i < x_i < u_i; \\ 1, & \text{se} \quad x_i = u_i. \end{cases} \tag{3.2}$$

A função δ determina os índices das componentes de x que estão na fronteira da caixa Ω e, conseqüentemente, os índices daquelas que são estritamente internas à caixa. Com efeito, os conjuntos

$$I = \{i \in \mathbb{N} \cap [1, n] : |\delta(i)| = 1\} \tag{3.3}$$

e

$$I^{\perp} = \{i \in \mathbb{N} \cap [1, n] : |\delta(i)| \neq 1\} \tag{3.4}$$

contêm, respectivamente, os índices das componentes de x que atingem a fronteira de Ω e os índices das componentes de x que satisfazem as desigualdades $l_i < x_i < u_i$.

Como os conjuntos I e $I^{\perp}$ são mutuamente excludentes, e $I \cup I^{\perp} = \{1, 2, \ldots, n\}$, podemos decompor o espaço $\mathbb{R}^n$ em uma soma direta de dois espaços ortogonais que são completamente determinados pelos conjuntos I e $I^{\perp}$.

Exemplo 3.1 *Seja* $\Omega = \{x \in \mathbb{R}^3 : l_i \leq x_i \leq u_i,\ i = 1, 2, 3\}$. *Supondo que* $x = (x_1, u_2, u_3)$, *com* $l_1 < x_1 < u_1$, *temos*

$$\delta(1) = 0,\ \delta(2) = 1 \quad e \quad \delta(3) = 1$$

e

$$I = \{2, 3\}.$$

Desta forma, a face ativa pode ser descrita por

$$\mathcal{F}_{ex_1} = \{x \in \mathbb{R}^3 :\ l_1 < x_1 < u_1,\ x_2 = u_2 \ e \ x_3 = u_3\}.$$

Enquanto o seu fecho é dado por

$$\overline{\mathcal{F}}_{ex_1} = \{x \in \mathbb{R}^3 :\ l_1 \leq x_1 \leq u_1,\ x_2 = u_2 \ e \ x_3 = u_3\}.$$

A Figura 3.1 mostra a face ativa para o exemplo acima.

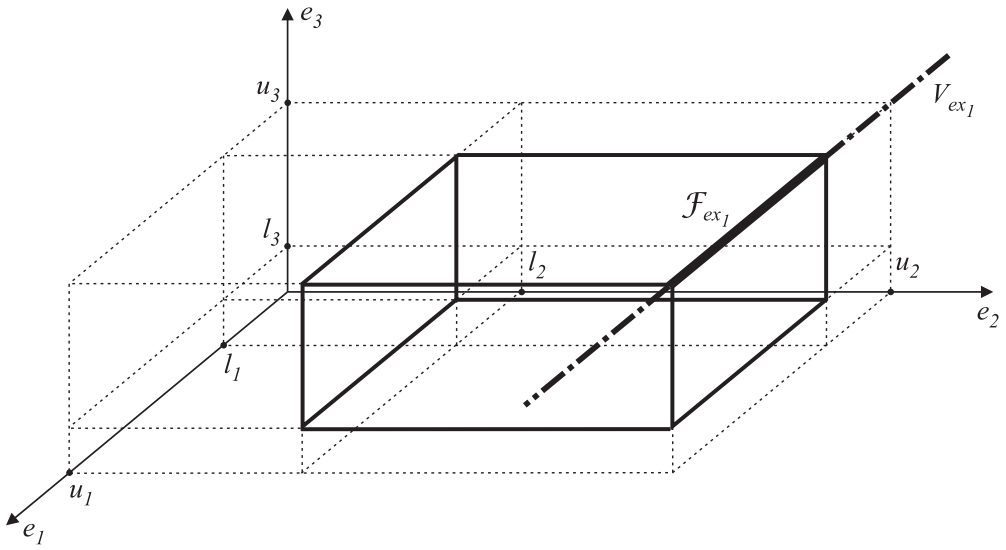

Figura 3.1: Caixa do Exemplo 3.1, destacando a face ativa, $\mathcal{F}_{ex_1}$, e a variedade afim a ela associada, V_{ex_1}.

Exemplo 3.2 *Seja Ω como no exemplo 3.1 e $x = (x_1, u_2, x_3)$ com $l_i < x_i < u_i,\ i = 1, 3$. Neste caso,*

$$\delta(1) = 0,\ \delta(2) = 1 \quad e \quad \delta(3) = 0$$

e

$$I = \{2\}.$$

A face ativa (ver Figura 3.2) é descrita por

$$\mathcal{F}_{ex_2} = \{x \in \mathbb{R}^3 :\ l_i < x_i < u_i,\ i = 1, 3 \quad e \quad x_2 = u_2\}.$$

Definição 3.1 *A dimensão de uma face $\mathcal{F}$ em um ponto x é a dimensão do menor subespaço afim V de $\mathbb{R}^n$ que a contém.*

Assim, no exemplo 3.1, a dimensão de $\mathcal{F}$ é 1, enquanto que, no exemplo 3.2, a face tem dimensão igual a 2. Com efeito, as variedades afins associadas a cada face são, respectivamente,

$$V_{ex_1} = x + span\{e_1\}$$

e

$$V_{ex_2} = x + span\{e_1, e_3\},$$

onde e_i é o i-ésimo vetor canônico de $\mathbb{R}^n$.

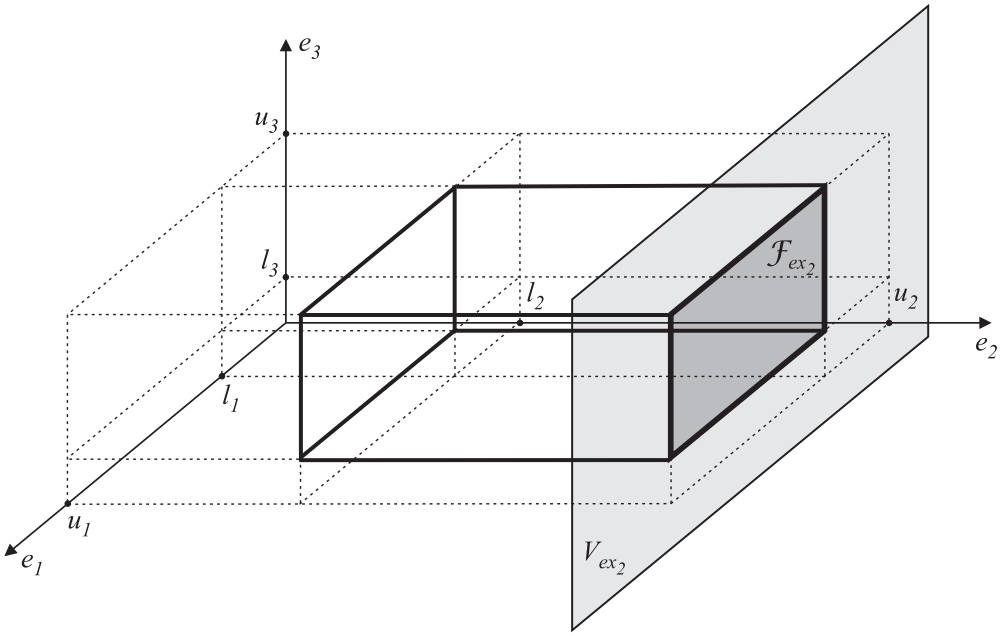

Figura 3.2: Caixa do Exemplo 3.2. $\mathcal{F}_{ex_2}$ é a face ativa, e V_{ex_2}, a variedade afim a ela associada.

Definição 3.2 *Uma face é chamada degenerada quando a sua dimensão é zero.*

Exemplo 3.3 *Se* $\Omega = \{x \in \mathbb{R}^3 : l_i \leq x_i \leq u_i\}$ *e* $x = (l_1, u_1, l_2)$*, então*

$$\delta(1) = -1, \delta(2) = 1 \text{ e } \delta(3) = -1.$$

O conjunto dos índices ativos é dado por

$$I = \{1, 2, 3\}$$

e a face ativa é definida por

$$\mathcal{F} = \{x\}.$$

De modo geral, se I é o conjunto de índices ativos em x com relação à caixa Ω, definimos a face ativa em x por

$$\mathcal{F}_I = \{x \in \Omega : x_i = l_i \text{ se } \delta(i) = -1; x_i = u_i \text{ se } \delta(i) = 1; l_i < x_i < u_i \text{ se } i \in I^\perp\}. \quad (3.5)$$

Definimos o fecho da face $\mathcal{F}_I$ por

$$\overline{\mathcal{F}}_I = \{x \in \Omega : x_i = l_i \text{ se } \delta(i) = -1; x_i = u_i \text{ se } \delta(i) = 1; l_i \leq x_i \leq u_i \text{ se } i \in I^\perp\}. \quad (3.6)$$

Também usamos o termo V_I para representar a menor variedade afim que contém $\mathcal{F}_I$, e $S_I = span\{e_i,\ i \in I^\perp\}$ para indicar o subespaço paralelo a V_I.

Observe que podemos definir V_I usando a regra

$$V_I = \{y \in \mathbb{R}^n : y = x + z : z \in S_I\}. \tag{3.7}$$

Como todo vetor $w \in V_I$ possui as componentes $w_i,\ i \in I$, fixas, podemos identificá-lo a partir das variáveis livres. Ou seja, se $\overline{n}$ é o número de variáveis livres de x com relação a Ω, e se $\overline{\Omega}$ é o subconjunto de Ω em que se considera apenas as variáveis livres, podemos associar a cada $w \in V_I$ um vetor $\overline{w} \in \mathbb{R}^{\overline{n}}$. Isto equivale a definir a aplicação bijetora projeção

$$\begin{aligned} \varphi_x : \Omega \subset \mathbb{R}^n &\rightarrow \overline{\Omega} \subset \mathbb{R}^{\overline{n}} && (3.8)\\ w &\rightarrow \overline{w} = [w_{j_1}\ w_{j_2}\ \ldots\ w_{j_{\overline{n}}}]^T, && (3.9) \end{aligned}$$

com $j_l \in I^\perp$ ordenado de modo que, se $s < t$, então $j_s < j_t$.

A inversa de φ_x é definida por

$$\begin{aligned} \varphi_x^{-1} : \overline{\Omega} \subset \mathbb{R}^{\overline{n}} &\rightarrow \Omega \subset \mathbb{R}^n \\ \overline{w} &\rightarrow w, \end{aligned}$$

com

$$w_i = \begin{cases} w_{j_i}, & \text{se} \quad j_i \in I^\perp; \\ x_i, & \text{se} \quad i \in I. \end{cases}$$

Exemplo 3.4 *Sejam $x \in \mathbb{R}^5$ e $w \in V_I$ com $\varphi_x(w) = y = (y_1, y_2, y_3)$. Se $I = \{1, 4\}$ e $I^\perp = \{2, 3, 5\}$, então*

$$\varphi_x^{-1}(y) = (x_1, y_1, y_2, x_4, y_3).$$

Dado o vetor $x \in \mathcal{F}_I$, aproximação atual da solução x^* de (3.1), queremos somar a x outro vetor d, tal que a nova aproximação

$$x^+ = x + d \tag{3.10}$$

esteja em $\overline{\mathcal{F}}_I$ e seja tal que $f(x^+) < f(x)$.

Como nos algoritmos SQP, determinamos a direção d como uma aproximação da solução do subproblema quadrático aproximante

$$\begin{aligned} min \quad & q(d) = f + g^T d + \frac{1}{2} d^T B d \\ suj.\ a \quad & l \leq x + d \leq u \\ & x + d \in V_I. \end{aligned} \tag{3.11}$$

onde f é o valor da função calculada em x, g é o gradiente de f em x e B é uma aproximação da matriz Hessiana de f em x.

Como $x + d$ deve pertencer a $\overline{\mathcal{F}}_I \subset V_I$, necessariamente devemos ter $d_i = 0$ para $i \in I$. Assim, o produto interno e o produto entre matriz e vetor na soma (3.11) são reduzidos a

$$g^T d = \sum_{i=1}^{\overline{n}} g_{j_i} d_{j_i}, \text{ com } j_i \in I^{\perp} \tag{3.12}$$

e

$$d^T B d = \sum_{i=1}^{\overline{n}} \sum_{j=1}^{\overline{n}} B_{k_i,k_j} d_{k_i} d_{k_j}, \text{ com } k_i, k_j \in I^{\perp}. \tag{3.13}$$

As identidades (3.12) e (3.13) só dependem das componentes do gradiente g e da aproximação B da Hessiana de f correspondentes aos índices das variáveis livres de x. Com isto, em lugar de resolvermos o subproblema (3.11) no espaço original de dimensão n, resolvemos o problema

$$\begin{aligned} min \quad & \overline{q}(d) = f + \overline{g}^T d + \frac{1}{2} d^T \overline{B} d \\ suj.\ a \quad & l \leq x + d \leq u \\ & x + d \in V_I. \end{aligned} \tag{3.14}$$

onde $\overline{g}$ é o vetor gradiente reduzido (obtido usando apenas as componentes de g que correspondem às variavéis livres de x) e $\overline{B}$ é a submatriz de B formada pelas linhas e colunas que correspondem aos índices das variáveis livres de x.

A direção d resultante do problema (3.14) é calculada pelo método de Levenberg-Marquardt, o que garante que ela seja uma direção de descida suficiente.

Na próxima seção vamos definir o algoritmo que age sobre a subvariedade afim V_I, que contém a face ativa $\mathcal{F}_I \equiv \mathcal{F}_I(x)$.

3.2 Algoritmo interno à face

Suponha que conheçamos x_k, uma aproximação da solução do problema (3.1). Neste caso, escrevemos I_k para representar o conjunto dos índices ativos em x_k. De maneira análoga, $\mathcal{F}_k$ é a face ativa em x_k, V_k é a menor variedade afim contendo $\mathcal{F}_k$ e S_k é o subespaço paralelo a V_k.

Como trabalhamos com um método de restições ativas, precisamos definir um algoritmo iterativo para minimização na face ativa $\mathcal{F}_k$, que parta de um ponto factível no interior da face e que convirja para um ponto estacionário no seu interior ou atinja a sua fronteira, obtendo um descréscimo do valor da função objetivo.

Encerramos a aplicação do algoritmo na face ativa quando não conseguimos obter um decréscimo do valor da função objetivo, o que ocorre quando não conseguimos reduzir o valor da quadrática aproximante, isto é, quando detectamos um passo muito pequeno, ou quando encontramos um novo conjunto de índices ativos. O abandono da face é feito utilizando o algoritmo do gradiente espectral projetado (ver [5] e [4]), descrito na seção 3.4.

Voltando à minimização na face, como as componentes correspondentes aos índices ativos I_k estão fixas e só consideramos os índices correspondentes às variáveis livres de x_k, o problema se resume a encontrar o minimizador de f restrito à variedade afim V_k, ou seja, encontrar uma solução aproximada do problema

$$\min_{x \in \overline{\Omega} \subset \mathbb{R}^{\overline{n}}} f(x), \tag{3.15}$$

com $f \equiv f|_{V_k}$ e $\overline{n} = dim(S_k)$. O conjunto

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

representa a face fechada de Ω na qual se considera apenas as variáveis livres.

O algoritmo que propomos é baseado no método descrito em Birgin e Martínez [4], embora contenha duas diferenças fundamentais. A primeira é que usamos o critério de decréscimo suficiente como nos métodos de regiões de confiança, para decidir quando aceitamos uma direção d de descida, com condições tipo Armijo quando executamos buscas lineares ao longo de d, onde empregamos a direção de Levenberg-Marquardt em substituição da direção de Newton truncada. A segunda, é a inclusão da técnica MDIIS proposta no Capítulo 2, seção 2.2, para correção da direção de busca.

O método de Levenberg-Marquardt foi escolhido porque sempre podemos conseguir uma direção de descida suficiente, e também por estarmos resolvendo problemas de quadrados mínimos, ou seja, de somas de quadrados de funções. Todavia, a direção LM é computacionalmente cara de se obter e , além disso, o vetor direção pode ter uma norma muito pequena no caso do parâmetro de Levenberg-Marquardt ser muito grande. Por isso, executamos uma busca linear ao longo de d_k, com o objetivo de acelerar a convergência do nosso algoritmo.

Como em Birgin e Martínez [4], dadas uma aproximação x_k da solução de (3.15) e uma direção de descida d_k, satisfazendo a condição de ângulo

$$g_k^T d_k \leq -\theta ||g_k|| ||d_k||, \tag{3.16}$$

definimos

$$x_{k+1}^t = x_k + \alpha_k d_k,$$

onde, inicialmente, α_k é o valor mínimo entre o passo unitário e o maior passo que pode ser dado ao longo da direção d_k de modo que o vetor x_{k+1}^t pertença a $\overline{\mathcal{F}}_k$.

Em seguida, tentamos acelerar, usando uma das técnicas descritas no Capítulo 2. O algoritmo que define a direção acelerada pode ser resumido nos seguintes passos.

Algoritmo 3.1 *Suponha que estamos em uma iteração k e que x_k é aproximação corrente da solução do problema. Suponha ainda que são dadas as últimas $\overline{p} = min\{k+1, p-1\}$ aproximações $x^{k-\overline{p}+i},\ i = 1, 2, \ldots, \overline{p}$. Defina x_{k+1}^t como a $(\overline{p}+1)$-ésima aproximação (que, por simplicidade, denotamos por x_{k+1}).*

P1 Encontre u^*, solução do problema

$$\begin{aligned} min \quad & \frac{1}{2}\left\|\sum_{i=1}^{\overline{p}+1} u_i F(x^{k-\overline{p}+i})\right\|^2 \\ suj.\ a \quad & \sum_{i=1}^{\overline{p}+1} u_i = c_k. \end{aligned} \tag{3.17}$$

P2 $\overline{x} = \sum_{i=1}^{\overline{p}+1} u_i^* x^{k-\overline{p}+i}$.

P3 $d_k^a = \overline{x} - x_k$.

Assim, a solução acelerada é dada pelo vetor

$$x_{k+1}^a = x_k + \alpha_a(\overline{x} - x_k),$$

onde α_a é definido de forma análoga a α_k.

Em cada iteração do método de Levenberg-Marquardt, só aceitamos o passo do processo de aceleração se a direção acelerada $d_k^a = x_{k+1}^a - x_k$ satisfaz a condição do ângulo

$$g_k^T d_k^a \leq -\theta ||g_k|| ||d_k^a||.$$

Se tal condição não é atendida, encerramos o processo de aceleração naquela iteração e continuamos o algoritmo LM normalmente, tentando acelerar na iteração seguinte.

Uma vez aceita a direção d_k^a, resta-nos decidir se x_{k+1}^a será aceito. Para tanto, dividimos nossa análise em dois casos:

- Se x_{k+1}^aatinge a fronteira da caixa $\overline{\Omega}$, verificamos se $f(x_{k+1}^a) < min\{f(x_k), f(x_{k+1}^t)\}$. Se isto ocorre, aceitamos $d_k = d_k^a$ como nova direção de busca. Caso contrário, descartamos a aproximação acelerada e utilizamos a direção de LM.

- Se x^a_{k+1} está no interior da caixa, temos duas outras possibilidades. Caso x^t_{k+1} já provoque um decréscimo suficiente no valor da função objetivo, adotamos $d_k = d^a_k$ como nova direção de busca se $f(x^a_{k+1}) < f(x^t_{k+1})$. Se x^t_{k+1} não reduz o valor da função, usamos uma condição tipo Armijo para decidir se aceitamos ou não a aproximação acelerada, isto é, verificamos se $f(x^a_{k+1}) \leq f_k + \gamma g_k^T(x^a_{k+1} - x_k)$. Caso isso aconteça, aceitamos a direção acelerada como nova direção de busca.

Note que apenas aceitamos a aproximação acelerada quando reduzimos o valor da função objetivo. No caso em que o processo de aceleração não gera uma aproximação que reduz o valor da função, adotamos a direção de LM. Se x_{k+1} não reduz o valor da função objetivo, tentamos obter uma nova aproximação com valor funcional menor através do processo de redução do passo e aumentamos o parâmetro Levenberg-Marquardt, λ_k, de modo a aumentar as chances de se satisfazer a condição de decréscimo na próxima iteração.

Uma vez obtida a direção de busca d_k, sendo ela a direção de Levenberg-Marquardt ou a direção acelerada, definimos $x_{k+1} = x_k + d_k$. Assim com Birgin e Martínez [4], quando x_{k+1} satisfaz a condição de descréscimo suficiente do valor da função objetivo e é interior à caixa, analisamos se vale a pena extrapolar ao longo da direção, o que é feito verificando se a derivada direcional $g_{k+1}^T d_k$ é menor do que uma fração de $g_k^T d_k$. Neste caso, d_k ainda é uma direção de descida com relação a x_{k+1} e está "longe" de ser ortogonal a g_{k+1}.

Com efeito, a condição

$$g_{k+1}^T d_k < \beta g_k^T d_k \leq -\theta\eta ||g_k|| ||d_k||, \quad 0 < \beta < 1, \tag{3.18}$$

indica que d_k satisfaz uma "condição de ângulo relaxada". No caso em que a condição (3.18) não é satisfeita, definimos x_{k+1} como nova aproximação e reiniciamos o algoritmo.

Por outro lado, se x_{k+1} atingiu a fronteira da caixa e tem um valor funcional menor que x_k, tentamos adicionar o maior número possível de restrições ao conjunto de índices ativos. Isto é feito iterativamente, aumentando o passo α_k e projetando o vetor resultante na caixa. Este processo é encerrado quando uma das projeções possui um valor funcional maior do que sua antecessora ou quando duas projeções consecutivas estão muito próximas.

