



UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E COMPUTAÇÃO
CIENTÍFICA

Sobre minimização de quadráticas em caixas

Bruna Lammoglia

Dissertação de Mestrado orientada pela
Profa. Dra. Maria Aparecida Diniz Ehrhardt

Sobre minimização de quadráticas em caixas

Este exemplar corresponde à redação final da dissertação de mestrado devidamente corrigida e defendida por Bruna Lammoglia e aprovada pela comissão julgadora.

20 de dezembro de 2007.



Profa. Dra. Maria Aparecida Diniz Ehrhardt

Banca Examinadora:

1. Profa. Dra. Maria Aparecida Diniz Ehrhardt
2. Profa. Dra. Véra Lucia Rocha Lopes
3. Profa. Dra. Luziane Ferreira de Mendonça

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

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP
Bibliotecária: Maria Júlia Milani Rodrigues**

Lammoglia, Bruna

L188s Sobre minimização de quadráticas em caixas / Bruna Lammoglia --
Campinas, [S.P. :s.n.], 2007.

Orientador : Maria Aparecida Diniz Ehrhardt

Dissertação (mestrado) - Universidade Estadual de Campinas,
Instituto de Matemática, Estatística e Computação Científica.

1. Otimização. 2. Algoritmos. 3. Programação não-linear. I. Diniz-
Ehrhardt, Maria Aparecida. II. Universidade Estadual de Campinas.
Instituto de Matemática, Estatística e Computação Científica. III. Título.

Título em inglês: About box constrained quadratic minimization.

Palavras-chave em inglês (Keywords): 1. Optimization. 2. Algorithms. 3. Non-linear programming.

Área de concentração: Otimização

Titulação: Mestre em Matemática Aplicada

Banca examinadora: Profa. Dra. Maria Aparecida Diniz Ehrhardt (IMECC-UNICAMP)
Profa. Dra. Luziane Ferreira de Mendonça (UFRJ)
Profa. Dra. Véra Lucia da Rocha Lopes (IMECC-UNICAMP)

Data da defesa: 20-12-2007

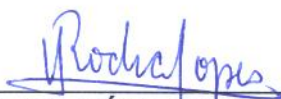
Programa de pós-graduação: Mestrado em Matemática Aplicada

Dissertação de Mestrado defendida em 20 de dezembro de 2007 e aprovada

Pela Banca Examinadora composta pelos Profs. Drs.



Prof. (a). Dr (a). MARIA APARECIDA DINIZ EHRHARDT



Prof. (a). Dr (a). VÉRA LÚCIA DA ROCHA LOPES



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

Resumo

Neste trabalho o objetivo principal foi a minimização de quadráticas em caixas. Dissertamos sobre os métodos de máxima descida e dos gradientes conjugados, bem como sobre um método mais recente denominado gradiente espectral. O GENCAN, um algoritmo que minimiza funções em caixas, foi estudado em detalhe, particularmente avaliando sua aplicação para quadráticas. O objetivo foi analisar o desempenho do GENCAN, comparado com algoritmos anteriores, como o LANCELOT e o QUACAN. Foram executados experimentos numéricos a fim de avaliar o desempenho das versões de GENCAN sem e com pré-condicionamento. Concluiu-se que pré-condicionar o método dos gradientes conjugados neste caso tornou o GENCAN mais robusto. No entanto, o pré-condicionador usado neste software mostrou-se computacionalmente caro. Em relação à comparação do GENCAN, LANCELOT e QUACAN, podemos afirmar que o GENCAN mostrou-se competitivo.

Abstract

The focus of this work was the minimization of quadratic functions with box constraints. We were mainly concerned about the steepest descent and conjugated gradient methods, besides a more recent approach called spectral gradient method. The GENCAN, an algorithm that minimizes functions on a box, was studied in details particularly evaluating this algorithm applied to quadratics. The objective was to analyze the efficiency of GENCAN, comparing it to classical algorithms, such as LANCELOT and QUACAN. We executed numerical experiments in order to investigate the efficiency of GENCAN version with and without preconditioning. Evaluating the results we concluded that preconditioning the conjugated gradient method makes the GENCAN work considerably better; despite the fact that the preconditioner used here turned the computational process more expensive. Comparing GENCAN, LANCELOT, and QUACAN we can state that GENCAN is competitive.

*Ao meu pai, Álvaro, que, tenho certeza, adoraria
compartilhar essa conquista.*

Agradecimentos

Gostaria de agradecer a todos que me ajudaram durante o tempo em que permaneci no mestrado. Alguns estiveram ao meu lado por muito tempo, outros por pouco, mas todos foram importantes para que eu chegasse onde cheguei.

Primeiramente gostaria de agradecer à minha orientadora Cheti por sua paciência, inteligência, bondade, e principalmente por ser a professora maravilhosa que é.

Gostaria de agradecer minha família por estar sempre do meu lado, ajudando no que fosse preciso e dando-me forças para continuar. Obrigada Marta, Talita, Rafaella, Norma, Maria Clara, Tuca, Lourdes, Carlito, Toninha, Júnior, Natália, Bárbara.

Agradeço aos meus amigos que me ajudaram nos estudos e são amigos que levarei para toda vida. Obrigada Marina, Maristela, Luciana, Jair, Rafael, Lucilene, Mateus, Viviana, Rodrigo e especialmente Igor, por estar comigo do