Apresentamos abaixo o algoritmo de minimização em cada face ativa. No que segue, denotamos por $P(x) = P_{\overline{\Omega}}(x)$ a projeção Euclidiana de x sobre a caixa $\overline{\Omega}$, onde

$$P_{\overline{\Omega}}(x) = \begin{cases} \overline{l}_i, & \text{se} \quad x_i \leq \overline{l}_i; \\ x_i, & \text{se} \quad \overline{l}_i < x_i < \overline{u}_i; \\ \overline{u}_i, & \text{se} \quad x_i \geq \overline{u}_i; \end{cases}$$

e consideramos

$$\overline{q}_k(d) = f_k + \overline{g}_k^T d + \frac{1}{2} d^T \overline{H} d \tag{3.19}$$

como sendo o modelo quadrático que aproxima f em x_k, restrito a V_k. Para simplificar a notação, definimos $f_k \equiv f(x_k)$, $f_{k+1}^t \equiv f(x_{k+1}^t)$ e $f_{k+1}^a \equiv f(x_{k+1}^a)$.

Algoritmo 3.2 *Algoritmo para minimização nas faces, com busca linear. O algoritmo começa com* $x_0 \in \overline{\Omega}$. *Devem ser fornecidos os parâmetros* $\gamma \in (0,1)$, $\beta \in (0,1), \theta \in (0,1), \eta \in (0,1), N > 1$ *e* $0 < \tau_1 < \tau_2 < 1$, *além das tolerâncias* ε_{abs}, $\varepsilon_{rel} > 0$. *Inicialmente,* $k = 0$.

P1 Cálculo da direção de busca

- **P1.1** Se $||\overline{g}_k|| = 0$, termine a execução do algoritmo.
- **P1.2** Usando o método LM (Algoritmo 3.3), calcule $d_k \in S_k$ tal que $g_k^T d_k \leq -\theta ||g_k|| ||d_k||$

P2 Decisões da busca linear

- **P2.1** $\alpha_{max} = \max\{\alpha > 0 : [x_k,\, x_k + \alpha d_k] \in \overline{\Omega}\}$
- **P2.2** $\alpha_k = \min\{\alpha_{max},\, 1\}$
- **P2.3** $x_{k+1}^t = x_k + \alpha_k d_k$
- **P2.4** $ared_k = f_k - f_{k+1}^t$
- **P2.5** $pred_k = q_k(0) - q_k(d_k)$
- **P2.6** Se $pred_k \leq \varepsilon_{abs}$, termine a execução do algoritmo
- **P2.7** $\rho = ared_k / pred_k$
- **P2.8** Se $\rho_k > \eta$,
  - **P2.8.1** $acxt = 1$
  - **P2.8.2** $\lambda_{k+1} = \lambda_k \max\{1/3,\, 1 - (2\rho_k - 1)^3\}$
- **P2.9** Senão,
  - **P2.9.1** $acxt = 0$
  - **P2.9.2** $\lambda_{k+1} = 2\lambda_k$
- **P2.10** Obtenha d_k^a pelo Algoritmo 3.1.
- **P2.11** $c_{max} = \max\{\alpha > 0 : [x_k,\, x_k + \alpha d_k^a] \in \overline{\Omega}\}$
- **P2.12** $c_k = \min\{c_{max},\, 1\}$

P2.13 $x^a_{k+1} = x_k + c_k d^a_k$

P2.14 Se $acxt = 1$ e $f^a_{k+1} < f^t_{k+1}$,
ou se $acxt = 0$, $x^a_{k+1} \in \text{Int } \overline{\Omega}$ e $f^a_{k+1} \leq f_k + \gamma c_k \overline{g}^T d^a_k$,
ou ainda se $acxt = 0$, $x^a_{k+1} \in \overline{\Omega} - \text{Int } \overline{\Omega}$ e $f^a_{k+1} < f_k$,

P2.14.1 $x^t_{k+1} = x^a_{k+1}$

P2.14.2 $d_k = d^a_k$

P2.14.3 $\alpha_k = c_k$

P2.14.4 $\alpha_{max} = c_{max}$

P2.14.5 $acxt = 1$

P2.15 Se $acxt \neq 1$, siga para **P4**

P2.16 Se $\alpha_k = 1$ (neste caso, $x_k + d_k \in Int(\overline{\Omega})$) e $\overline{g}^T_{k+1} d_k \geq \beta \overline{g}^T_k d_k$

P2.16.1 $x_{k+1} = x_k + d_k$

P2.16.2 Siga para **P5**

P3 Extrapolação

P3.1 Se $\alpha_k < \alpha_{max}$, $\alpha_{trial} = \min\{N\alpha_k, \alpha_{max}\}$

P3.2 Senão, $\alpha_{trial} = N\alpha_k$

P3.3 Se $\alpha_k \geq \alpha_{max}$ e
$||P(x_k + \alpha_{trial} d_k) - P(x_k + \alpha_k d_k)||_\infty < \max\{\varepsilon_{abs}, \varepsilon_{rel}||P(x_k + \alpha_k d_k)||_\infty\}$,

P3.3.1 $x_{k+1} = P(x_k + \alpha_k d_k)$

P3.3.2 Termine a execução do algoritmo

P3.4 Se $f(P(x_k + \alpha_{trial} d_k)) \geq f(P(x_k + \alpha_k d_k))$,

P3.4.1 $x_{k+1} = P(x_k + \alpha_k d_k)$

P3.4.2 Siga para **P5**

P3.5 Senão,

P3.5.1 $\alpha_k = \alpha_{trial}$

P3.5.2 Volte a **P3.1**

P4 Processo de redução do passo

P4.1 Repita

P4.1.1 Calcule $\alpha_{new} \in [\tau_1 \alpha_k, \tau_2 \alpha_k]$

P4.1.2 $\alpha_k = \alpha_{new}$

P4.2 Até que $f(x_k + \alpha_k d_k) \leq f(x_k) + \gamma \alpha_k \overline{g}_k^T d_k$

P4.3 $x_{k+1} = x_k + \alpha_k d_k$

P5 Fim do laço

P5.1 Se $\alpha_k = \alpha_{max}$, termine a execução do algoritmo

P5.2 $k = k + 1$

P5.3 Volte a **P1**

Como se observa no passo 2.8.2, quando a redução real da função objetivo é satisfatória, λ_{k+1} é calculado segundo a fórmula de Nielsen (1.46), tomando $\beta = \nu = 2$, $\mu = 3$ e $p = 3$. Por outro lado, se não há uma redução suficiente de f, λ_{k+1} é igual ao dobro de λ_k.

Em P2.14, o passo acelerado é aceito se:

1. x_{k+1}^t satisfaz a condição de descréscimo suficiente

$$\rho_k > \eta \tag{3.20}$$

   e a condição $f_{k+1}^a < f_{k+1}^t$. Neste caso, o passo acelerado proporciona um decréscimo da função objetivo ainda maior do que o fornecido por x_{k+1}^t;

2. a condição (3.20) não é satisfeita por x_{k+1}^t, mas x_{k+1}^a, que pertence ao interior da caixa, fornece um decréscimo suficiente da função objetivo;

3. a condição (3.20) não é satisfeita, mas x_{k+1}^a, que pertence a fronteira da caixa, apresenta função objetivo com valor inferior ao obtido por x_{k+1}^t.

Os passos P2.1-P2.14 estão associados exclusivamente à escolha da direção de busca e ao tamanho do passo máximo que pode ser dado sem sair da caixa. Após executar esses passos, caso x_{k+1}^t ainda não tenha sido aceito, executamos uma busca linear, que só é encerrada quando satisfazemos a condição de decréscimo de Armijo

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \gamma \alpha_k \overline{g}_k^T d_k. \tag{3.21}$$

Executamos o processo de redução do passo em lugar de aumentar o parâmetro λ_k e reiniciar o processo de encontrar uma direção que forneça o decréscimo desejado porque, em geral, é mais barato calcular alguns valores funcionais do que um vetor direção, uma vez que este exige a resolução de sistemas lineares.

Por outro lado, se x_{k+1}^t não está no interior da caixa $\overline{\Omega}$ e parece vantajoso caminhar ao longo da direção definida por d_k, isto é, se $f(x_k + \alpha_{max} d_k) < f(x_k)$, então extrapolamos, multiplicando o passo por um fator N, fixo, e projetando o ponto resultante na caixa.

Além disso, também extrapolamos quando $x_k + d_k$ está no interior da caixa, a condição de descréscimo (3.20) é satisfeita e a condição da derivada direcional

$$\overline{g}_{k+1}^T d_k \geq \beta \overline{g}_k^T d_k$$

(definida no passo P2.16) não se verifica. Neste caso, a extrapolação pode produzir um decréscimo no valor da função objetivo, uma vez que d_k pode ser uma direção de descida suficiente para x_{k+1} (ver Figura 3.3).

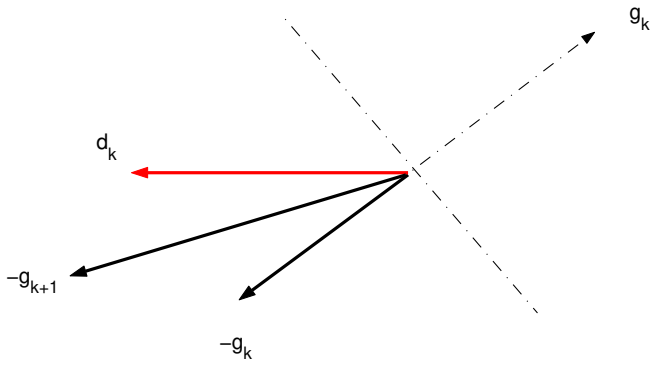

Figura 3.3: Exemplo em que d_k continua sendo uma direção de descida suficiente para x_{k+1} e $g_{k+1}^T d_k < g_k^T d_k$.

Assim como em GENCAN [4], no processo de extrapolação, tentamos sucessivas projeções de $x_k + \alpha d_k$ sobre a caixa $\overline{\Omega}$, aumentando o passo α por um fator N fixo. Se o ponto $x_k + \alpha d_k$ está no interior da caixa, mas $x_k + \alpha N d_k$ não está, testamos o ponto $x_k + \alpha_{max} d_k$. O passo é aumentado até obtermos $f(P(x_k + \alpha_{trial} d_k)) \geq f(P(x_k + \alpha d_k))$. Neste caso, aceitamos o último ponto, $x_{k+1} = P(x_k + \alpha d_k)$. Também encerramos o processo de extrapolação quando a distância d_e entre duas projeções consecutivas é muito pequena, conforme ilustrado na Figura 3.4.

O Algoritmo 3.2 acaba quando um iterando x_{k+1} pertence a fronteira da caixa Ω, tendo reduzido o valor da função objetivo, ou quando a norma do gradiente projetado interno é muito pequena.

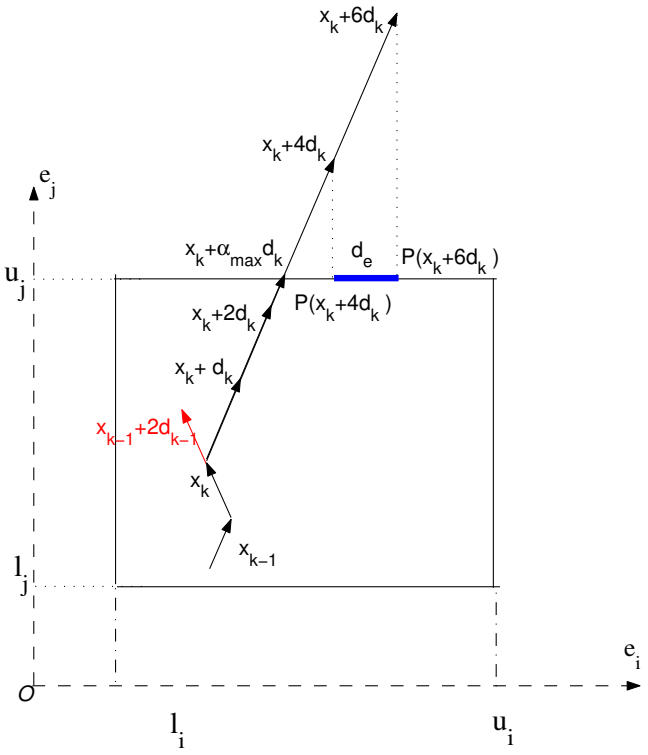

Figura 3.4: Exemplo do processo de extrapolação.

Assim como é feito para a maioria dos algoritmos que empregam regiões de confiança e o método de Newton modificado, usaremos o lema de Powell, reproduzido abaixo, para mostrar que a seqüência dos gradientes gerados pelo Algoritmo 3.2 converge a zero.

Lema 3.1 *(Powell) Considere o problema*

$$\min_{||d||\leq\Delta} q_k(d) = g_k^T d + \frac{1}{2} d^T B_k d. \tag{3.22}$$

Se d^* é uma solução de (3.22), então

$$q_k(0) - q_k(d^*) \geq \frac{1}{2}||g_k|| \min\left\{\Delta, \frac{||g_k||}{||B_k||}\right\}. \tag{3.23}$$

Prova: Ver Nocedal e Yuan [34].■

O Lema 3.1 fornece uma estimativa da redução de q. Esta estimativa é utilizada na demonstração do Teorema 3.2 apresentado ao final desta seção.

As hipóteses necessárias para análise de convergência do Algoritmo 3.2 são

H1 $f \in \mathcal{C}^2$.

H2 $||J_k|| \leq \sigma,\ \forall k$ e para algum $\sigma > 0$.

H3 $\sup_k ||\nabla_k^2 f - B_k|| < M,$ para algum $M > 0$.

O teorema seguinte mostra que o número de iterações consecutivas do processo de redução do passo é sempre finito.

Teorema 3.1 *Qualquer subconjunto K_C da seqüência $\{x_k\}$ formado por iterandos consecutivos que não satisfazem (3.20) no Algoritmo 3.2 é finito.*

Prova: Suponha, por absurdo, que K_C é infinito. Neste caso, a partir de uma iteração $\overline{k}$, a condição (3.20) é sempre rejeitada e atualizamos o parâmetro LM usando $\lambda_{k+1} = 2\lambda_k$. Desse modo, $\lambda_k \to \infty$. Agora, usando o Teorema do Valor Médio, temos

$$ared_k = f_k - f_{k+1} = -g_k^T d_k - \frac{1}{2} d_k^T \nabla_k^2 f(x_k + \theta d_k) d_k, \ \theta \in (0,1) \tag{3.24}$$

e

$$pred_k = q_k(0) - q_k(d_k) = -g_k^T d_k - \frac{1}{2} d_k^T B_k d_k. \tag{3.25}$$

Por simplicidade, vamos escrever $\nabla_k^2 f(x_k + \theta d_k) \equiv \nabla_k^2 f(\theta)$. Assim,

$$ared_k - pred_k = -\frac{1}{2} d_k^T (\nabla_k^2 f(\theta) - B_k) d_k. \tag{3.26}$$

Dividindo ambos os lados da equação (3.26) por $pred_k$, e usando o fato de que $pred_k \geq \lambda_k ||d_k||^2$ (ver (1.20)), obtemos

$$\left| \frac{ared_k}{pred_k} - 1 \right| \leq \frac{1}{2\lambda_k} \sup_k ||\nabla_k^2 f(\theta) - B_k||. \tag{3.27}$$

Como $\lambda_k \to \infty$, temos $||d_k|| \to 0$. Além disso, a continuidade de $\nabla^2 f$ nos garante que $\nabla_k^2 f(\theta) \approx \nabla_k^2 f$. Pela hipótese **H3**, $\sup_k ||\nabla_k^2 f - B_k||$ é limitado, o que implica que $\rho_k = \frac{ared_k}{pred_k} \to 1$. Conseqüentemente, a condição (3.20) vai valer para algum k muito grande, contradizendo a hipótese de que K_C é infinito.■

Provaremos, agora, que qualquer seqüência gerada pelo Algoritmo 3.2 pára em um ponto estacionário irrestrito, ou pára na fronteira da caixa $\overline{\Omega}$, ou gera, no limite, pontos estacionários irrestritos. Parte da demonstração do teorema é similar à encontrada em Birgin e Martínez [4]. Mesmo assim, resolvemos apresentá-la integralmente.

Teorema 3.2 *O Algoritmo 3.2 está bem definido. Se $\{x_k\}$ é uma sequência gerada pelo Algoritmo 3.2, uma das seguintes possibilidades acontece:*

1. *A sequência $\{x_k\}$ pára na fronteira $\partial\overline{\Omega} = \overline{\Omega} - Int\overline{\Omega}$.*
2. *A sequência $\{x_k\}$ gera uma sequência $\{f_k\} = \{f(x_k)\}$ estritamente decrescente.*

3. *A sequência pára em x_k, com $||\overline{g}(x_k)|| = 0$.*

4. *A sequência $\{x_k\}$ é infinita, tendo pelo menos um ponto limite x^* que satisfaz $||\overline{g}(x^*)|| = 0$.*

Prova: Provemos, primeiramente, que o Algoritmo 3.2 está bem definido. Com efeito, o laço do passo P4 é o processo clássico de redução do passo e é finito pelos já bem conhecidos argumentos da derivada direcional (ver Dennis - Schnabel [15]). O *loop* do passo P3 também é finito porque nós apenas multiplicamos uma direção não nula por um um número maior do que um, ou tomamos o passo máximo factível permitido. Neste caso, após um número finito de passos, ou a fronteira é atingida ou a condição de descréscimo estipulada no passo P3.4 é satisfeita. Claramente, o algoritmo exige que $f(x_k + \alpha_k d_k) < f(x_k)$, logo a sequência $\{f_k\}$ é estritamente decrescente.

Como a sequência $\{x_k\}$ está contida no conjunto compacto $\Omega \subset \mathbb{R}^n$, pelo Teorema de Cauchy ela possui pelo menos um ponto limite. Resta provar que todo ponto limite x^* da seqüência $\{x_k\}$ satisfaz $||\overline{g}(x^*)|| = 0$.

Seja $K_1 \subset \mathbb{N}$ um subconjunto infinito do conjunto dos números naturais, tal que

$$\lim_{k \in K_1} x_k = x^*. \tag{3.28}$$

Definamos $d_k = x_{k+1} - x_k$, para todo $k \in K_1$. Vamos dividir a prova em dois casos:
Caso A. Existe $\rho > 0$ tal que

$$||d_k|| > \rho, \ \forall k \in K_1. \tag{3.29}$$

Caso B. Existe $K_2 \subset K_1$ tal que $\lim_{k \in K_2} ||d_k|| = 0$.

Analisemos inicialmente o caso A. Pelo Teorema 3.1, podemos supor, sem perda de generalidade (senão passamos para uma subseqüência), que a seqüência $\{x_k\}$ gerada pelo Algoritmo 3.2 sempre satisfaz (3.20) ou a condição

$$f_{k+1}^a \leq f_k + \gamma c_k \overline{g}^T d_k^a,$$

definida no passo P2.14 do Algoritmo 3.2. Por outro lado, se o número de iterandos que satisfazem a condição (3.20) fosse finito, esta condição falharia sempre a partir de um certo inteiro $\overline{k}$. Deste modo, o passo P2.9.2 do algoritmo implicaria que $\lambda \to +\infty$. Conseqüentemente, teríamos uma contradição. Assim, existe um conjunto infinito de índices $k \in K_1$ tal que x_k sempre satisfaz (3.20). Porém, vamos supor, para não sobrecarregar a notação, que a própria seqüência $\{x_k\}$ sempre satisfaz (3.20), isto é, esta condição é verdadeira para cada $k \in K_1$.

Suponhamos, então, que a seqüência $\{x_k\}_{k \in K_1}$ seja tal que

$$f_k - f_{k+1} \geq \eta \ pred_k, \tag{3.30}$$

em que $pred_k$ é calculado a partir da direção LM $d_k^t = x_{k+1}^t - x_k$.

Usando (3.29) e definindo $\Delta_k = ||d_k||$, temos que $\Delta_k > \rho$, $k \in K_1$. Pelo Lema 1.2, item (ii), s_k resolve o problema

$$\min_{||d|| \leq \Delta_k} q_k(d) = g_k^T d + \frac{1}{2} d^T B_k d. \tag{3.31}$$

Assim, pelo Lema 3.1, segue que

$$f_k - f_{k+1} \geq \frac{1}{2}\eta ||g_k|| \min\left\{ ||d_k||, \frac{||g_k||}{||B_k||} \right\}. \tag{3.32}$$

Uma vez que $B_k = J_k^T J_k$ e que a hipótese **H2** é satisfeita, existe $A > 0$ tal que $||B_k|| \leq A$, $\forall k \in K_1$. Logo,

$$f_k - f_{k+1} \geq \frac{1}{2A}\eta ||g_k|| \min\{\rho A, ||g_k||\}. \tag{3.33}$$

Como o lado esquerdo da última desigualdade vai a zero, pois $\{f_k\}$ é decrescente, limitada inferiormente e f é contínua, concluímos que $||g_k||_{k \in K_1}$ também tende a zero.

A prova do caso B é inteiramente análoga àquela dada em Birgin e Martínez [4]. Entretanto, vamos nos permitir escrevê-la para completar a demonstração.

Suponha que existe $K_2 \subset K_1$ tal que $\lim\limits_{\substack{k \to +\infty \\ k \in K_2}} ||d_k|| = 0$.

Seja $K_3 \subset K_2$ o conjunto tal que α_k é calculado no Passo P2.16.1 para todo $k \in K_3$. Analogamente, seja $K_4 \subset K_2$ o conjunto dos índices das iterações em que α_k é calculado pelo Passo P3 e seja $K_5 \subset K_2$ o conjunto tal que α_k é calculado pelo passo P4 para todo $k \in K_5$. Vamos considerar três possibilidades:

(i) K_3 é infinito;

(ii) K_4 é infinito;

(iii) K_5 é infinito.

Consideremos, inicialmente, o caso (i). A condição imposta no passo P2.16 nos garante que

$$\overline{g}(x_k + d_k)^T d_k \geq \beta \overline{g}_k^T d_k,$$

de modo que

$$\overline{g}(x_k + d_k)^T \frac{d_k}{||d_k||} \geq \beta \overline{g}_k^T \frac{d_k}{||d_k||}, \tag{3.34}$$

para todo $k \in K_3$. Como K_3 é infinito, podemos tomar uma subsequência convergente $d_k/||d_k|| \to d$, passar o limite em (3.34) e usar a continuidade do gradiente, de modo a obter

$$\overline{g}(x_*)^T d \geq \beta \overline{g}(x_*)^T d. \tag{3.35}$$

Uma vez que $0 < \beta < 1$, a desigualdade (3.35) implica que $\overline{g}(x_*)^T d \geq 0$. Por (3.16) e por continuidade, temos

$$\overline{g}(x_*)^T d \leq -\theta ||\overline{g}(x_*)||.$$

Logo, $||\overline{g}(x_*)|| = 0$.

Consideremos, agora, o caso (ii). Como estamos supondo que K_4 é infinito, o processo de extrapolação só é executado no interior da caixa. Ou seja, $P(x_k + \alpha_k d_k) = x_k + \alpha_k d_k$. Além disso, pelo passo P3.1, $\alpha_{trial} \leq \alpha_k N$ e $P(x_k + \alpha_{trial} d_k) = x_k + \alpha_{trial} d_k$. Assim, para todo $k \in K_4$, escrevendo $\alpha'_k = \alpha_{trial}$, temos que $\alpha'_k \in [\alpha_k, \alpha_k N]$ e

$$f(x_k + \alpha'_k d_k) \geq f(x_k + \alpha_k d_k). \tag{3.36}$$

Portanto, pelo Teorema do Valor Médio, para todo $k \in K_4$, existe $\xi_k \in [\alpha_k, \alpha'_k]$ tal que

$$\overline{g}(x_k + \xi_k d_k)^T (\alpha'_k - \alpha_k) d_k \geq 0. \tag{3.37}$$

Logo, uma vez que $\alpha'_k > \alpha_k$ para todo $k \in K_4$, temos

$$\overline{g}(x_k + \xi_k d_k)^T d_k \geq 0.$$

Como $||\alpha_k d_k||$ tende a zero (α_k é limitado pelo passo máximo permitido na caixa, α_{max}), e $||\alpha'_k d_k||$ também tende a zero (α'_k é limitado pelo produto entre um número real limitado N e α_{max}), temos $||\xi_k d_k|| \to 0$. Logo, dividindo por $||d_k||$ e tomando uma subseqüência convergente de $d_k/||d_k||$, obtemos

$$\overline{g}(x_*)^T d \geq 0.$$

Agora, usando (3.16), tomando o limite e considerando a continuidade do gradiente, segue que $\overline{g}(x_*)^T d \leq -\theta ||\overline{g}(x_*)||$. Isto implica que $||\overline{g}(x_*)|| = 0$.

Resta-nos analisar o caso em que K_5 é infinito. Neste caso, para todo $k \in K_5$, existe d'_k tal que

$$f(x_k + d'_k) \geq f(x_k) + \gamma \overline{g}(x_k)^T d'_k \tag{3.38}$$

e

$$||d'_k|| \leq \frac{||d_k||}{\tau_1}. \tag{3.39}$$

Por (3.39), $\lim_{k \in K_5} ||d'_k|| = 0$. Além disso, por (3.38),

$$f(x_k + d'_k) - f(x_k) \geq \gamma \overline{g}(x_k)^T d'_k, \tag{3.40}$$

para todo $k \in K_5$. Logo, pelo Teorema do Valor Médio, existe $\xi_k \in [0, 1]$, tal que

$$\overline{g}(x_k + \xi_k d'_k)^T d'_k \geq \gamma \overline{g}(x_k)^T d'_k, \quad \forall k \in K_5.$$

Dividindo por $||d_k^{'}||$ e tomando o limite para uma subseqüência de $d_k^{'}/||d_k^{'}||$ convergente a d, obtemos

$$\overline{g}(x_*)^T d \geq \gamma \overline{g}(x_*)^T d.$$

Esta desigualdade é similar a (3.35), de modo que, usando os mesmos argumentos, concluímos a tese do teorema.■

3.3 Implementação do algoritmo interno à face

O algoritmo irrestrito utilizado dentro das faces exige que, a cada iteração k, determinemos uma direção de busca d_k que seja uma direção de descida suficiente e que satisfaça a condição de ângulo

$$g_k^T d_k \leq -\theta ||g_k|| ||d_k||, \tag{3.41}$$

com $\theta \in (0, 1)$.

O algoritmo GENCAN [4] usa a direção de Newton truncada, calculada pelo método dos gradientes conjugados. Em nosso algoritmo, empregamos a direção Levenberg-Marquardt, que é comprovadamente uma direção de descida suficiente. Além disso, é possível fazer com que a direção LM satisfaça a condição do ângulo (3.41), uma vez que, quando o parâmetro LM λ_k tende ao infinito, a direção LM associada tende continuamente à direção de máxima descida (ver Levenberg [26]). Assim, por continuidade, existe um λ_k que torna admissível a direção LM.

Quando o algoritmo principal é iniciado, e sempre que adotamos uma iteração do gradiente projetado espectral, atribuímos ao parâmetro λ_k um valor pequeno, na tentativa de usar a direção Gauss-Newton. Este valor de λ_k é determinado seguindo as idéias de Nielsen [33]. Mais precisamente, tomamos

$$\lambda_k = \max\{\varepsilon_{abs} \max_{i=1,\overline{n}}\{|(B_k)_{ii}|\}, \varepsilon_{rel}\},$$

onde $\overline{n}$ é a dimensão do subespaço em que trabalhamos, $(B_k)_{ij}$ é o elemento da linha i e da coluna j da matriz B_k e $\varepsilon_{rel}, \varepsilon_{abs}$ são valores fixos não nulos. Esta salvaguarda é necessária para que a fórmula de atualização $\lambda_{k+1} = 2\lambda_k$ faça sentido. Doravante, para simplificar a notação, escrevemos n em lugar de $\overline{n}$.

Se, para algum λ_k não nulo, a matriz $B_k + \lambda_k I$ não é definida positiva ou se a direção correspondente não satisfaz a condição (3.41), dobramos o valor de λ_k até que as duas condições sejam satisfeitas simultaneamente.

Apresentamos abaixo o algoritmo que calcula uma direção LM admissível. Neste algoritmo, k indica a iteração atual. Já a variável *ackey* indica o tipo de passo adotado

na iteração $k-1$. Se $ackey = 0$, uma iteração gradiente projetado espectral (SPG) foi executada. Por outro lado, se $ackey = 1$, trabalhamos dentro da face.

Algoritmo 3.3 *Direção de decréscimo suficiente*

P1 Se $k = 0$ ou $ackey = 0$, $\lambda_k = \max\{\varepsilon_{abs} \max_{i=1,\overline{n}}\{|(B_k)_{ii}|\}, \varepsilon_{rel}\}$

P2 Se $B_k + \lambda_k I$ é definida positiva,

P2.1 Resolva o sistema $(B_k + \lambda_k I)d_k = -g_k$, obtendo d_k

P2.2 Se $g_k^T d_k \leq -\theta||g_k||||d_k||$, termine a execução do algoritmo

P3 Se $\lambda_k < \varepsilon_{rel}$, $\lambda_k = \max\{\varepsilon_{abs} \max_{i=1,\overline{n}}\{|(B_k)_{ii}|\}, \varepsilon_{rel}\}$

P4 Senão, $\lambda_k = 2\lambda_k$

P5 Volte a **P2**

No Algoritmo 3.3 acima, reiniciamos o parâmetro LM sempre que utilizamos uma iteração do gradiente espectral projetado. Dessa forma, sugerimos uma salvaguarda da aproximação inicial do parâmetro LM relativamente pequena em detrimento do parâmetro inicial adotado em Levenberg [26] e Marquardt [28]. Aqui utilizamos $\varepsilon_{rel} = 10^{-10}$.

O lema abaixo mostra que sempre é possível encontrar um parâmetro λ_k tal que a direção d_k, obtida pelo Algoritmo 3.3, seja uma direção de descida e satisfaça a condição do ângulo (3.41). Mais que isso, é possível encontrar uma estimativa para tal parâmetro com base nos autovalores $\xi_1 \geq \xi_2 \geq \cdots \geq \xi_n \geq 0$ da matriz simétrica B_k.

Lema 3.2 *Seja dado $\theta \in (0,1)$. Qualquer que seja $\lambda \geq \max\{\frac{\theta\xi_1 - \xi_n}{1-\theta}, 0\}$, a direção d dada pelo sistema*

$$(B_k + \lambda I)d = -g_k$$

é uma direção de descida e satisfaz (3.41).

Prova: Seja $\lambda \geq \frac{\theta\xi_1 - \xi_n}{1-\theta}$. Dado que $1 - \theta > 0$, podemos escrever

$$\lambda(1-\theta) \geq \theta\xi_1 - \xi_n.$$

Reordenando esta desigualdade, obtemos

$$\xi_n + \lambda \geq \theta(\xi_1 + \lambda),$$

de onde segue que

$$\frac{\xi_n + \lambda}{(\xi_1 + \lambda)} \geq \theta. \tag{3.42}$$

Além disso, é fácil ver que, se $\lambda > -\xi_n$, então a matriz $B_k + \lambda I$ é definida positiva. Seja, então, $B_k = U_k^T \Sigma_k U_k$ a decomposição espectral da matriz B_k, em que U_k é uma matriz ortogonal e $\Sigma_k = diag(\xi_1, \xi_2, \ldots, \xi_n)$ é uma matriz diagonal formada pelos autovalores de B_k. Neste caso,

$$\begin{aligned} ||g_k||^2 &= ||(B_k + \lambda I)d||^2 = ||U_k^T(\Sigma_k + \lambda I)U_k)d||^2 \\ &= ||(\Sigma_k + \lambda I)w_k||^2, \end{aligned}$$

com $w_k = U_k d$ e $||w_k|| = ||d||$. Assim,

$$\begin{aligned} ||g_k||^2 &= \sum_{i=1}^{n} (\xi_i + \lambda)^2 (w_k)_i^2 \\ &\geq (\xi_n + \lambda)^2 \sum_{i=1}^{n} (w_k)_i^2 \\ &= (\xi_n + \lambda)^2 ||w_k||^2 \\ &= (\xi_n + \lambda)^2 ||d||^2. \end{aligned}$$

Portanto,

$$||g_k|| \geq (\xi_n + \lambda)||d||. \tag{3.43}$$

Por outro lado, para todo $\lambda > -\xi_n$, temos

$$\begin{aligned} g_k^T d_k &= -g_k^T (B_k + \lambda I)^{-1} g_k \\ &= -\sum_{i=1}^{n} \frac{(y_k)_i^2}{\xi_i + \lambda}, \end{aligned}$$

com $y_k = U_k g_k$ e $||y_k|| = ||g_k||$. Como $\xi_i \leq \xi_1$, para $i = 1, 2, \ldots, n$, segue que

$$g_k^T d_k \leq -\frac{1}{\xi_1 + \lambda} ||y_k||^2.$$

Ou seja,

$$g_k^T d_k \leq -\frac{||g_k||^2}{\xi_1 + \lambda}. \tag{3.44}$$

De (3.42), (3.43) e (3.44), obtemos

$$\begin{aligned} g_k^T d_k &\leq -(\frac{\xi_n + \lambda}{\xi_1 + \lambda})||g_k||||d|| \\ &\leq -\theta ||g_k||||d||. \blacksquare \end{aligned}$$

Podemos determinar o número máximo de iterações necessárias para se obter uma direção admissível pelo Algoritmo 3.3. Para tanto, denotemos por λ_k^* o parâmetro LM, gerado a partir de λ_k, tal que a matriz

$$B_k + \lambda_k^* I$$

é definida positiva e a direção d_k que resolve o sistema

$$(B_k + \lambda_k^* I)d_k = -g_k$$

satisfaça a condição (3.41). Como

$$\lambda_k^* = 2^j \lambda_k \ \text{ e } \ \lambda_k > 0,$$

para algum $j \in \mathbb{N}$, pelo Lema 3.2, devemos ter

$$\lambda_k^* = 2^j \lambda_k \geq \max\{\frac{\theta\xi_1 - \xi_n}{1-\theta}, 0\}. \tag{3.45}$$

Observe que, se $\frac{\theta\xi_1 - \xi_n}{1-\theta} \leq 0$, qualquer valor não negativo de λ_k^* é admissível. Pela regra de atualização de λ_k definida no passo P3 do Algoritmo 3.3, obtemos um tal valor positivo no máximo na segunda iteração.

Consideremos, então, que $\frac{\theta\xi_1 - \xi_n}{1-\theta} > 0$. Neste caso, com base em (3.45), obtemos

$$2^j \lambda_k \geq \frac{\theta\xi_1 - \xi_n}{1-\theta}.$$

Assim,

$$j \ \geq \ log_2(\theta\xi_1 - \xi_n) - log_2(\lambda_k(1-\theta)). \tag{3.46}$$

Logo, em no máximo $j+1$ iterações, com j satisfazendo (3.46), obtemos λ_k^* tal que a direção LM associada é de descida e satisfaz a condição do ângulo exigida.■

3.4 Algoritmo principal

Nesta seção, apresentamos o algoritmo que utilizamos para resolver o problema de minimização em caixa

$$\min_{x\in\Omega} f(x) = \frac{1}{2}||F(x)||^2, \tag{3.47}$$

com $F : \mathbb{R}^n \to \mathbb{R}^m$ e $\Omega = \{x \in \mathbb{R}^n \ : \ l \leq x \leq u\}$.

O algoritmo apresentado nesta secção é similar ao apresentado em Birgin e Martínez [4]. A diferença fundamental reside na implementação da direção de busca. Por isso,

os teoremas desta seção foram, originalmente, apresentados e demonstrados em [4], já que independem da direção de busca adotada, exigindo apenas que esta satisfaça uma condição de ângulo, condição atendida pela direção de busca do Algoritmo 3.2.

No Algoritmo 3.4, apresentado a seguir, a direção adotada a cada iteração k é definida com base no gradiente espectral projetado (contínuo) em $x_k \in \Omega$, dado por

$$g_P(x_k) = P_\Omega(x_k - \sigma_k g(x_k)) - x_k, \tag{3.48}$$

onde σ_k é o coeficiente espectral (ver Barzilai e Borwein [1] e Raydan [38]). A partir do gradiente projetado, definimos também o gradiente projetado interno (à face), que é dado por

$$g_I(x_k) = P_{V_I}(g_P(x_k)) - x_k.$$

A Figura 3.5 ilustra esses dois vetores. Na figura, supomos que x_k pertence à face lateral esquerda do paralelepípedo. Em vermelho, definimos a direção do gradiente espectral projetado na caixa e, em azul, a direção do gradiente projeto interno à face ativa.

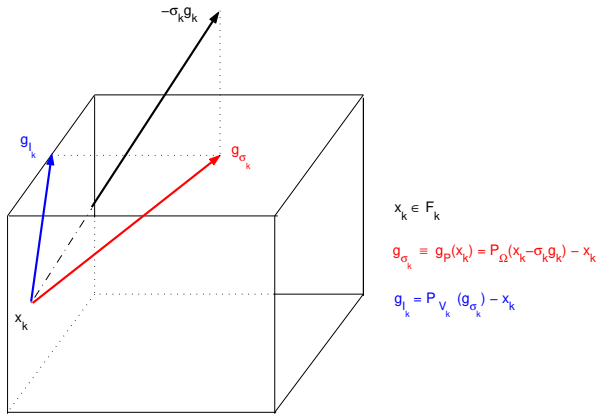

Figura 3.5: Direção do *Gradiente Projetado*

O próximo resultado, citado em Birgin, Martinez e Raydan [5], relaciona a direção do gradiente projetado com os pontos estacionários do problema (3.47) e caracteriza a direção do gradiente projetado como uma direção de descida.

Dado $x \in \Omega$, escrevemos $g \equiv g(x)$ para o gradiente de f em x, e $g_\sigma \equiv g_\sigma(x) = P_\Omega(x - \sigma g) - x$ para a direção do gradiente projetado.

Lema 3.3 *Seja Ω um subconjunto convexo, fechado e não vazio de $\mathbb{R}^n$. Seja $\overline{\sigma} > 0$ um número real dado. Para todo $x \in \Omega$ e para todo $\sigma \in [0, \overline{\sigma}]$ as seguintes asserções se verificam:*

1. $g^T g_\sigma \leq -\frac{1}{\sigma}||g_\sigma||_2^2 \leq -\frac{1}{\overline{\sigma}}||g_\sigma||_2^2$.
2. $x = P_\Omega(x - \sigma g), \ \ \forall \sigma > 0$ *se, e somente se, x é um ponto estacionário do problema (3.47).*
3. $||g_\sigma|| = 0$ *se, e somente se, x é um ponto estacionário do problema (3.47).*

Prova: Provemos o item 1. Seja $z_\sigma = P_\Omega(x - \sigma g), \sigma > 0$. Logo, $z_\sigma \in \Omega$. Pelo *Teorema B.2, item 2* (vide Apêndice B), temos

$$(z_\sigma - y)^T(z_\sigma - (x - \sigma g)) \leq 0, \ \ \forall y \in \Omega.$$

Em particular, tomando $y = x$, e observando que $g_\sigma = z_\sigma - x$, obtemos

$$g_\sigma^T(g_\sigma + \sigma g) \leq 0, \ \ \forall \sigma > 0.$$

Distribuindo o produto interno e tomando $\sigma \leq \overline{\sigma}$, obtemos

$$g_\sigma^T g \leq -\frac{1}{\sigma}||g_\sigma||^2 \leq -\frac{1}{\overline{\sigma}}||g_\sigma||^2,$$

encerrando a demostração.

Provemos agora o item 2. Por definição, x^* é um ponto estacionário se, e somente se,

$$\nabla f(x^*)^T(x - x^*) \geq 0, \ \ \forall x \in \Omega.$$

Isto é equivalente a

$$((x^* - \sigma \nabla f(x^*)) - x^*)^T(x - x^*) \geq 0, \ \ \forall x \in \Omega, \ \ \forall \sigma > 0.$$

Novamente, recorrendo ao Teorema B.2 da Projeção, item (2), isto ocorre se, e somente se, $x^* = P_\Omega(x^* - \sigma \nabla f(x^*))$, qualquer que seja $\sigma > 0$.

Passemos ao item 3. Pelo item 2 anteriormente provado, x^* é um ponto estacionário se, e somente se, $x^* = P_\Omega(x^* - \sigma \nabla f(x^*))$, qualquer que seja $\sigma > 0$, o que é equivalente a $g_\sigma(x^*) = P_\Omega(x^* - \sigma g^*) - x^* = 0, \ \ \forall \sigma > 0$.■

Abaixo, descrevemos o algoritmo definido em Birgin-Martínez [4], para minimização de uma função suave, não linear, com restrições de canalização, com a diferença de que, dentro das faces, utilizamos o Algoritmo 3.2 desenvolvido nesta tese.

Algoritmo 3.4 (GCNDLM) $\{GENCAN\}$ *O algoritmo começa com* $k = 0$ *e* $x_0 \in \Omega$. *Devem ser definidos os escalares* $\eta \in (0,1)$, $\gamma \in (0,1)$, $0 < \tau_1 < \tau_2 < 1$ *e* $0 < \sigma_{min} \leq \sigma_{max} \leq \infty$.

P0 Calcule $g_P \equiv g_P(x_0)$.

P1 $\sigma_0 = \max\{1, ||x_0||/||g_P||\}$

P2 Se $||g_P|| \leq \varepsilon$, termine a execução do algoritmo.

P3 Determine a face ativa $\mathcal{F}_I$ em x_k

P4 Calcule $g_I \equiv g_I(x_k)$.

P5 Se $||g_I|| \geq \eta||g_P||$,

P5.1 Execute uma iteração do Algoritmo 3.2.

P6 Senão (*execute uma iteração do gradiente espectral projetado*)

P6.1 $d_k = P_\Omega(x_k - \sigma_k g_k) - x_k$

P6.2 $\alpha_k = 1.0$

P6.3 $x_{k+1} = x_k + \alpha_k d_k$

P6.4 Enquanto $f(x_{k+1}) > f(x_k) + \alpha_k \gamma g_P^T d_k$,

P6.4.1 Escolha $\alpha_k \in [\tau_1 \alpha_k, \tau_2 \alpha_k]$

P6.5 $s_k = x_{k+1} - x_k$

P6.6 $y_k = g_{k+1} - g_k$

P6.7 Se $s_k^T y_k \leq 0$,

6.7.1 $\sigma_k = \max\{1, ||x_k||/||g_P||\}$

P6.8 Senão,

P6.8.1 $\sigma' = ||s_k||^2/(s_k^T y_k)$

P6.8.2 $\sigma_k = \max\{\sigma_{min},\ \min\{\sigma_{max},\ \sigma'\}\}$

P7 $k = k + 1$.

P8 Calcule $g_P \equiv g_P(x_k)$.

P9 Volte ao passo **P2**.

Antes de passarmos à análise de convergência, teçamos algumas considerações importantes sobre o algoritmo acima.

Para abandonar a face ativa, no Passo P6, aplicamos uma iteração do algoritmo apresentado em Birgin-Martínez [4], que adota como direção de busca a direção do gradiente projetado espectral $g_\sigma(x_k) = P_\Omega(x_k - \sigma g_k) - x_k$, sendo σ o *quociente de Rayleigh inverso*

$$\sigma = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T y_{k-1}},$$

introduzido por Barzilai e Borwein [1] e depois analisado por Raydan [38].

Como este quociente, também chamado de *quociente espectral*, pode ser negativo, nulo ou excessivamente grande, restringimo-lo a um intervalo $[\sigma_{min}, \sigma_{max}]$ previamente estabelecido. Vale lembrar que, para que a direção $d_k = g_\sigma(x_k)$ seja de descida, é suficiente tomar σ estritamente positivo. Por isso, escolhemos $\sigma_{min} > 0$.

Cabe observar ainda que σ_k pode ser escrito na forma

$$\sigma_k = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T H_{k-1} s_{k-1}}, \tag{3.49}$$

onde H_{k-1} é a matriz "Hessiana média"

$$H_{k-1} = \int_0^1 \nabla^2 f(x_{k-1} + t s_{k-1}) dt.$$

Como o coeficiente de Rayleigh está entre o menor e o maior autovalor da matriz associada, estamos dando um passo com informações de segunda ordem ao longo da direção de máxima descida. Tal informação torna-se visível quando notamos que a expressão dada em (3.49) é uma aproximação do minimizador da quadrática ao longo de $-g_k$ quando x_k pertence ao interior da caixa Ω. Ou seja, com esta escolha, embutimos uma espécie de busca linear ao longo da direção de máxima descida, antes de a projetarmos sobre a caixa.

Enquanto a condição de Armijo do passo P6.4 não é satisfeita, reduzimos o passo, mas continuamos a trabalhar ao longo da direção d_k. Em conseqüência, o termo $g_P^T d_k$ e a projeção sobre a caixa são calculados apenas uma vez por iteração.

O passo α ao longo da direção do gradiente projetado espectral é obtido por interpolação quadrática unidimensional. Como salvaguarda, tomamos $\alpha = \alpha/2$ quando o mínimo desta quadrática não está no intervalo $[\tau_1\alpha, \tau_2\alpha]$. Em nossos experimentos, adotamos $\tau_1 = 0.1$ e $\tau_2 = 0.9$. Cabe notar, também, que a condição de descréscimo suficiente imposta no passo P6.4 garante que a seqüência $\{x_k\}$ esteja contida no subconjunto $\Omega_0 = \{x \in \Omega \ : \ 0 \leq f(x_k) \leq f(x_0)\}$.

Observamos que, após uma iteração do gradiente projetado espectral, $x_{k+1} \notin \overline{\mathcal{F}_I}$ se $x_k \in \mathcal{F}_I$. Além disso, neste caso, $||g_I|| < \eta||g_P||$. Como as componentes de $g_I(x_k)$ e $g_P(x_k)$ correspondentes às variáveis livres são as mesmas, $g_P(x_k)$ tem componentes não nulas associadas às variáveis fixas. Portanto, $x_k + \alpha g_P(x_k) \notin \overline{\mathcal{F}_I}$ para todo $\alpha > 0$. Logo, $P_\Omega(x_k + \sigma g(x_k)) \notin \overline{\mathcal{F}_I},\ \forall \sigma > 0$. Uma vez que, de acordo com o passo P6 do algoritmo,

$$x_{k+1} = x_k + \alpha'[P_\Omega(x_k + \sigma_k g(x_k)) - x_k],$$

para algum $\alpha' > 0$, fica claro que $x_{k+1} \notin \overline{\mathcal{F}_I}$.

Os resultados teóricos apresentados a seguir provam que o Algoritmo 3.4 está bem definido e que um ponto *Karush-Kuhn-Tucker* é obtido com uma precisão pré-estabelecida. Tais resultados, estabelecidos originalmente por Birgin e Martínez em [4], também podem ser aplicados ao Algoritmo 3.4, uma vez que este só se distingue do algoritmo GENCAN no que se refere à minimização na faces ativas, mantendo a condição de ângulo exigida na teoria de convergência.

Teorema 3.3 *O Algoritmo 3.4 está bem definido.*

Prova: Isto resulta do fato de que o Algoritmo 3.2 e o Algoritmo do Gradiente Espectral Projetado (Algoritmo 3.2 em Birgin e Martínez [4]) estão bem definidos.

Teorema 3.4 *Seja $\{x_k\}$ uma seqüência infinita gerada pelo Algoritmo 3.4. Neste caso, $\{x_k\}$ possui pelo menos um ponto de acumulação.*

Prova: Isto é uma conseqüência imediata do fato de f ser limitada inferiormente por $f_{min} \geq 0$, e de toda a seqüência $\{x_k\}$ pertencer ao conjunto compacto $\Omega_0 = \{x \in \mathbb{R}^n : f_{min} \leq f(x) \leq f(x_0)\}$.■

Teorema 3.5 *Seja $\{x_k\}$ uma seqüência gerada pelo Algoritmo 3.4. Suponha que existe $\overline{k} \in \mathbb{N}$ tal que $x_k \in \mathcal{F}_I$ para todo $k \geq \overline{k}$. Então, cada ponto limite da seqüência $\{x_k\}$ é um ponto estacionário de primeira ordem do problema (3.47).*

Prova: Ver Teorema 3.2 de Birgin e Martínez [3].

Teorema 3.6 *Suponha que, para todo $x_k \in \mathcal{F}_I,\ k \in \mathbb{N}$, existe $k' > k$ tal que $x_{k'} \notin \mathcal{F}_I$. Então existe um ponto limite da sequência $\{x_k\}$ que satisfaz as condições necessárias de primeira ordem do problema (3.47).*

Prova: Ver Teorema 3.3 de [3].

Teorema 3.7 *Suponha que todos os pontos estacionários de (3.47) são não degenerados, isto é,* $\frac{\partial f}{\partial x_i} = 0$ *somente se* $l_i < x_i < u_i$*. Então as hipóteses do Teorema 3.5 (e, portanto, sua tese) se verificam.*

Prova: Ver Teorema 3.4 de [3].

O último teorema deste capítulo garante que, se uma sequência infinita é gerada pelo Algoritmo 3.4, esta possui pelo menos uma ponto limite estacionário do problema (3.47).

Teorema 3.8 *Suponha que* $\{x_k\}$ *seja uma seqüência gerada pelo Algoritmo 3.4, e que* $\varepsilon > 0$ *seja um número pequeno arbitrário. Então existe* $\overline{k} \in \{0, 1, 2, \dots\}$ *tal que*

$$||g_P(x_k)|| \leq \varepsilon, \ \forall k \geq \overline{k}.$$

Prova: Isto é uma conseqüência direta dos Teoremas 3.5 e 3.6, apresentados acima.

Capítulo 4

Experimentos numéricos

A comparação de programas para minimizar funções é sempre uma tarefa muito difícil. Como decidir, por exemplo, que programa foi melhor entre dois que conseguiram encontrar minimizadores locais distintos, sendo um com valor funcional menor, porém obtido com maior tempo de execução? Além disso, será que uma pequena porção dos problemas não está interferindo na comparação dos dados? A análise de resultados ainda depende muito do julgamento humano, o que torna difícil a obtenção de conclusões.

Recentemente, Dolan-Moré [16] desenvolveram uma função, denominada *perfil de desempenho* (ver apêndice C), que indica qual, dentre vários programas, obtem maior sucesso ao resolver um conjunto de problemas em uma margem de tempo definida. Neste capítulo, usamos o perfil de desempenho para comparar o algoritmo GENCAN original às variantes apresentadas nos capítulos anteriores.

Entretanto, o perfil de desempenho só é empregado quando os programas encontram o mesmo minimizador ou quando algum programa não encontra a solução, de modo que apresentamos os demais resultados em tabelas, para permitir que o leitor faça uma análise completa dos algoritmos avaliados.

Antes de passar aos resultados numéricos, que serão apresentados na Seção 4.3, introduzimos, nas Seções 4.1 e 4.2, a descrição dos problemas que usamos em nossos testes.

4.1 Grupo de experimentos 1

Nosso objetivo é resolver um conjunto de problemas de otimização em caixa nos quais a função objetivo é uma soma de quadrados de funções suaves. Para criar problemas dessa natureza, consideramos os últimos quinze problemas de dimensão variável descritos por Moré, Garbow e Hillstrom em [31], e introduzimos restrições de canalização utilizando as idéias apresentadas por Facchinei, Júdice e Soares em [17].

Queremos gerar um problema de otimização na forma

$$\begin{aligned} min \quad & f(x) = \frac{1}{2}\sum_{k=1}^{m} f_k^2(x) \\ suj.\ a \quad & l \leq x \leq u. \end{aligned} \tag{4.1}$$

Denotemos por $g(x)$ o gradiente de f em x. Vamos relembrar que um ponto x^* de mínimo local para (4.1) deve satisfazer

$$g_i(x^*) \begin{cases} = 0 & \text{se} \quad l_i < x_i^* < u_i, \\ \geq 0 & \text{se} \quad l_i = x_i^*, \\ \leq 0 & \text{se} \quad u_i = x_i^*. \end{cases} \tag{4.2}$$

Também podemos caracterizar um ponto estacionário de (4.1) a partir do gradiente projetado contínuo $g_P(x)$, definido em (3.48). Neste caso, x^* é estacionário se, e somente se, $g_P(x^*) = 0$.

Usando a caracterização (4.2), podemos gerar, a partir de um problema de minimização irrestrito

$$\min_{x \in \mathbb{R}^n} \overline{f}(x) = \frac{1}{2}\sum_{i=1}^{m} \overline{f}_i^2(x), \tag{4.3}$$

com solução $\overline{x}$ conhecida, um problema na forma (4.1) que tem como uma solução local x^* a solução ótima $\overline{x}$ de (4.3).

Fazemos isto escolhendo uma partição arbitrária do conjunto de índices $\{1, 2, \ldots, n\}$ em três subconjuntos L, F e U. O problema gerado será tal que $x_i^* = l_i$ para $i \in L$, $l_i \leq x_i^* \leq u_i$ para $i \in F$ e $x_i^* = u_i$ para $i \in U$. Assim, L, F e U são, respectivamente, os conjuntos de índices das variáveis ativas inferiormente, livres e ativas superiormente em x^*.

Os conjuntos de índices L e U são escolhidos de modo a determinar quais restrições serão ativas na solução x^*. Em nossos experimentos, o conjunto L foi formado pelas $\lfloor n/4 \rfloor$ primeiras componentes e o conjunto U foi formado pelas últimas $\lfloor n/4 \rfloor$ componentes do vetor x. Mais precisamente, dada a dimensão n do problema, definimos $ml = \lfloor n/4 \rfloor$ e $mu = \lfloor 3ml \rfloor$. Neste caso,

$$L = \{1, 2, \ldots, ml\}$$

e

$$U = \{mu + 1, mu + 2, \ldots, n\}.$$

Para o limite inferior da caixa Ω, tomamos

$$l_i = \begin{cases} \overline{x}_i, & \text{se} \quad i \in L, \\ \overline{x}_i - c_i, & \text{se} \quad i \in F \cup U, \end{cases} \tag{4.4}$$

com $c_i > 0$ escolhido arbitrariamente. Já para o limite superior u, utilizamos

$$u_i = \begin{cases} \overline{x}_i, & \text{se} \quad i \in U, \\ \overline{x}_i + c_i, & \text{se} \quad i \in L \cup F, \end{cases} \tag{4.5}$$

com $c_i > 0$ também escolhido arbitrariamente. Se $c_i = +\infty$, consideramos que a variável correspondente não tem limite inferior (ou superior).

Nos experimentos, tomamos $c_i = 0,75\, d(x_0, \overline{x})$, ou seja, c_i corresponde a 75% da distância entre a aproximação inicial x_0 (ainda não projetada) e a solução exata $\overline{x}$ do problema gerador (4.3).

Consideremos, agora, a função objetivo

$$f(x) = \overline{f}(x) + \sum_{i \in L} \Psi_i(x_i) + \sum_{i \in U} \Phi_i(x_i), \tag{4.6}$$

onde Ψ_i é uma função $\mathcal{C}^2\{\mathbb{R}\}$ não-decrescente, e Φ_i é uma função $\mathcal{C}^2\{\mathbb{R}\}$ não-crescente.

O gradiente de f é dado por

$$\nabla f_i(x) = \begin{cases} \nabla \overline{f}_i(x) + \nabla \Psi_i(x_i) & \text{se} \quad i \in L, \\ \nabla \overline{f}_i(x) & \text{se} \quad i \in F, \\ \nabla \overline{f}_i(x) + \nabla \Phi_i(x_i) & \text{se} \quad i \in U, \end{cases} \tag{4.7}$$

para cada $i = 1, 2, \ldots, n$. Como as funções Ψ_i e Φ_i são, respectivamente, não-decrescente e não-crescente, segue que $\nabla \Psi_i \geq 0$ e $\nabla \Phi_i \leq 0$. Portanto, de (4.2) e (4.7), concluimos que $\overline{x}$ é um mínimo local para o problema de otimização em caixa (4.1) com f definida por (4.6).

Deve-se observar que a Hessiana de f só difere da Hessiana de $\overline{f}$ nos elementos diagonais. Assim, a esparsidade da Hessiana é mantida. Por outro lado, a escolha das funções Φ_i e Ψ_i influencia diretamente o condicionamento da Hessiana de f.

Em geral, não podemos garantir que o problema gerado por esta técnica não tenha outros pontos estacionários além de $\overline{x}$. Entretanto, isto pode ser obtido se f for estritamente convexa, o que ocorre, por exemplo, quando $\overline{f}$ é estritamente convexa e as funções Φ_i e Ψ_i são convexas para $i \in L \cup U$ em Ω.

Como estamos interessados em estudar o caso em que a função objetivo é uma soma de quadrados de funções, tomamos

$$\Psi_i(x) = \begin{cases} \alpha_i^2 (x_i - \overline{x}_i + \beta_i)^2 & \text{se} \quad i \in L, \\ 0 & \text{se} \quad i \in F \cup U, \end{cases} \tag{4.8}$$

e

$$\Phi_i(x) = \begin{cases} 0 & \text{se} \quad i \in L \cup F, \\ \alpha_i^2 (x_i - \overline{x}_i + \overline{\beta}_i)^2 & \text{se} \quad i \in U, \end{cases} \tag{4.9}$$

com as constantes reais $\beta_i \geq 0$, $\overline{\beta}_i \leq 0$ e α_i escolhidas arbitrariamente. Constata-se facilmente que Ψ_i é não-decrescente no conjunto $\{x \in \mathbb{R}^n : \ x_i \geq \overline{x}_i = l_i; i \in L\}$, e Φ_i é não-crescente no conjunto $\{x \in \mathbb{R}^n : \ x_i \leq \overline{x}_i = u_i; i \in U\}$.

Com base nessas funções, geramos quatro conjuntos de problemas testes a partir dos quinze últimos problemas irrestritos, de dimensão variável, do artigo Moré, Garbow e Hillstrom [31], apresentados na Tabela 1.1. Cada conjunto de problemas é caracterizado por uma combinação das constantes $\beta_i, \overline{\beta}_i$ e α_i. Assim, para cada $i \in L \cup U$, definimos

- Conjunto de Problemas 1: $\alpha_i = \beta_i = \overline{\beta_i} = 0$.
- Conjunto de Problemas 2: $\alpha_i = 1$ e $\beta_i = \overline{\beta_i} = 0$.
- Conjunto de Problemas 3: $\alpha_i = 1$, $\beta_i = 1$ e $\overline{\beta_i} = -1$.
- Conjunto de Problemas 4: $\alpha_i = ||Jac(:, i)||$, e $\beta_i = \overline{\beta_i} = 0$, onde $Jac(:, i)$ é a i-ésima coluna da matriz Jacobiana de f.

A escolha de $\beta_i = \overline{\beta}_i = 0$ implica que o gradiente de f é idêntico ao gradiente de $\overline{f}$ na solução $x^* = \overline{x}$. Já a escolha de α_i não nulo implica na adição de elementos na forma α_i^2 à diagonal da Hessiana de $\overline{f}$, tornando a Hessiana de f melhor condicionada do que a Hessiana do problema irrestrito. Desse modo, geramos uma diversidade de problemas para que seja possível estabelecer com confiança a eficiência do novo algoritmo.

Para todos os conjuntos de testes, usamos $n = 200$ para o problema *Penalty II*, $n = 80$ para o problema *Chebyquad* e $n = 1000$ para os demais problemas.

4.2 Grupo de experimentos 2

Nesta seção, descrevemos mais três conjuntos de problemas testes que compõem nosso quadro de experimentos numéricos. Estes problemas são tratados como problemas esparsos. Os problemas foram extraídos da literatura. Alguns deles foram adaptados ao problema de minimização em caixa.

4.2.1 O problema das esferas [E1]

Este problema, apresentado por Martínez [29], consiste em encontrar p pontos, $P_1, P_2, \ldots, P_p$, pertencentes à hiperesfera de raio r definida por $\mathcal{S}^{\overline{n}-1} = \{x \in \mathbb{R}^{\overline{n}}; ||x|| \leq r\}$, tal que a distância mínima entre eles seja máxima. Isto é equivalente a exigir que o produto escalar máximo entre eles seja mínimo. Um problema equivalente de programação não

linear pode ser definido como

$$\min_{\substack{P_i \in \mathcal{S}^{\overline{n}-1} \\ i \in \{1,2,\ldots,p\}}} \frac{1}{2} \sum_{i=1}^{p} \sum_{j=i+1}^{p} \left(\frac{1}{d(P_i, P_j) + c_{ij}} \right)^2, \tag{4.10}$$

onde $d(P_i, P_j) = \|P_i - P_j\|_2$ e $c_{ij} > 0,\ i \in \{1, 2, \ldots, p\},\ j \in \{i+1, i+2, \ldots, p\}$. Os termos c_{ij} são empregados para evitar a singularidade na função objetivo.

A dimensão do problema (4.10) é $n = p\overline{n}$. Para adaptar este problema à minimização em caixas, consideramos um sistema de coordenadas esféricas generalizadas em $\mathbb{R}^{\overline{n}}$. Assim, a cada $x \in \mathcal{S}^{\overline{n}-1}$ associamos as seguintes coordenadas:

$$\begin{array}{lcll}
x_{\overline{n}} & = & \rho\,\mathrm{sen}(\theta_1)\mathrm{sen}(\theta_2)\mathrm{sen}(\theta_3)\mathrm{sen}(\theta_4) & \ldots \quad \mathrm{sen}(\theta_{\overline{n}-3})\mathrm{sen}(\theta_{\overline{n}-2})\mathrm{sen}(\theta_{\overline{n}-1}) \\
x_{\overline{n}-1} & = & \rho\,\mathrm{sen}(\theta_1)\mathrm{sen}(\theta_2)\mathrm{sen}(\theta_3)\mathrm{sen}(\theta_4) & \ldots \quad \mathrm{sen}(\theta_{\overline{n}-3})\mathrm{sen}(\theta_{\overline{n}-2})\cos(\theta_{\overline{n}-1}) \\
x_{\overline{n}-2} & = & \rho\,\mathrm{sen}(\theta_1)\mathrm{sen}(\theta_2)\mathrm{sen}(\theta_3)\mathrm{sen}(\theta_4) & \ldots \quad \mathrm{sen}(\theta_{\overline{n}-3})\cos(\theta_{\overline{n}-2}) \\
x_{\overline{n}-3} & = & \rho\,\mathrm{sen}(\theta_1)\mathrm{sen}(\theta_2)\mathrm{sen}(\theta_3)\mathrm{sen}(\theta_4) & \ldots \quad \cos(\theta_{\overline{n}-3}) \\
\vdots & & & \\
x_4 & = & \rho\,\mathrm{sen}(\theta_1)\mathrm{sen}(\theta_2)\mathrm{sen}(\theta_3)\cos(\theta_4) & \\
x_3 & = & \rho\,\mathrm{sen}(\theta_1)\mathrm{sen}(\theta_2)\cos(\theta_3) & \\
x_2 & = & \rho\,\mathrm{sen}(\theta_1)\cos(\theta_2) & \\
x_1 & = & \rho\cos(\theta_1) &
\end{array}$$

em que

$$\begin{cases}
0 \leq \rho \leq r, & \\
0 \leq \theta_i \leq \pi, & i = 1, 2, \ldots \overline{n} - 2, \\
0 \leq \theta_{\overline{n}-1} \leq 2\pi. &
\end{cases}$$

Se $\overline{n} = 3$, por exemplo, um ponto P, de coordenadas retangulares (x, y, z), é localizado por suas coordenadas esféricas habituais $(\rho, \theta_1, \theta_2)$, onde $\rho = |\overline{OP}|$, θ_2 é a medida em radianos do ângulo polar da projeção de P sobre o plano polar xy, e θ_1 é a medida em radianos não negativa do menor ângulo medido do lado positivo do eixo z à reta OP (ver Figura 4.1).

Para representar o problema das esferas de uma forma que seja possível resolvê-lo utilizando o algoritmo GENCAN e suas variantes, associamos ao conjunto de p pontos

$$P_1(\rho^1, \theta_1^1, \ldots, \theta_{\overline{n}-1}^1), \ldots, P_p(\rho^p, \theta_1^p, \ldots, \theta_{\overline{n}-1}^p)$$

um único ponto $\tilde{x} \in \mathbb{R}^{\overline{n}p}$, definido por

$$\tilde{x} = (\rho^1, \theta_1^1, \ldots, \theta_{\overline{n}-1}^1, \rho^2, \theta_1^2, \ldots, \theta_{\overline{n}-1}^2, \rho^p, \theta_1^p, \ldots, \theta_{\overline{n}-1}^p).$$

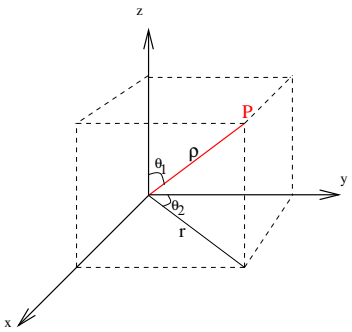

Figura 4.1: Exemplo de um sistema de coordenadas esféricas em $\mathbb{R}^3$.

Deste modo, a $j-$ésima coordenada do ponto P_i, é representada por $\tilde{x}_{(i-1)\bar{n}+j}$.

Expressamos o problema de otimização (4.10) no formato geral (4.1), definindo

$$\begin{array}{llll} f_1(x) = \hat{f}_{12}(x) & & & \\ f_2(x) = \hat{f}_{13}(x) & f_p(x) = \hat{f}_{23}(x) & & \\ f_3(x) = \hat{f}_{14}(x) & f_{p+1}(x) = \hat{f}_{24}(x) & f_{2p-2}(x) = \hat{f}_{34}(x) & \\ \vdots & \vdots & \vdots & \ddots \\ f_{p-1}(x) = \hat{f}_{1p}(x) & f_{2p-3}(x) = \hat{f}_{2p}(x) & f_{3p-6}(x) = \hat{f}_{3p} & \ldots \quad f_{(p-1)p/2}(x) = \hat{f}_{(p-1)p}(x) \end{array}$$

onde

$$\hat{f}_{ij}(x) = \frac{1}{d(P_i, P_j) + c_{ij}};\ i = 1, 2, \ldots, p;\ j = i+1, \ldots, p.$$

Para relacionar f_k à função correspondente $\hat{f}_{ij}$, usamos a fórmula implícita

$$k = (i-1)p - \frac{i(i-1)}{2} + (j-i). \tag{4.11}$$

4.2.2 O problema do Aeroporto [E2]

Considere p círculos $C_i,\ i \in \{1, 2, \ldots, p\}$, disjuntos em $\mathbb{R}^2$. Este problema, extraído de Friedlander, Martínez e Santos [22], consiste em obter um ponto (x_i, y_i) sobre cada círculo C_i, tal que a soma

$$\frac{1}{2}\sum_{i=1}^{p}\sum_{j=i+1}^{p} ||(x_i, y_i) - (x_j, y_j)||^2$$

seja mínima.

Em nossos experimentos, geramos aleatoriamente os centros $c_i = (\overline{x}_i, \overline{y}_i)$ dos círculos em $[-10, 10]^2$. Os raios, ρ_i, também foram escolhidos aleatoriamente em $[0, \frac{\rho}{2}]$, sendo ρ o mínimo das distâncias entre os diferentes centros.

Para facilitar a implementação, escrevemos cada ponto $P_i = (x_i, y_i)$ em coordenadas polares,

$$x_i = \overline{x_i} + r_i\cos(\theta_i), \tag{4.12}$$
$$y_i = \overline{y_i} + r_i\text{sen}(\theta_i), \tag{4.13}$$

onde $\theta_i \in [0, 2\pi]$ é o ângulo formado pelo ponto P_i e o eixo canônico principal com origem em c_i, e $r_i \in [0, \rho_i]$ é a distância do ponto P_i ao centro c_i. Desta forma, obtemos o seguinte problema de minimização

$$min\frac{1}{2}\sum_{i=1}^{p}\sum_{j=i+1}^{p}(\overline{x}_i + r_i\cos(\theta_i) - \overline{x}_j - r_j\cos(\theta_j))^2 + (\overline{y}_i + r_i\text{sen}(\theta_i) - \overline{y}_j - r_j\text{sen}(\theta_j))^2. \tag{4.14}$$

Utilizando a fórmula (4.11) para relacionar o índice $1 \leq k \leq p(p-1)/2$ aos índices i e j, e definindo

$$\begin{cases} f_{2k-1} = \overline{x}_{2i-1} + r_i\cos(\theta_i) - \overline{x}_{2j-1} - r_j\cos(\theta_j), \\ f_{2k} = \overline{x}_{2i} + r_i\text{sen}(\theta_i) - \overline{x}_{2j} - r_j\text{sen}(\theta_j), \end{cases}$$

podemos expressar o problema (4.14) na forma

$$\begin{array}{c} \min \\ 0 \leq \tilde{x}_{2i-1} \leq \rho_i \\ 0 \leq \tilde{x}_{2i} \leq 2\pi \\ i = 1, 2, \ldots p \end{array} \frac{1}{2}\sum_{i=1}^{p(p-1)} f_i^2(\tilde{x}), \tag{4.15}$$

onde $\tilde{x} \in \mathbb{R}^{2p}$ é dado por

$$\tilde{x} = (r_1, \theta_1, r_2, \theta_2, \ldots, r_p, \theta_p).$$

4.2.3 O problema do empacotamento de cilindros [E3]

Este problema, extraído de Birgin e Martínez [4], consiste em armazenar p cilindros iguais em um contêiner em forma de paralelepípedo. Como os cilindros têm a mesma altura, o problema se resume a distribuir os círculos C_i, $i = 1, 2, \ldots, p$, cada qual representando a base de um cilindro, no retângulo $[0, d_1] \times [0, d_2]$, de tal sorte que a interseção entre dois círculos quaisquer se dá em no máximo um ponto.

Seja $c_i = (x_i, y_i)$ o centro do círculo $C_i, i = 1, \ldots, p$ e r o raio de todos os círculos. Seguindo o modelo apresentado em Birgin-Martínez [4], o problema pode ser escrito na

forma

$$\begin{aligned} min \quad & \frac{1}{2}\sum_{i=1}^{p}\sum_{j=i+1}^{p}(\max\{0, 2r-(x_i-x_j)^2+(y_i-y_j)^2\})^2 \\ suj.\ a \quad & r \leq x_i \leq d_1 - r \\ & r \leq y_i \leq d_2 - r \end{aligned} \tag{4.16}$$

Utilizando, mais uma vez, a fórmula (4.11) para relacionar o índice $1 \leq k \leq p(p-1)/2$ aos índices i e j, e definindo

$$\tilde{x} = (x_1, y_1, x_2, y_2, \ldots, x_p, y_p)$$

e

$$f_k(\tilde{x}) = (\max\{0, 2r-(\tilde{x}_{2i-1}-\tilde{x}_{2j-1})^2-(\tilde{x}_{2i}-\tilde{x}_{2j})^2\})^2,$$

podemos expressar o problema (4.16) na forma

$$\begin{aligned} min \quad & \frac{1}{2}\sum_{k=1}^{p(p-1)/2} f_k^2(\tilde{x}) \\ suj.\ a \quad & l \leq x \leq u. \end{aligned}$$

Se o valor da função objetivo deste problema de otimização é zero, então o problema original foi resolvido.

Em nossos experimentos, adotamos $r = 0.5$, e $d_1, d_2 \in \{25, 50, 100\}$, conforme sugerido em [4].

4.3 Análise do desempenho dos algoritmos

Nesta seção, apresentamos os resultados da aplicação das variantes do algoritmo GENCAN apresentadas na Tabela 4.1 submetidos aos problemas descritos acima. Todos os programas foram implementados em FORTRAN 77 e gerados usando o compilador *g77*. Os testes foram feitos em um computador com processador Intel Pentium III, com $1Gb$ de memória RAM.

Em todos os problemas testes, usamos $p = 5$ para o processo de aceleração, uma vez que este valor foi aquele apresentou os melhores resultados de maneira geral. Consideramos que um algoritmo converge se ele pára porque o gradiente projetado possui norma Euclidiana menor do que 10^{-6} ou norma infinito menor do 10^{-5}, ou ainda porque o valor da função objetivo é menor que 10^{-20}. Naturalmente, dois programas podem convergir para minimizadores locais distintos.

Abreviatura	*Algoritmo*
GDNT	GENCAN com direção de Newton truncada, segundo Birgin e Martínez [4]
GDLM	GENCAN com direção LM sem aceleração
GAMD	GENCAN com direção LM e com a aceleração MDIIS, proposta na seção 2.2
GADP	GENCAN com direção LM e com a aceleração proposta por Pulay [37]

Tabela 4.1: Algoritmos analisados e suas abreviaturas.

Dizemos que um algoritmo diverge para um problema se atinge 1000 iterações ou se o módulo da diferença entre dois valores funcionais consecutivos não se altera em 5000 cálculos de função.

Usamos o gráfico de perfil de desempenho para comparar os problemas para os quais os algoritmos atingiram o mesmo valor funcional ou divergiram, apresentando em tabelas os resultados dos problemas para os quais os algoritmos testados apresentaram diferentes valores funcionais.

4.3.1 Resultados para o grupo de experimentos 1

Consideremos, inicialmente, os problemas do grupo de experimentos 1 (GE1). Este grupo possui 60 problemas, divididos em quatro conjuntos de mesmo tamanho, conforme descrito na Seção 4.1. A Figura 4.2 apresenta o gráfico dos perfis de desempenho dos quatro algoritmos analisados para 42 problemas desse grupo. Como se observa, o algoritmo com a nova proposta de aceleração é mais robusto e mais eficiente do que os demais algoritmos para este tipo de problemas.

Por outro lado, o algoritmo GENCAN com direção de Newton truncada (GDNT) resolveu aproximadamente 45% dos problemas no menor tempo, sendo superado apenas por GAMD, que foi o mais rápido em 50% dos casos.

Entretanto, o perfil de desempenho do algoritmo GDNT ficou bem abaixo dos perfis dos demais algoritmos. Em grande parte, esse mal desempenho de GDNT está relacionado aos problemas 13, 14 e 15. Em três dos quatro conjuntos de testes de GE1, o algoritmo não convergiu quando aplicado aos problemas 13 e 14 (*função linear com posto deficiente*). Já a dificuldade encontrada por GDNT para resolver o problema 15 (*função Chebyquad*) ocorreu porque, à diferença do que ocorre com os algoritmos que utilizam a direção LM, a versão de GENCAN que utilizamos não armazena a matriz Hessiana. Assim, quando se depara com um problema que exige um número grande de iterações do método dos gradientes conjugados truncado, o algoritmo perde desempenho.

Analisaremos, a seguir, o comportamento dos algoritmos para cada um dos conjuntos que formam o grupo GE1.

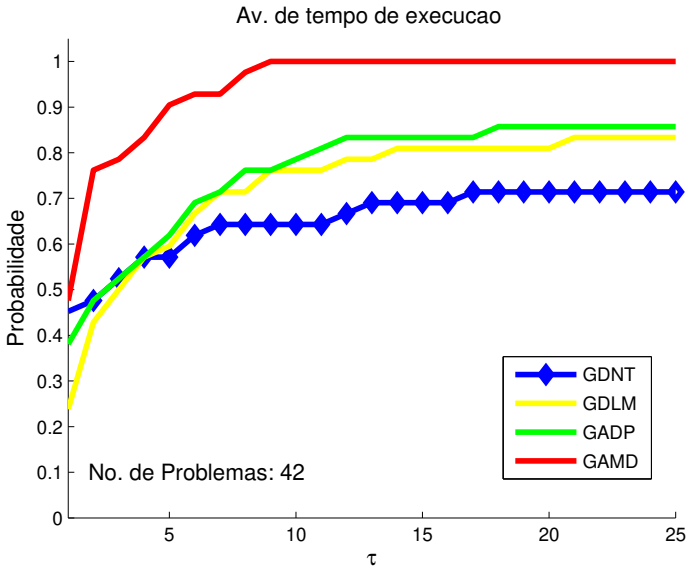

Figura 4.2: Desempenho dos algoritmos relativo aos 42 problemas de GE1 para os quais todos os algoritmos convergiram ao mesmo ponto estacionário.

Conjunto de problemas 1

A Figura 4.3 apresenta os perfis de desempenho dos algoritmos para os problemas do conjunto 1 de GE1. Nela, observamos que GDNT é o mais eficiente, pois resolve 50% dos problemas com o menor tempo de execução. Entretanto, em linhas gerais, GAMD apresentou o melhor desempenho. Analisando o perfil de GADP, concluímos que o processo de aceleração proposto por Pulay foi mais eficiente que o algoritmo LM sem aceleração.

A Tabela 4.2 apresenta os problemas para os quais os algoritmos obtiveram pontos estacionários diferentes. Observamos nesta tabela que, em geral, GAMD teve um desempenho superior aos demais algoritmos, atingindo o menor valor de função no menor tempo para os problemas 2 e 8. Para os problemas 5 e 6, dentre os algoritmos que usam a direção LM, GAMD foi aquele que encontrou o menor valor funcional. Entretanto, no primeiro problema, apesar de ter encontrado um minimizador local com menor tempo, GAMD obteve uma solução com valor funcional maior do que o apresentado pelos demais algoritmos.

Conjunto de problemas 2

Os perfis de desempenho para o conjunto 2 de GE1 são apresentados na Figura 4.4. Observamos que, para esse conjunto, GAMD obteve um desempenho muito superior aos

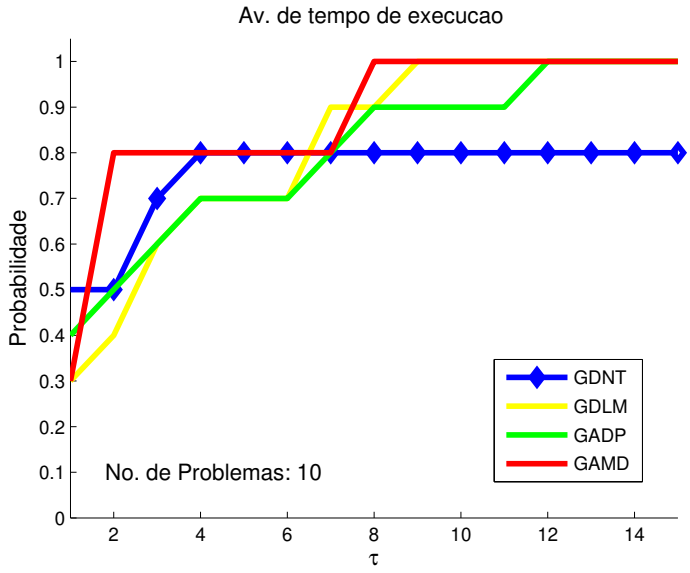

Figura 4.3: Perfil de desempenho para o conjunto de problemas 1 de GE1.

Prob.	GDNT		GDLM		GAMD		GADP	
	f	t(s)	f	t(s)	f	t(s)	f	t(s)
1	0.22842e-05	7	0.21643e-13	30	0.26628e-04	25	0.49849e-14	31
2	0.26205e-03	4	0.24256e-04	6	0.87433e-06	2	0.24256e-04	7
5	0.36027e-11	228	0.81210e-05	47	0.85334e-07	28	0.19290e-05	28
6	0.76891e-03	4309	0.10494e-02	206	0.89165e-03	253	0.92276e-03	192
8	0.27751e-03	216	0.20211e-05	1	0.20211e-05	1	0.20211e-05	1

Tabela 4.2: Valor funcional e tempo de execução para os problemas do conjunto 1 de GE1 nos quais os algoritmos obtiveram pontos estacionários diferentes.

demais algoritmos. Além disso, notamos que, embora tenha sido o mais eficiente, GDNT mostrou, mais uma vez, ser menos robusto para este tipo de problema.

Na Tabela 4.3, apresentamos os resultados obtidos nos casos em que os algoritmos chegaram a pontos estacionários diferentes. Para os problemas 8 e 9, nenhum dos dois algoritmos acelerados conseguiu executar uma iteração do processo de aceleração. No problema 8, os algoritmos que usam a direção LM obtiveram um valor funcional menor do que GDNT. Já para o problema 9, GDNT encontrou o minimizador global.

No problema 2, GAMD e GADP encontraram o mínimo global, enquanto GDNT só obteve um mínimo local. Para este problema, GAMD consumiu menos de 50% do tempo gasto por GDLM e GADP. Por outro lado, para o problema 6, GADP precisou de cerca de 50% do tempo gasto por GDLM e GAMD para obter o melhor mínimo local dentre

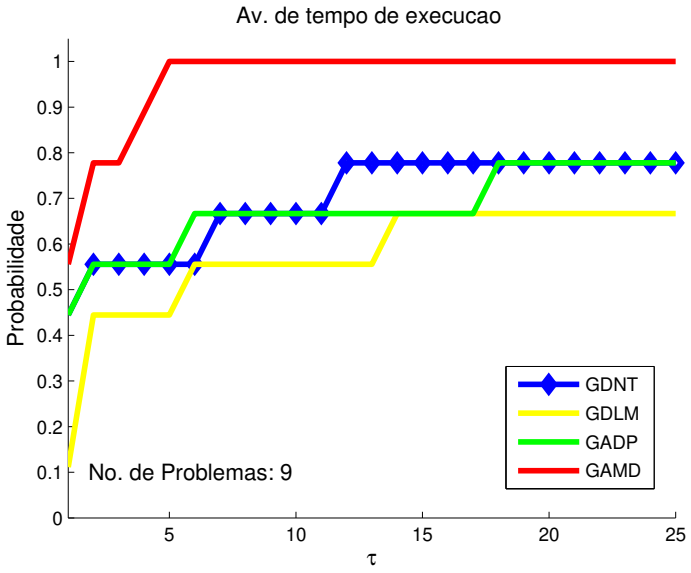

Figura 4.4: Perfil de desempenho para o conjunto de problemas 2 de GE1.

todos os algoritmos.

Prob.	GDNT		GDLM		GAMD		GADP	
	f	t(s)	f	t(s)	f	t(s)	f	t(s)
1	0.90178e-09	7	0.39447e-05	29	0.27717e-04	19	0.18100e-04	28
2	0.44163e-03	4	0.72989e-04	5	0.92323e-06	2	0.12137e-06	5
6	0.83773e-03	3297	0.71238e-03	190	0.71245e-03	215	0.58204e-03	101
8	0.34916e-03	115	0.19839e-05	2	0.19839e-05	2	0.19839e-05	2
9	0.35085e-06	1168	0.66291e-04	33	0.66291e-04	33	0.66291e-04	33
15	0.12262e+00	1316	0.90377e-01	41	0.91835e-01	48	0.88304e-01	47

Tabela 4.3: Valor funcional e tempo de execução para os problemas do conjunto 2 de GE1 nos quais os algoritmos obtiveram pontos estacionários diferentes.

Prob.	GDNT		GDLM		GAMD		GADP	
	f	t(s)	f	t(s)	f	t(s)	f	t(s)
13	0.19875e+02	4	0.15800e+02	11	0.15800e+02	5	0.15800e+02	11
14	0.19913e+02	39	0.15847e+02	11	0.15847e+02	2	0.15847e+02	11

Tabela 4.4: Valor funcional e tempo de execução para os problemas do conjunto 2 de GE1 para os quais pelo menos um dos algoritmos não convergiu.

Analisando a Tabela 4.4, reparamos que, ainda com relação ao segundo grupo de problemas de G1, apenas GDNT não atingiu a solução ótima para os problemas 13 e 14 (*função linear com posto deficiente*). Nestes problemas, GAMD foi muito superior aos demais algoritmos, obtendo um tempo computacional inferior à metade do tempo gasto pelos outros algoritmos com direção LM. Nestes casos, o processo de aceleração foi fundamental para os resultados obtidos.

Conjunto de problemas 3

Os 14 problemas do conjunto 3 de GE1 para os quais os algoritmos atingiram o mesmo ponto estacionário estão representados na Figura 4.5. Mais uma vez, GAMD foi superior aos demais algoritmos, seguido por GADP. Além disso, GDNT foi logo superado por GDLM, que acompanhou de perto o desempenho de GADP para $\tau \geq 2$.

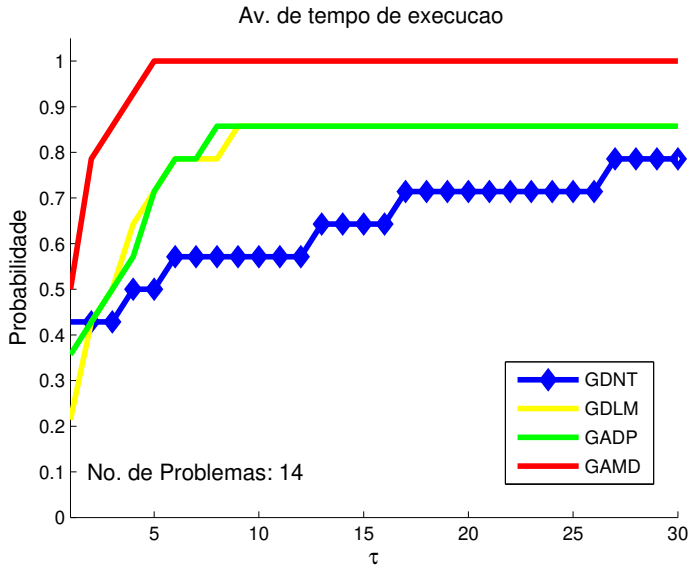

Figura 4.5: Perfil de desempenho para o conjunto de problemas 3 de GE1.

Prob.	GDNT		GDLM		GAMD		GADP	
	f	t(s)	f	t(s)	f	t(s)	f	t(s)
15	0.63253e+01	341	0.63315e+01	8	0.63275E+01	17	0.63277e+01	17

Tabela 4.5: Valor funcional e tempo de execução para os problemas do conjunto 3 de GE1 nos quais os algoritmos obtiveram pontos estacionários diferentes.

No terceiro grupo de experimentos, houve diferença entre os valores funcionais apenas nos últimos três problemas testados, como mostram as Tabelas 4.5 e 4.6. No problema 15,

Prob.	GDNT		GDLM		GAMD		GADP	
	f	t(s)	f	t(s)	f	t(s)	f	t(s)
13	0.35629e+02	15	0.29272e+02	17	0.27379e+02	6	0.29272e+02	17
14	0.35650e+02	2	0.29301e+02	17	0.27407e+02	2	0.29301e+02	17

Tabela 4.6: Valor funcional e tempo de execução para os problemas do conjunto 3 de GE1 para os quais pelo menos um dos algoritmos não convergiu.

os algoritmos convergiram para minimizadores locais distintos, sendo GDNT o algoritmo que apresentou o menor valor funcional, embora com um tempo de execução maior. Já nos problemas 13 e 14, somente GAMD convergiu.

Conjunto de problemas 4

Os resultados do último grupo de GE1 são apresentados na Figura 4.6 e nas Tabelas 4.7 e 4.8. A Figura 4.6 apenas reforça a superioridade de GAMD, já observada nos gráficos dos três outros grupos de GE1.

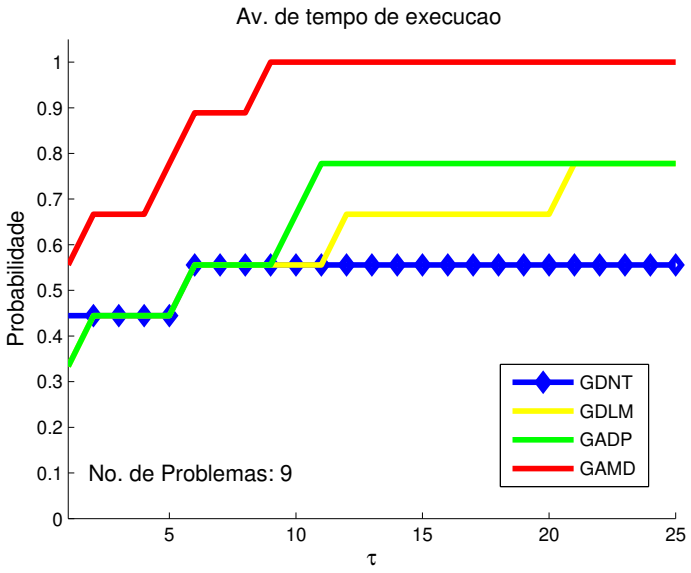

Figura 4.6: Perfil de desempenho para o conjunto de problemas 4 de GE1.

Analisando os problemas da Tabela 4.7, observamos que GAMD não usou a aceleração no problema 8, enquanto GADP não acelerou nos problemas 1, 2 e 8. Os algoritmos GDNT, GAMD e GADP encontraram a solução global do problema 5. Entretanto, GAMD e GADP o fizeram com um tempo significativamente melhor. O algoritmo GAMD foi o único a encontrar a solução ótima do problema 2. Dentre os algoritmos com

Prob.	GDNT		GDLM		GAMD		GADP	
	f	t(s)	f	t(s)	f	t(s)	f	t(s)
1	0.36739e-07	6	0.77265e-06	9	0.43236e-04	11	0.77265e-05	9
2	0.11678e-02	3	0.49567e-03	5	0.11209e-05	2	0.49567e-03	5
5	0.50819e-12	256	0.10880e-04	31	0.22882e-06	21	0.34103e-08	22
6	0.57723e-03	7499	0.10926e-02	258	0.92671e-03	181	0.11302e-02	210
8	0.34916e-03	118	0.19839e-05	2	0.19839e-05	2	0.19839e-05	2
10	0.17599e-05	2	0.21758e-04	4	0.17905e-05	1	0.21758e-04	5

Tabela 4.7: Valor funcional e tempo de execução para os problemas do conjunto 4 de GE1 nos quais os algoritmos obtiveram pontos estacionários diferentes.

direção LM, GAMD obteve, com o menor tempo, uma solução melhor para os problemas 6 e 10. Contudo, para esses problemas, GDNT obteve mínimos locais com valores funcionais menores, apesar de ter exigido um tempo de execução maior. O algoritmo GDNT também apresentou o melhor desempenho para o problema 1.

Na Tabela 4.8, constatamos que GDLM e GAMD atingiram a solução ótima nos problemas 13 e 14, e que GAMD convergiu em um tempo muito inferior ao consumido pelos demais algoritmos.

Prob.	GDNT		GDLM		GAMD		GADP	
	f	t(s)	f	t(s)	f	t(s)	f	t(s)
13	0.35107e+02	447	0.15800e+02	773	0.15800e+02	17	0.17725e+02	759
14	0.47248e+02	380	0.15847e+02	414	0.15847e+02	4	0.17847e+02	409

Tabela 4.8: Valor funcional e tempo de execução para os problemas do conjunto 4 de GE1 para os quais pelo menos um dos algoritmos não convergiu.

Comparando, de uma forma geral, o comportamento dos algoritmos com direção LM para os quatro conjuntos do grupo de experimentos 1, constatamos que os algoritmos acelerados foram os que mais reduziram o valor da função objetivo, como se comprova nas Tabelas 4.2 a 4.5. Observamos ainda que, em alguns problemas, o algoritmo não acelerado convergiu mais rapidamente para um minimizador, mas obteve um valor funcional maior do que os algoritmos acelerados.

4.3.2 Resultados para o grupo de experimentos 2

De modo análogo ao adotado para o grupo GE1, a Figura 4.7 mostra os perfis de desempenho dos algoritmos para os problemas no grupo de experimentos 2 nos quais a solução final coincidiu. Como isto ocorreu em poucos casos, o gráfico inclui todos

os problemas de $GE2$. Nas tabelas seguintes, apresentamos os resultados para aqueles problemas em que houve diferença entre as soluções.

Apesar de GDNT ter convergido em todos os casos, a Figura 4.7 nos permite concluir que este algoritmo exigiu um tempo de execução muito superior ao apresentado pelos algoritmos que usam a direção Levenberg-Marquardt. Para estes problemas, não se conseguiu nenhuma aproximação proveniente do processo de aceleração, motivo pelo qual há uma superposição dos gráficos que utilizam a direção LM. Ainda assim, a figura sugere que o tempo computacional adicional gasto na tentativa de acelerar não comprometeu o desempenho do algoritmos.

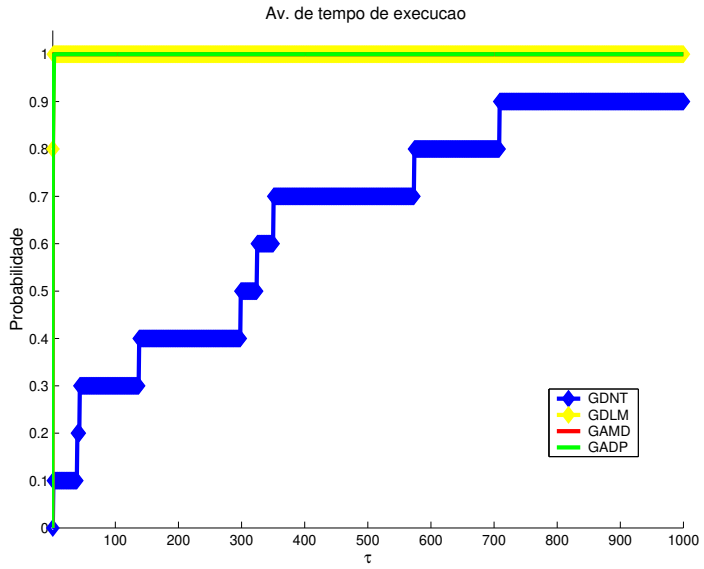

Figura 4.7: Desempenho dos algoritmos para os 10 problemas de GE2 nos quais todos os algoritmos convergiram para o mesmo ponto estacionário.

Observamos na Tabela 4.9 que os algoritmos com direção LM tiveram um desempenho melhor na maioria dos testes do problema das esferas. Além disso, o processo de aceleração tornou-os mais eficientes. Constatamos também que GDNT só obteve a melhor solução nos dois problemas iniciais. Contudo, nestes problemas, o tempo computacional exigido por GDNT foi maior ou igual ao obtido pelos demais algoritmos.

Entre os algoritmos acelerados, houve um certo equilíbrio, tendo cada algoritmo obtido a melhor solução no menor tempo para seis problemas. Entretanto, GADP consumiu um tempo excessivo para resolver o problema em que $n = 3$ e $p = 50$.

Na Tabela 4.10, apresentamos os casos em que os algoritmos encontraram mínimos locais distintos para o problema do aeroporto. Como se observa, os quatro programas encontraram soluções bastante próximas, com leve desvantagem para GDNT nos casos

		GDNT		GDLM		GAMD		GADP	
n	p	f	t(s)	f	t(s)	f	t(s)	f	t(s)
2	9	0.1420e+02	1	0.1465e+02	1	0.1465e+02	1	0.1465e+02	1
2	20	0.1263e+03	339	0.1282e+03	1	0.1282e+03	1	0.1263e+03	1
3	20	0.6759e+02	977	0.6567e+02	56	0.6567e+02	24	0.6567e+02	88
3	30	0.3216e-03	1	0.2262e-03	1	0.2266e-03	1	0.2266e-03	1
3	50	0.5703e+01	5795	0.5642e+01	310	0.5643e+01	127	0.5687e+01	86104
3	100	0.4045e-02	1025	0.2753e-02	33	0.2753e-02	34	0.2757e-02	34
3	150	0.1026e-01	69767	0.6632e-02	212	0.6632e-02	221	0.6632e-02	212
3	200	0.1928e-01	72982	0.1228e-01	887	0.1228e-01	884	0.1228e-01	884
4	25	0.1852e-03	2	0.1826e-03	1	0.1572e-03	1	0.1751e-03	1
4	50	0.8133e-03	62	0.5198e-03	37	0.5422e-03	24	0.5150e-03	26
5	25	0.1592e-03	2	0.1582e-03	1	0.1582e-03	1	0.1455e-03	1
5	40	0.4312e-03	90	0.3362e-03	2	0.3362e-03	3	0.4029e-03	2
5	150	0.7141e-02	39469	0.4404e-02	309	0.4404e-02	322	0.4404e-02	308
10	50	0.6371e-03	2085	0.5105e-03	129	0.4922e-03	89	0.5082e-03	69

Tabela 4.9: Valor funcional e tempo de execução para os problemas do conjunto E1 de GE2 nos quais os algoritmos obtiveram pontos estacionários diferentes.

em que $p = 25$ e $p = 50$. Com relação ao tempo de execução, o algoritmo GAMD foi superior para $p = 80$ e $p = 100$, enquanto GADP superou os demais no caso em que $p = 50$.

		GDNT		GDLM		GAMD		GADP	
n	p	f	t(s)	f	t(s)	f	t(s)	f	t(s)
2	25	0.1834e+05	66	0.1828e+05	1	0.1828e+05	1	0.1828e+05	1
2	50	0.7776e+05	71	0.7775e+05	28	0.7775e+06	20	0.7775e+05	18
2	80	0.2083e+06	1405	0.2083e+06	34	0.2084e+06	20	0.2083e+06	29
2	100	0.3487e+06	4204	0.3487e+06	120	0.3488e+06	47	0.3487e+06	127

Tabela 4.10: Valor funcional e tempo de execução para os problemas do conjunto E2 de GE2 nos quais os algoritmos obtiveram pontos estacionários diferentes.

Finalmente, apresentamos na Tabela 4.11 os testes do problema de empacotamento de cilíndros nos quais houve diferença na solução. No primeiro problema, o algoritmo GDNT obteve o melhor ponto entre os algoritmos, necessitando, entretanto, de um tempo de execução significativamente maior. No mesmo problema, GADP não conseguiu acelerar e GAMD obteve um mínimo local com maior valor funcional. Nos demais problemas,

GAMD foi bastante rápido e obteve mínimos locais com valores funcionais menores. Em nenhum do problemas apresentados na tabela 4.11, GADP acelerou.

			GDNT		GDLM		GAMD		GADP	
p	d1	d2	f	t(s)	f	t(s)	f	t(s)	f	t(s)
80	100	100	0.3718e+01	81349	0.4956e+01	13	0.5270e+01	14	0.4956e+01	14
80	25	3	0.7388e+01	299	0.7371e+01	83	0.7360e+01	11	0.7371e+01	87
100	25	3	0.8525e+01	22600	0.8354e+01	17	0.7971e+01	18	0.8554e+01	17

Tabela 4.11: Valor funcional e tempo de execução para os problemas do conjunto E3 de GE2 nos quais os algoritmos obtiveram pontos estacionários diferentes.

Capítulo 5

Conclusão

Neste trabalho, apresentamos um algoritmo para a minimização da soma de quadrados de funções com restrições de caixa. Este novo algoritmo pode ser classificado como um método de restrições ativas que combina:

- o uso do método Levenberg-Marquardt para minimização dentro da face ativa;
- o uso de uma técnica de aceleração do método LM que gera uma nova direção de busca a partir da combinação linear convexa das últimas p aproximações das funções f_i, $i = 1, \ldots, n$, que definem a função objetivo do problema;
- o uso do gradiente projetado espectral para o abandono da face, conforme proposto por Birgin e Martínez em [4].

Os experimentos numéricos apresentados sugerem que esta nova proposta é eficiente se comparada ao algoritmo GENCAN original e à técnica de aceleração proposta por Pulay, uma vez que, em geral, o novo algoritmo mostrou-se mais rápido e robusto que os demais, como comprovam os gráficos dos perfis de desemepenho.

Com efeito, analisando nosso primeiro grupo de experimentos, nos quais a multiplicidade dos pontos de mínimo é pequena, notamos uma considerável diminuição do tempo computacional gasto pelo algoritmo para encontrar a solução.

Para os experimentos que possuíam muitos minimizadores, o novo processo também foi capaz de encontrar um ponto estacionário em um tempo computacional reduzido. Além disso, na maioria dos casos, os valores funcionais encontrados eram comparáveis aos melhores valores obtidos.

Para a classe de experimentos com a qual trabalhamos, o algoritmo GENCAN original encontrou grandes dificuldades ao se deparar com problemas cuja matriz Hessiana é singular ou indefinida, e foi penalizado por não armazenar a Hessiana, que precisou ser

calculada parcialmente sempre que uma iteração do método dos gradientes conjugados truncado foi exigida.

Já o processo de aceleração proposto por Pulay, apesar de não ter-se destacado ao ser aplicado à resolução do primeiro grupo de experimentos, produziu resultados comparáveis àqueles apresentados pelo novo processo de aceleração para os problemas do segundo grupo.

Em linhas gerais, acreditamos que o novo processo de aceleração aqui proposto é uma ferramenta que aumenta a eficiência do algoritmo de otimização. Por esse motivo, sugerimos, como trabalho futuro, sua aplicação a outros métodos de minimização de somas de quadrados de funções.

Por fim, achamos que é necessário analisar o critério de escolha para a constante c_k que define a restrição do problema de aceleração, em vista de atribuir uma regra não empiríca que melhore o desempenho do algoritmo acelerado qualquer que seja o conjunto de testes.

Apêndice A

O método de Levenberg-Marquardt e regiões de confiança

Ao tentarmos resolver o subproblema quadrático

$$\min_{||s||\leq\Delta} q(s) = f + g^T s + \frac{1}{2} s^T B s \tag{A.1}$$

nos deparamos com a possibilidade de B não ser definida positiva ou o passo de Newton $s^N = -B^{-1}g$ ser muito grande (isto é, infactível). Nestes casos, o cálculo do passo corretor s^* que representa a solução do subproblema (A.1) exige a determinação de um número real $\lambda > 0$ que seja solução da equação não linear

$$||(B + \lambda I)^{-1} g|| = \Delta. \tag{A.2}$$

Existem muitas iterações aproximantes para λ (ver, por exemplo, [23], [43], [25], [32] e [35]). A iteração básica, devido à convergência rápida provocada pela concavidade da função utilizada, é baseada no método de Newton aplicado à função

$$\varphi(\lambda) = \frac{1}{\omega(\lambda)} - \frac{1}{\Delta},$$

onde $\omega(\lambda) = || - (B + \lambda I)^{-1} g||$. Se é dada uma aproximação $\overline{\lambda}_k$ que torna a matriz $B + \overline{\lambda}_k I$ definida positiva, mas fornece uma passo inaceitável, então calculamos uma nova aproximação, $\overline{\lambda}_{k+1}$, da solução de (A.2), pela fórmula de recorrência

$$\overline{\lambda}_{k+1} = \overline{\lambda}_k - \frac{\varphi(\overline{\lambda}_k)}{\varphi'(\overline{\lambda}_k)}. \tag{A.3}$$

O cálculo do valor da função φ e de sua derivada exige o cálculo de ω e de sua derivada. Isto não é um problema sério, uma vez que, nesta etapa, já temos a decomposição de Cholesky $R^T R$ da matriz $B + \overline{\lambda}_k I$, com R triangular superior.

Vejamos como reescrever $\overline{\lambda}_{k+1}$ simplificando a razão $\frac{\varphi(\overline{\lambda}_k)}{\varphi^{'}(\overline{\lambda}_k)}$. Sabemos que

$$\varphi^{'}(\lambda) \;=\; -\frac{\omega'(\lambda)}{\omega^2(\lambda)}.$$

Logo,

$$\overline{\lambda}_{k+1} \;=\; \overline{\lambda}_k - \left(\frac{\omega(\overline{\lambda}_k)-\Delta}{\omega^{'}(\overline{\lambda}_k)}\right)\frac{\omega(\overline{\lambda}_k)}{\Delta}. \tag{A.4}$$

Por outro lado, seja $s = s(\lambda)$ dado por

$$(B+\lambda I)s(\lambda) = -g. \tag{A.5}$$

Então,

$$\omega(\lambda) = ||s(\lambda)|| = \sqrt{s^T(\lambda)s(\lambda)}.$$

Pela Regra da Cadeia,

$$\begin{aligned}
\omega^{'}(\lambda) &= \frac{1}{2\sqrt{s^T(\lambda)s(\lambda)}}\left[\sum_{i=1}^{n} s_i^2(\lambda)\right]^{'} \\
&= \frac{2\sum_{i=1}^{n} s_i(\lambda)s_i^{'}(\lambda)}{2\sqrt{s^T(\lambda)s(\lambda)}} \\
&= \frac{s^{'}(\lambda)^T s(\lambda)}{||s(\lambda)||}.
\end{aligned}$$

Derivando implicitamente a equação (A.5), temos

$$Bs^{'}(\lambda) + \lambda s^{'}(\lambda) + s(\lambda) = 0 \Leftrightarrow (B+\lambda)s^{'}(\lambda) = -s(\lambda).$$

Segue que

$$s^{'}(\lambda) = -(B+\lambda I)^{-1}s(\lambda) = -(R^TR)^{-1}s(\lambda).$$

Portanto,

$$\begin{aligned}
\omega^{'}(\lambda) &= \frac{s^{'}(\lambda)^T s(\lambda)}{||s(\lambda)||} = \frac{-(R^{-1}R^{-T}s(\lambda))^T s(\lambda)}{||s(\lambda)||} \\
&= -\frac{s(\lambda)^T R^{-1}R^{-T}s(\lambda)}{||s(\lambda)||} = -\frac{(R^{-T}s(\lambda))^T(R^{-T}s(\lambda))}{||s(\lambda)||}.
\end{aligned}$$

Ou seja,

$$\omega^{'}(\lambda) = -\frac{(R^{-T}s)^T(R^{-T}s)}{||s||}. \tag{A.6}$$

Definindo $s_k \equiv s(\overline{\lambda}_k) = -(B + \overline{\lambda}_k I)^{-1} g$ e $r_k = R^{-T} s_k$ e substituindo a expressão (A.6) em (A.4), com $\lambda = \overline{\lambda}_k$, podemos reescrever a iteração de Newton como

$$\overline{\lambda}_{k+1} = \overline{\lambda}_k + \left(\frac{||s_k|| - \Delta}{\Delta} \right) \frac{||s_k||^2}{||r_k||^2}.$$

O algoritmo seguinte atualiza $\overline{\lambda}$ pelo método de Newton aplicado a $(A.3)$, com o objetivo de garantir a igualdade $||s(\overline{\lambda})|| = \Delta$.

Algoritmo A.1 *Seja $\overline{\lambda} \geq 0$ um número real dado tal que a matriz $B + \overline{\lambda} I$ é positiva definida.*

P1 *Encontre a fatoração de Cholesky da matriz $B + \overline{\lambda} I = R^T R$, onde R é uma matriz triangular superior.*

P2 *Resolva o sistema $R^T R s = -g$*

P3 *Se $||s|| \simeq \Delta$ pare! Caso contrário, resolva o sistema $R^T r = v$*

P4 *Calcule*

$$\overline{\lambda} = \overline{\lambda} + \frac{||s||^2}{||r||^2} \left(\frac{||s|| - \Delta}{\Delta} \right).$$

P5 *Retorne ao passo $P1$.*

Embora não tenhamos mencionado neste algoritmo, é necessário definir, a cada iteração, um intervalo que contenha o parâmetro λ^* ótimo. Este mesmo intervalo é usado para detectar o "caso difícil", mencionado adiante. Leitores interessados devem consultar [23], [43] ou [32].

Vejamos, agora, como encontrar o parâmetro Levenberg-Marquardt aplicando o método de Newton à função φ.

Considere a função

$$\varphi(\lambda) = \frac{1}{||s(\lambda)||} - \frac{1}{\Delta}, \tag{A.7}$$

onde o vetor $s(\lambda)$ é definido por

$$(B + \lambda I) s(\lambda) = -g,$$

com $B \in \mathbb{R}^{n \times n}$ simétrica, $g \in \mathbb{R}^n$ e $\lambda \in \mathbb{R}$ um parâmetro que torna a matriz $B_k + \lambda I$ definida positiva.

O Lema abaixo, apresentado sem demonstração, será usado para mostrar que φ definida em (A.7) é uma função côncava.

Lema A.1 *Seja $\varphi : \mathcal{S} \subset \mathbb{R}^n \to \mathbb{R}$ uma função real, tal que $\mathcal{S}$ é um conjunto convexo, com interior não vazio. Então, φ é côncava se, e somente se, $\nabla^2\varphi(x) \leq 0$ para todo $x \in \mathcal{S}$.*

Seja $\mathcal{S} = (-\xi_1, +\infty) \subset \mathbb{R}$. Claramente $\mathcal{S}$ é convexo e não vazio. Seja $\varphi : \mathcal{S} \subset \mathbb{R} \to \mathbb{R}$ definida por (A.7). Pelo Lema A.1, resta verificar que

$$\nabla^2\varphi(x) \leq 0.$$

Com efeito, sejam $\xi_1 \leq \xi_2 \leq \ldots \xi_n$ os autovalores da matriz B ordenados em ordem crescente. Considere $B = U^T\Sigma U$ a decomposição espectral da matriz B, onde U é uma matriz ortogonal e

$$\Sigma = diag(\xi_1, \xi_2, \ldots, \xi_n).$$

Então,

$$\begin{aligned} s(\lambda) &= -U^T(\Sigma + \lambda I)^{-1}Ug = -U^T(\Sigma + \lambda I)^{-1}v \\ &= -U^T(\frac{v_1}{\xi_1 + \lambda}, \ldots, \frac{v_n}{\xi_n + \lambda})^T. \end{aligned}$$

Como U preserva a norma Euclidiana, temos

$$||s(\lambda)||^2 = \sum_{i=1}^{n}(\frac{v_i}{\xi_i + \lambda})^2.$$

Denotando por $\overline{\omega}(\lambda) = ||s(\lambda)||^2$, temos que

$$\varphi(\lambda) = \frac{1}{\overline{\omega}^{\frac{1}{2}}(\lambda)} - \frac{1}{\Delta}.$$

Usando a *Regra da Cadeia*, obtemos

$$\varphi^{'}(\lambda) = -\frac{1}{2}\overline{\omega}^{-\frac{3}{2}}(\lambda)\overline{\omega}^{'}(\lambda)$$

e

$$\varphi^{''}(\lambda) = \frac{3}{4}\overline{\omega}^{-\frac{5}{2}}(\lambda)\overline{\omega}^{'}(\lambda) - \frac{1}{2}\overline{\omega}^{-\frac{3}{2}}(\lambda)\overline{\omega}^{''}(\lambda). \tag{A.8}$$

Mas,

$$\begin{aligned} \overline{\omega}^{'}(\lambda) &= 2\sum_{i=1}^{n}\left(\frac{v_i}{\xi_i + \lambda}\right)\left(\frac{v_i}{\xi_i + \lambda}\right)^{'} \\ &= -2\sum_{i=1}^{n}\frac{v_i^2}{(\xi_i + \lambda)^3}. \end{aligned}$$

Escrevendo

$$\overline{\omega}^{'}(\lambda) = -2\sum_{i=1}^{n} v_i^2(\xi_i + \lambda)^{-3}, \tag{A.9}$$

obtemos

$$\overline{\omega}^{''}(\lambda) = +6\sum_{i=1}^{n} v_i^2(\xi_i + \lambda)^{-4}. \tag{A.10}$$

Substituindo (A.9) e (A.10) em (A.8), obtemos

$$\begin{aligned}
\varphi^{''}(\lambda) &= \frac{3}{4\overline{\omega}^2(\lambda)\sqrt{\overline{\omega}(\lambda)}}\left[-2\sum_{i=1}^{n}\frac{v_i^2}{(\xi_i+\lambda)^3}\right] - \frac{1}{2\overline{\omega}(\lambda)\sqrt{\overline{\omega}(\lambda)}}\left[6\sum_{i=1}^{n}\frac{v_i^2}{(\xi_i+\lambda)^4}\right] \\
&= -\left\{\frac{3}{2\overline{\omega}^2(\lambda)\sqrt{\overline{\omega}(\lambda)}}\sum_{i=1}^{n}\frac{v_i^2}{(\xi_i+\lambda)^3} + \frac{3}{\overline{\omega}(\lambda)\sqrt{\overline{\omega}(\lambda)}}\sum_{i=1}^{n}\frac{v_i^2}{(\xi_i+\lambda)^4}\right\}.
\end{aligned}$$

Como $\lambda > -\xi_1$, temos $\lambda + \xi_i > 0$ para cada $i = 1, 2, \ldots, n$. Assim, cada parcela da soma

$$\sum_{i=1}^{n}\frac{v_i^2}{(\xi_i+\lambda)^3}$$

é positiva. Uma vez que a segunda parcela em $\varphi^{''}$ é sempre positiva, segue que $\varphi^{''}(\lambda) < 0$ qualquer que seja $\lambda \in (-\xi_1, +\infty)$.■

No que segue, apresentamos um resultado (ver Nocedal e Yuan [35], ou Nocedal e Wright [34]) que fornece estimativas inferior e superior para o parâmetro Levenberg-Marquardt λ que satisfaz

$$B + \lambda I \succ 0 \tag{A.11}$$

e

$$||s(\lambda)|| = \Delta, \tag{A.12}$$

onde $s(\lambda) = -(B+\lambda I)^{-1}g$ e usamos o símbolo $\succ$ para indicar que uma matriz é definida positiva.

Lema A.2 *O parâmetro λ que satisafaz (A.11) e (A.12) satisfaz à seguinte condição intervalar:*

$$-\xi_1 < \lambda \leq \frac{||g||}{\Delta} - \sigma,$$

com $\sigma \leq \xi_1$, sendo ξ_1 o menor autovalor da matriz B.

Prova: Considere a decomposição espectral $B = U^T\Sigma U$, sendo U uma matriz ortogonal e $\Sigma = diag(\xi_1, \xi_2, \ldots, \xi_n)$ a matriz diagonal formada pelos autovalores $\xi_1 \leq \xi_2 \leq \cdots \leq \xi_n$ da matriz B.

Como Σ é diagonal, o menor autovalor da matriz $\Sigma + \lambda I$ é $\sigma_1(\Sigma + \lambda I) = \xi_1 + \lambda$. Para que $B + \lambda I$ seja definida positiva, devemos ter

$$\xi_1 + \lambda > 0. \tag{A.13}$$

Por outro lado,

$$\begin{aligned} (B + \lambda I)s(\lambda) = -g \quad &\Rightarrow \\ U^T(\Sigma + \lambda I)Us(\lambda) = -g \quad &\Rightarrow \\ (\Sigma + \lambda I)\overline{s}(\lambda) = -\overline{g}, & \end{aligned}$$

com $\overline{s} = Us$ e $\overline{g} = Ug$. Uma vez que $\sigma_1(A) \leq ||A||$ temos

$$\sigma_1(\Sigma + \lambda I)||\overline{s}|| \leq ||(\Sigma + \lambda I)\overline{s}(\lambda)|| = ||\overline{g}||.$$

Logo

$$(\xi_1 + \lambda)||\overline{s}|| \leq ||\overline{g}||. \tag{A.14}$$

Por outro lado, como U é ortogonal, temos

$$||\overline{s}|| = ||Us|| = ||s|| \text{ e } ||\overline{g}|| = ||Ug|| = ||g||.$$

Destas considerações e de (A.14), segue que

$$(\xi_1 + \lambda)||s|| \leq ||g||,$$

ou ainda que

$$\lambda \leq \frac{||g||}{||s||} - \xi_1. \tag{A.15}$$

De (A.13) e (A.15), com $||s|| = \Delta$, obtemos o resultado desejado.■

A.1 Analisando o *caso difícil*

Considere o problema de minimizar a quadrática

$$\min_{||s|| \leq \Delta} q(s) = g^T s + \frac{1}{2} s^T B s. \tag{A.16}$$

O lema abaixo, demonstrado por Sorensen [43] e analisado por Coleman [10], associa o Método Levenberg-Marquardt aos métodos do tipo *regiões de confiança,* a partir do subproblema quadrático (A.16).

Lema A.3 *O vetor s resolve (A.16) com* $||\cdot|| = ||\cdot||_2$ *se, e somente se, existe* $\lambda \geq 0$ *tal que:*

(i) $B + \lambda I$ *é definida positiva.*

(ii) $(B + \lambda I)s = -g$.

(iii) $||s|| \leq \Delta$.

(iv) $\lambda(||s|| - \Delta) = 0$.

Considere que $B = U^T \Sigma U$, onde as colunas de U são autovetores ortogonais de B e

$$\Sigma = diag(\xi_1, \xi_2, \ldots, \xi_n), \quad \xi_1 \leq \xi_2 \leq \cdots \leq \xi_n,$$

é a matriz diagonal definida pelos autovalores de B. Naturalmente,

$$B + \lambda I = U^T(\Sigma + \lambda I)U.$$

Logo, (ii) é equivalente a

$$(\Sigma + \lambda I)\overline{s} = \alpha, \quad \text{onde} \quad \alpha = -Ug \quad \text{e} \quad \overline{s} = Us. \tag{A.17}$$

Vamos supor que a multiplicidade algébrica de ξ_1 seja c. Assim,

$$U = [u_{1_1}, u_{1_2}, \ldots, u_{1_c}, u_{c+1}, \ldots, u_n],$$

com u_{1_k} e u_k representando a k-ésima coluna de U. Os vetores $\{u_{1_1}, u_{1_2}, \ldots, u_{1_c}\}$ são os autovetores associados ao autovalor ξ_1, os quais formam uma base para o núcleo de $B - \xi_1 I$ e são tais que

$$u_j^T u_{1_k} = 0, \quad \forall k = 1, 2, \ldots, c \quad \text{e} \quad \forall j = c+1, \ldots, n.$$

Podemos reescrever (A.17) como

$$\begin{pmatrix} \xi_1 + \lambda & \ldots & 0 & 0 & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \xi_1 + \lambda & 0 & 0 & 0 \\ 0 & \ldots & 0 & \xi_{c+1} + \lambda & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \xi_n + \lambda \end{pmatrix} \begin{pmatrix} \overline{s}_1 \\ \vdots \\ \overline{s}_c \\ \overline{s}_{c+1} \\ \vdots \\ \overline{s}_n \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_c \\ \alpha_{c+1} \\ \vdots \\ \alpha_n \end{pmatrix}. \tag{A.18}$$

De (A.18), tiramos algumas conclusões:

1. Se $\xi_i > 0,\ \forall i = 1, 2, \ldots, n$, então, para todo $\lambda \geq 0$, a matriz $B + \lambda I$ é definida positiva e o problema (A.16) tem solução. Para constatar isso, basta perceber que, se $\lambda \to +\infty$, então $||s(\lambda)|| \to 0$, e se $\lambda \to -\xi_1$, então $||s(\lambda)|| \to +\infty$. Ou seja, $Im(||s(\lambda)||) = (0, +\infty)$. Assim, se $||s(0)|| \leq \Delta$, então $\lambda = 0$ é solução. Caso contrário, pela continuidade da função $||s(\lambda)||$, existe $\lambda^* \in (-\xi_1, +\infty)$ tal que $||s(\lambda^*)|| = \Delta$. Ou seja, as condições (i)–(iv) sempre se verificam.

2. Vamos supor agora que exista algum autovalor negativo. Assim, $\xi_1 < 0$. Ou seja, B não é definida positiva. Suponha que $\exists\, i \in \{1, 2, \ldots, c\} : \alpha_i \neq 0$. Note que $\alpha_i = g^T u_i$. Assim, g tem alguma componente não nula no núcleo de $B - \xi_1 I$. Suponha, sem perda de generalidade, que $i = c$. Assim, $\alpha_i = 0$, para cada $i = 1, 2, \ldots, c-1$. Logo, para (A.17) ou (A.18) ter solução, basta exigir que $\overline{s}_i = 0$ para $i = 1, 2, \ldots, c-1$ e definir

$$s(\lambda) = \sum_{i=c}^{n} (\frac{\alpha_i}{\lambda + \xi_i}) u_i, \quad \text{com} \quad \lambda > -\xi_1.$$

   Com efeito, note que $||s(\lambda)|| \to 0$ quando $\lambda \to +\infty$ e que $||s(\lambda)|| \to +\infty$ quando λ tende a $-\xi_1$ pela direita, gararantindo a existência de uma solução para algum $\lambda > -\xi_1$.

3. Agora, suponha que $\alpha_i = 0, \quad \forall i = 1, 2, \ldots, c$. Logo, $g \notin N(B + \xi_1 I)$. De (A.17), devemos ter

   3.a $(\xi_1 + \lambda)\overline{s}_i = 0, \quad i = 1, 2, \ldots, c;$

   3.b $(\xi_1 + \lambda)\overline{s}_i = \alpha_i, \quad i = c+1, 2, \ldots, n.$

   Defina

$$s(\lambda) = \sum_{i=c+1}^{n} (\frac{\alpha_i}{\lambda + \xi_i}) u_i, \quad \text{com} \quad \lambda \geq -\xi_1.$$

   Se $||s(-\xi_1)|| \geq \Delta$, existe solução. De fato, define-se $\lambda = -\xi_1$ ou escolhe-se $\lambda > -\xi_1$ de modo que $||s(\lambda)|| = \Delta$. Isto é possível uma vez que $||s(\lambda)|| \to 0$ quando $\lambda \to +\infty$. Caso contrário, não existe solução para (A.17), qualquer que seja $\lambda > -\xi_1$. Este caso é frequentemente chamado de *caso difícil*. Entretanto, podemos encontrar uma solução para (A.16) tomando $\lambda = -\xi_1$ e usando as variáveis livres de $\overline{s}_i$ em $(3.a)$. Fazemos isto através da seguinte definição:

$$s(-\xi_1) = \sum_{i=c+1}^{n} (\frac{\alpha_i}{\xi_i - \xi_1}) u_i + \sum_{i=1}^{c} \beta_i u_{1_c}, \tag{A.19}$$

   com $\sum_{i=1}^{c} \beta_i^2 = \Delta^2 - \sum_{i=c+1}^{n} (\frac{\alpha_i}{\xi_i - \xi_1})^2$.

Infelizmente, a distinção entre os casos 2 e 3 não é clara do ponto de vista numérico. Ainda assim, podemos obter uma solução para (A.16) através da equação (A.19), adotando $\alpha_i = 0$ quando α_i é muito pequeno.

Coleman [10] afirma que esta estratégia funciona bem na prática, uma vez que a solução de (A.16) com α_i pequeno está próxima de (A.19) com $\alpha_i = 0$. Para ilustrar isso, consideremos um exemplo em que a multiplicidade geométrica do autovalor λ_1 é igual a um, ou seja, $c = 1$, de modo que $\xi_1 < \xi_2$.

Primeiramente, suponhamos que $\alpha_1 = 0$. Neste caso, a solução de (A.16) é

$$s = \sum_{i=2}^{n} (\frac{\alpha_i}{\xi_i - \xi_1}) u_i + \beta u_1,$$

onde $\beta^2 = \Delta^2 - \sum_{i=2}^{n} (\frac{\alpha_i}{\xi_i - \xi_1})^2$ e consideramos que $\Delta^2 > \sum_{i=2}^{n} (\frac{\alpha_i}{\xi_i - \xi_1})^2$, caso contrário, s é solução com $\beta = 0$. Definamos $s_1 = \sum_{i=2}^{n} (\frac{\alpha_i}{\xi_i - \xi_1}) u_i$. Neste caso, a solução de (A.16) pode ser escrita como $s = s_1 + \beta u_1$.

Agora, consideremos $\alpha_1 = \varepsilon$, um número pequeno. Podemos, então, escrever a solução de (A.16) como

$$s(\lambda) = \sum_{i=2}^{n} (\frac{\alpha_i}{\lambda + \xi_i}) u_i + \frac{\alpha_1}{\lambda + \xi_1} u_1.$$

Mas, se $\alpha_1 \to 0$, então $\lambda \to -\xi_1$ (para manter $||s(\lambda)|| = \Delta$). E se $\lambda \to -\xi_1$, então

$$\sum_{i=2}^{n} (\frac{\alpha_i}{\lambda + \xi_i}) u_i \to s_1,$$

o que implica que

$$(\frac{\alpha_1}{\lambda + \xi_i})^2 \to \beta^2 \quad \text{sempre que} \quad \alpha_1 \to 0.$$

Portanto, a solução de (A.16) com $\alpha_1 = \varepsilon$ está bem próxima da solução (A.16) com $\alpha_1 = 0$.

Em geral, se $c > 1$, a solução de (A.16), com alguma componente α_i, $i \leq c$, próxima de zero, estará próxima da solução de (A.16) com esta componente igual a zero. No caso em que várias componentes α_i, $i \leq c$, são nulas o problema (A.16) não possui solução única.

Para mais informações sobre o subproblema com região de confiança, o leitor deve consultar [7], [11], [23], [32], [41] e [43].

Apêndice B

Teorema da projeção

Neste apêndice, vamos relembrar o Teorema da Projeção que foi usado na demonstração de alguns resultados apresentados nos capítulos desta tese. Os resultados apresentados, bem como as demonstrações, fazem parte da literatura (ver, por exemplo, Bertsekas [2]) e são apresentados aqui para facilitar o entendimento dos contéudos abordados e de algumas demonstrações apresentadas. Começaremos apresentando algumas definições.

Definição B.1 *Seja Ω um subconjunto de $\mathbb{R}^n$.*

1. *Uma função $f : \Omega \to \mathbb{R}^n$ é contínua em um ponto $x \in \Omega$ se $\lim_{y \to x} f(y) = f(x)$. A função f é dita ser contínua sobre Ω se é contínua em cada elemento x pertencente a Ω.*

2. *Uma função real $f : \Omega \to \mathbb{R}$ é chamada* semicontínua superiormente *(respectivamente,* semicontínua inferiormente*) em um elemento $x \in \Omega$ se $f(x) \geq \limsup_{k \to \infty} f(x_k)$ (respectivamente, $f(x) \leq \limsup_{k \to \infty} f(x_k)$) para cada seqüência x_k em Ω convergindo a x.*

3. *Uma função real $f : \Omega \to \mathbb{R}$ é chamada* coerciva *se*

$$\lim_{k \to \infty} f(x_k) = \infty$$

para cada sequência $\{x_k\}$ de elementos de Ω tais que $||x_k|| \to \infty$ para alguma norma $|| \cdot ||$.

Abaixo, apresentamos o teorema de *Weierstrass*, utilizado na demonstração do teorema da projeção.

Teorema B.1 *(**Weierstrass**) Seja Ω um subconjunto não vazio de $\mathbb{R}^n$ e seja $f : \Omega \to \mathbb{R}$ uma função real semicontínua inferior em todos os pontos de Ω. Suponha que uma das seguintes condições vale:*

1. *Ω é compacto.*
2. *Ω é fechado e f é coerciva.*
3. *Existe um escalar γ tal que o conjunto de nível*

$$\{x \in \Omega \ : \ f(x) \leq \gamma\}$$

é não vazio e compacto.

Então existe um vetor $x \in \Omega$ tal que $f(x) = \inf_{z \in \Omega} f(z)$.

Prova: Suponha inicialmente que Ω é compacto. Seja $\{z_k\}$ uma seqüência de elementos de Ω tais que

$$\lim_{k \to \infty} f(z_k) = \inf_{z \in \Omega} f(z).$$

Como Ω é limitado, esta seqüência tem um ponto limite. Seja x este ponto, ou seja,

$$\lim_{k \to \infty} z_k = x.$$

Uma vez que Ω é fechado, $x \in \Omega$. Por definição, f é semicontínua inferiormente, o que implica que

$$f(x) \leq \lim_{k \to \infty} f(z_k) = \inf_{z \in \Omega} f(z).$$

Mas, $\inf_{z \in \Omega} f(z) \leq f(x)$. Portanto, temos $f(x) = \inf_{z \in \Omega} f(z)$.

Suponha, agora, o caso 2. Considere a seqüência $\{z_k\}$ como na parte anterior. Como f é coerciva, a seqüência $\{z_k\}$ não pode ser ilimitada, e a prova segue de modo inteiramente análogo ao que foi apresentado no caso anterior.

Considere o caso 3. Se γ é igual a $\inf_{z \in \Omega} f(z)$, o conjunto dos pontos de mínimo de f sobre Ω é $\{x \in \Omega \ : \ f(x) \leq \gamma\}$. Como, por hipótese, este conjunto é não vazio, a afirmação está provada.

Se $\gamma > \inf_{z \in \Omega} f(z)$, considere uma seqüência $\{z_k\}$ de elementos em Ω tais que

$$\lim_{k \to \infty} f(z_k) = \inf_{z \in \Omega} f(z).$$

Então, para todo k suficientemente grande, z_k deve pertencer ao conjunto $\{x \in \Omega : f(x) \leq \gamma\}$. Uma vez que este conjunto é compacto, o conjunto $\{z_k\}$ é limitado e a prova procede como nos casos anteriores.■

Teorema B.2 *(**Teorema da Projeção**) Seja Ω um conjunto convexo e fechado, e seja $||\cdot||$ a norma Euclidiana.*

1. *Para cada $x \in \mathbb{R}^n$, existe um único vetor $z \in \Omega$ que minimiza $||z-x||$ sobre todos os elementos $z \in \Omega$. Este vetor é chamado* projeção de x sobre Ω*, e é denotado por*
$$z \equiv P_\Omega(x) = arg\min_{z\in\Omega} ||z-x||.$$

2. *Dado algum vetor $x \in \mathbb{R}^n$, um vetor $z \in \Omega$ é a projeção de x sobre Ω se, e somente se,*
$$(y-z)^T(x-z) \leq 0, \quad \forall y \in \Omega. \tag{B.1}$$

3. *Quaisquer que sejam x e y tem-se*
$$||P_\Omega(x) - P_\Omega(y)||^2 \leq (P_\Omega(x) - P_\Omega(y))^T(x-y). \tag{B.2}$$

Prova: Afirmação 1. Para verificar a existência da projeção, fixe x e considere y algum elemento de Ω. Resolver o problema $\min_{z\in\Omega} ||z-x||$ é equivalente a minimizar a mesma função sobre todos os elementos z em Ω tais que $||z-x|| \leq ||y-x||$, que é um conjunto compacto. Além disso, a função g definida por

$$g(z) = ||z-x||^2$$

é contínua. O vetor minimizante segue do Teorema de Weierstrass.

Para verificar a unicidade da projeção, note que o quadrado da norma Euclidiana é uma função convexa estrita. Logo, o seu mínimo é atingido em um único ponto.

Afirmação 2. Para todo y e z em Ω, tem-se

$$||y-x||^2 = ||y-z||^2 + ||z-x||^2 - 2(y-z)^T(x-z) \geq ||z-x||^2 - 2(y-z)^T(x-z).$$

Logo, se z é tal que $(y-z)^T(x-z) \leq 0, \quad \forall z \in \Omega$, então

$$||y-x||^2 \geq ||z-x||^2, \quad \forall y \in \Omega,$$

o que implica que $z = P_\Omega(x)$.

Reciprocamente, seja $z = P_\Omega(x)$, considere qualquer $y \in \Omega$, e para qualquer $\alpha > 0$, defina

$$y_\alpha = \alpha y + (1-\alpha)z.$$

Neste caso, tem-se

$$\begin{aligned} ||x-y_\alpha||^2 &= ||(1-\alpha)(x-z) + \alpha(x-y)||^2 \\ &= (1-\alpha)^2||z-x||^2 + \alpha^2||x-y||^2 + 2(1-\alpha)\alpha(x-z)^T(x-y). \end{aligned}$$

Defina $\xi(\alpha) = ||x - y_\alpha||^2$. Então

$$\begin{aligned} \frac{\partial \xi}{\partial \alpha}|_{\alpha=0} &= -2||x-z||^2 + 2(x-z)^T(x-y) \\ &= -2(y-z)^T(x-z). \end{aligned}$$

Portanto, se $(y-z)^T(x-z) > 0$ para algum $y \in \Omega$, então

$$\frac{\partial \xi}{\partial \alpha}|_{\alpha=0} < 0$$

e, para $\alpha > 0$ suficientemente pequeno, obtém-se

$$||x - y_\alpha|| \leq ||x - z||.$$

Isto contradiz o fato que z é a projeção de x sobre Ω e mostra que $(y-z)^T(x-z) \leq 0$ para todo $y \in \Omega$.

Afirmação 3. Utilizando o item 2 do Teorema da Projeção, tem-se

$$(P_\Omega(x) - x)^T(P_\Omega(x) - P_\Omega(y)) \leq 0 \tag{B.3}$$

e

$$(P_\Omega(y) - y)^T(P_\Omega(y) - P_\Omega(x)) \leq 0. \tag{B.4}$$

Somando (B.3) e (B.4), e usando a identidade

$$||w - z||^2 = w^T w + z^T z - 2w^T z, \quad \forall z, w \in \mathbb{R}^n$$

obtém-se o resultado desejado■

O item 2, do teorema acima, diz que a direção dada pela diferença do vetor x que se deseja projetar da sua projeção, $x - P_\Omega(x)$, forma um ângulo não agudo (ver Figura B.1) em relação ao vetor obtido pela diferença de qualquer y em Ω pela projeção de x, $y - P_\Omega(x), y \in \Omega$. Isto equivale a dizer que o produto interno entre estes é não positivo.

Um consequência direta do *Teorema da Projeção* é o fato de que toda projeção sobre um conjunto convexo é uma aplicação não expansiva. Esta observação é comprovada pelo corolário seguinte.

Corolário B.3 *Seja Ω um subconjunto convexo não vazio de $\mathbb{R}^n$. Seja $P_\Omega(x)$ a projeção de um vetor $x \in \mathbb{R}^n$ sobre Ω. Então,*

$$||P_\Omega(x) - P_\Omega(y)|| \leq ||x - y||, \quad \forall x, y \in \mathbb{R}^n. \tag{B.5}$$

Prova: Se $P_\Omega(x) = P_\Omega(y)$, (B.5) é trivialmente satisfeita. Se $P_\Omega(x) \neq P_\Omega(y)$, aplicamos a desigualgade de Schwartz ao lado direito da desigualde dada pelo item 3 do Teorema da Projeção, obtendo o resultado desejado.■

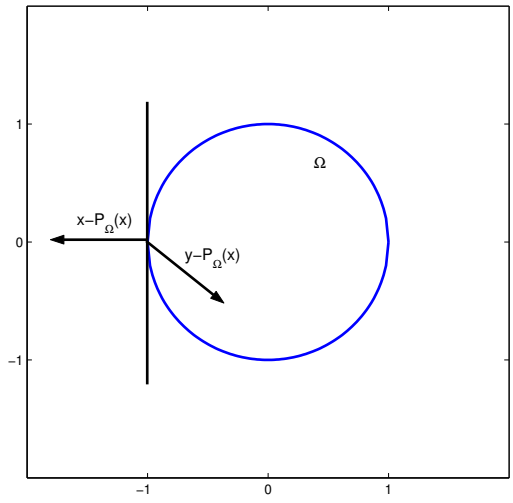

Figura B.1: Ilustração da condição satisfeita pela projeção $P_\Omega(x)$. Para cada elemento $y \in \Omega$, os vetores $x - P_\Omega(x)$ e $y - P_\Omega(x)$ formam um ângulo maior ou igual a 90^o. Equivalentemente, o produto interno $(x - P_\Omega(x))^T(y - P_\Omega(x))$ é não positivo.

Corolário B.4 *Seja Ω um subconjunto convexo não vazio de $\mathbb{R}^n$. Seja $P_\Omega(x)$ a projeção de um vetor $x \in \mathbb{R}^n$ sobre Ω. Se o vetor nulo $\mathbf{0} \in \Omega$, então*

$$||P_\Omega(x)|| \leq ||x||, \quad \forall x \in \mathbb{R}^n.$$

Prova: Basta tomar $y = \mathbf{0}$ em (B.5).■

Finalizamos esta seção apresentando um resultado utilizado para mostrar que a aplicação

$$h(s) = \frac{||P_\Omega(x - sz) - x|| - x}{s}, \quad s > 0,$$

onde $P_\Omega(\cdot)$ é a projeção sobre um conjunto convexo Ω e x e z são vetores fixados, é não decrescente.

Proposição B.1 *Seja $v \in \mathbb{R}^n$ e $x \in \Omega \subset \mathbb{R}^n$, com Ω convexo. As seguintes afirmações são equivalentes:*

1. $P_\Omega(x + v) = x$
2. $v^T x \geq v^T w, \quad \forall w \in \Omega$
3. $P_\Omega(x + tv) = x, \quad$ *para todo* $t > 0$.

Prova: Provemos inicialmente que a afirmação 1 implica na afirmação 2. Pelo item 2 do Teorema da Projeção, temos

$$(x + v - P_\Omega(x + v))^T(w - P_\Omega(x + v)) \leq 0, \quad \forall w \in \Omega. \tag{B.6}$$

Substituindo $P_\Omega(x + v) = x$ e subtraindo os termos iguais em (B.6), obtemos

$$v^T(w - x) \leq 0, \quad \forall w \in \Omega,$$

que é equivalente ao resultado esperado.

Vejamos porque a afirmação 2 implica 3. Suponha que $v^T w \leq v^T x, \quad \forall w \in \Omega$, e que $x \in \Omega$. Logo, qualquer que seja $t > 0, \ t \in \mathbb{R}$, a desigualdade

$$(tv)^T(w - x) \leq 0, \quad \forall w \in \Omega, \tag{B.7}$$

continua verdadeira. Dado um $t > 0$ qualquer, definamos $z = x + tv$. Então, por (B.7), temos

$$(z - x)^T(w - x) \leq 0, \quad \forall w \in \Omega. \tag{B.8}$$

Pelo item 2 do teorema da projeção, (B.8) ocorre se, e somente se, $x = P_\Omega(z)$. Como $t > 0$ foi tomado arbitrariamente, o resultado segue.

A demonstração de que a afirmação 3 implica na afirmação 1 é trivial, bastando tomar, em particular, $t = 1$■

Apêndice C

Perfil de desempenho

Nosso último apêndice é dedicado ao perfil de desempenho introduzido por Dolan e Moré [16], que é usado para comparar o desempenho de vários programas quando aplicados a um conjunto de problemas testes.

No que segue, se C é um conjunto qualquer, $card(C)$ representa a cardinalidade do conjunto C e, se $x \in \mathbb{R}$, o símbolo $\lfloor x \rfloor$ representa o maior inteiro menor ou igual a x.

Seja $\mathcal{S}$ o conjunto dos n_s programas que queremos comparar e seja $\mathcal{P}$ o conjunto dos n_p problemas testes que pretendemos resolver. Consideremos como medida de comparação o tempo computacional, embora a idéia aqui desonvolvida possa ser estendida a outras medidas, como o número de cálculo de funções.

Para cada $s \in \mathcal{S}$ e para cada $p \in \mathcal{P}$, definimos $t_{p,s}$ como o tempo consumido pelo programa s para resolver o problema p.

Também definimos a *razão de desempenho*, $r_{p,s}$, do programa s com relação ao problema p, como a razão entre $t_{p,s}$ e o menor tempo obtido por todos os programas para resolver o mesmo problema, ou seja,

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} \ : \ s \in \mathcal{S}\}}.$$

Observamos que $r_{p,s} \geq 1$, quaisquer que sejam $p \in \mathcal{P}$ e $s \in \mathcal{S}$.

Definimos

$$\rho_s(\tau) = \frac{\text{card}\left(\{p \in \mathcal{P} \ : \ r_{p,s} \leq \tau\}\right)}{n_p}$$

como a fração do conjunto de todos os problemas testes resolvida pelo programa s com uma razão de desempenho menor ou igual a τ. A função $\rho_s : \mathbb{R} \to [0, 1]$ é constante por partes e não decrescente. Além disso, o valor de $\rho_s(1)$ é a fração do conjunto de problemas em que o programa s teve um desempenho absoluto melhor que os outros.

Para cada valor de τ, o melhor dentre os n_s programas analisados será aquele que possuir o valor mais alto de $\rho_s(\tau)$. Notamos, também, que $1 - \rho_s(\tau)$ é a fração de

problemas que o programa s não consegue resolver dentro de um fator τ do tempo gasto pelo melhor algoritmo, incluindo os problemas para os quais o programa em questão falhou.

Vamos supor que r_M seja um parâmetro real, escolhido de modo que

i) $r_M \geq r_{p,s},\ \forall p \in \mathcal{P},\ \forall s \in \mathcal{S}$;

ii) $r_{p,s} = r_M$ para algum problema p e para algum programa s se, e somente se, o programa s não resolver o problema p.

A escolha de r_M não afeta o gráfico do perfil de desempenho (ver Dolan-Moré [16]). Além disso, $\rho_s(r_M) = 1$ e

$$\rho_s^* = \lim_{t \to r_M^-} \rho_s(t)$$

é a fração do conjunto de problemas resolvida pelo programa s, independentemente do tempo gasto. Logo, se estamos interessados apenas nos programas com probabilidade alta de sucessos, devemos comparar os valores de ρ_s^* e escolher o programa que possuir o maior valor.

Ao traçarmos o gráfico do perfil de desempenho de um algoritmo, utilizamos um intervalo $[1,\ \tau_M]$ para a variável τ. A seleção do limitante τ_M deve ser feita com cuidado, já que a escolha de um valor pequeno demais pode não capturar o comportamento total do programa com relação ao conjunto de problemas testes. Além disso, como, em muitos casos, o intervalo $[1,\ \tau_M]$ é grande, pode ser preferível traçar o gráfico do perfil de desempenho em escala logarítmica, considerando

$$\rho_s(\tau) = \frac{card\left(\{p \in \mathcal{P}\ :\ log_2(r_{p,s}) \leq \tau\}\right)}{n_p}.$$

Neste caso, ampliamos a região abrangida pelo gráfico, embora, por outro lado, percamos um pouco a intuição sobre o que está acontecendo.

Em resumo, o perfil de desempenho é a função de distribuição (acumulada) que indica quais programas têm maior probabilidade de serem os mais robustos (aqueles que atingiram o valor 1) e quais são (ou possuem maior probabilidade de serem) mais eficientes. Se o número de problemas testes for consideravelmente grande, esta estratégia elimina o risco de um pequeno grupo de problemas dominar os resultados e é estável para variações pequenas nos dados de entrada.

Bibliografia

[1] Barzilai, J. & Borwein, J. M. Two point step size gradient methods. *IMA J. Numerical Analysis* **8**, 1988, pp. 141–148.

[2] Bertsekas, D. P. *Nonlinear programming*. 2. ed. Belmont, Athena Scientific, 1999.

[3] Birgin, E. G. & Martínez, J. M. A box constrained optimization algorithm with negative curvature directions and spectral projected gradients. *Computing*, Sup. **15**, 2001, pp. 49–60.

[4] Birgin, E. G. & Martínez, J. M. Large-scale active-set box-constrained optimization method with spectral projected gradients. *Comput. Optim. and Aplics.* **23**, 2002, pp. 101–125.

[5] Birgin, E. G.; Martínez, J. M.; Raydan, M. Nonmonotone spectral projected gradient methods on convex sets. *SIAM J. Optimization* **10**, 2000, pp. 1196–1211.

[6] Björck, A. *Numerical methods for least squares problems*. Philadelphia, SIAM, 1996.

[7] Byrd, R. H.; Schnabel, R. B.; Shultz, G. Approximate solution of the trust region problem by minimization over two-dimensional subspaces. *Math. Programming* **40**, 1988, pp. 247–263.

[8] Chan, L. & Szeto, C. Training recurrent network with block-diagonal approximated Levenberg-Marquardt algorithm. *IEEE Computer Society* **3**, 1999, pp. 1521–1526.

[9] Chen, T.; Han, D.; Au, F. T. K.; Tham, L. G. Acceleration of Levenberg-Marquardt training of neural networks with variable decay rate. Em *Proceedings of the International Joint Conference on Neural Networks*, 2003, pp 1873–1878.

[10] Coleman, T. F. *The solution of large-scale optimization problems (using MATLAB)*. Acessado em http://www.tc.cornell.edu/c̃oleman. Setembro, 2004.

[11] Coleman, T. F. & Hempel, C. *Computing a trust region step for a penalty function.* SIAM J. Sci. Stat. Comput. **11**, 1990, pp. 180–201.

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

[13] Csaszar, P. & Pulay, P. Geometry Optimization by DIIS, *J. Molec. Structure, THEOCHEM* **114**, 1984, pp. 31–34.

[14] Davies, M. & Whitting, I. J. A modified form of Levenberg's correction. Em *Numerical Methods for Nonlinear Optimization*, F. A. Lootsma (ed.) London, Academic Press, 1972, pp. 191–201.

[15] Dennis, J. E. & Schnabel, R. B. *Numerical methods for unconstrained optimization and nonlinear methods.* Englewood Cliffs, Prentice Hall, 1983.

[16] Dolan, E. D. & Moré, J. J. Benchmarking optimization software with performance profiles. *Math. Programming* (Ser. A) **91**, 1981, pp. 201–213.

[17] Facchinei, F.; Júdice, J.; Soares, J. Generating box-constrained optimization problems. *ACM Trans. Math. Software* **23**, 1997, pp. 443–447.

[18] Fan, J. & Yuan, Y. On the convergence of the a new Levenberg-Marquardt method. Technical Report, AMSS, Chinese Academy of Sciences, Beijing, China, 2001.

[19] Farkas, Ö. & Schlegel, H. B. An improved algorithm for geometry optimization using direct inversion of the iterative subspace GDIIS. *Phys. Chem. Physics* **4**, 2002, pp. 11–15.

[20] Farkas, Ö. & Schlegel, H. B. Geometry optimization methods for modeling large molecules. *J. Molec. Structure, THEOCHEM* **666–667**, 2003, pp. 31–39.

[21] Fletcher, R. *Pratical methods of optimization*, 2.ed. Chichester, Wiley, 1987.

[22] Friedlander, A.; Martínez, J. M.; Santos, S. A. A new trust region algorithm for bound constrained minimization. *Appl. Math. Optim.* **30**, 1994, pp. 235–266.

[23] Gay, D. M. Computing optimal locally constrained steps. *SIAM J. Sci. Stat. Comput.* **2**, 1981, pp. 186–197.

[24] Goldfield, S. M.; Quandt, R. E.; Trotter, H. F. Maximization by quadratic hill-climbing. *Econometrica* **34**, 1966, pp. 541–551.

[25] Hebden, M. D. An Algorithm for minimization using exact second derivatives. Report TP515, A.E.R.E., Harwell, England, 1973.

[26] Levenberg, K. A method for the solution of certain problems in least squares. *Quart. Ap. Math.* **2**, 1944, pp. 164–168.

[27] Madsen, K.; Nielsen, H. B.; Tingleff, O. Methods for non-linear least-squares problems, 2.ed. Lecture note, Informatics and Mathematical Modelling, Technical University of Denmark, Lyngby, Dinamarca, 2004.

[28] Marquardt, D. An algorithm for least squares estimation of nonlinear parameters. *J. Soc. Indust. Ap. Math.* **11**, 1963, pp. 431–441.

[29] Martínez, J. M. BOX-QUACAN and the implementation of augmented Lagrangian algorithms for minimization with inequality. *Comput. Appl. Mathematics* **19**, 2000, pp. 31–56.

[30] Moré, J. J. The Levenberg-Marquardt algorithm: implementation and theory. Em *Proceedings of the 1977 Dundee conference on numerical analysis*, Lecture Notes in Mathematics 630. G. A. Watson. (ed.) Berlin, Springer, 1978, pp. 105–116.

[31] Moré, J. J.; Garbow, B. S.; Hillstrom, K. E. Testing unconstrained optimization software. *ACM Trans. Math. Software* **7**, 1981, pp. 17–41.

[32] Moré, J. J. & Sorensen, D. C. Computing a trust region step. *SIAM J. Sci. Stat. Comput.* **4**, 1983, pp. 553–572.

[33] Nielsen, H. B. Damping parameter in Marquardt's method. Report IMM-REP-1999-05, Technical University of Denmark, Lyngby, Dinamarca, 2005.

[34] Nocedal, J. & Wright, S. J. *Numerical optimization*. Berlin, Springer, 1999.

[35] Nocedal, J. & Yuan, Y. Combining trust region and line search tecniques. Em *Advances in Nonlinear Programming*, Y. Yuan (ed.) Dordrecht, Kluwer, 1998, pp. 153–175.

[36] Osborne, M. R. Nonlinear least squares–the Levenberg algorithm revisited. *J. Austral. Math. Soc.* (Ser. B) **19**, 1976, pp. 343–357.

[37] Pulay, P. Convergence acceleration of iterative sequences: the case of SCF iteration. *Chem. Phys. Letters* **73**, 1980, pp. 392–398.

[38] Raydan, M. On the Barzilai and Borwein choice of steplength for the gradient method. *IMA J. Numer. Analysis* **13**, 1993, pp. 321–326.

[39] Rojas, Rojas, M.; Santos, S. A.; Sorensen, D. C. A new matrix-free algorithm for the large-scale trust-region subproblem, *SIAM J. Optimization* **11**, 2000, pp. 611–646.

[40] Sakamoto, H.; Matsumoto, K.; Kuwahara, A.; Hayami, Y. Acceleration and stabilization techniques for the Levenberg-Marquardt method. *IEICE Trans. Fundamentals* **E88-A**, 2005, pp. 1971–1978.

[41] Schubert, L. Modification of a quasi-Newton method for nonlinear equations with a sparse Jacobian. *Math. Comp.* **24**, 1970, pp. 27–30.

[42] Sherrill, C. D. Some comments on accelerating convergence of iterative sequences using direct inversion of iterative subspace (DIIS). Acessado em http://vergil.chemistry.gatech.edu/notes/diis. Setembro, 2004.

[43] Sorensen, D. C. Newton's method with a model trust region modification. *SIAM J. Numer. Anal.* **19**, 1982, pp. 409–426.

[44] Wilamowski, B. M.; Chen, Y.; Malinowski, A. Efficient algorithm for training neural networks with one hidden layer. Em *Proceedings of the International Joint Conference on Neural Networks*, 1999, pp 1725–1728.

[45] Yamashita, N. & Fukushima, M. On the rate of convergence of the Levenberg-Marquardt method. Technical Report 2000-008, Kyoto University, Kyoto, Japão, 2000.

[46] Yuan, Y. *Nonlinear optimization: trust region algorithms.* Acessado em http://citeseer.ist.psu.edu/232660.html. Abril, 2005.

[47] Zhou, G. & Si, J. Advanced neural-network training algorithm with reduced complexity based on Jacobian deficiency. *IEEE Trans. on Neural Net.* **9**, 1998, pp. 448–453.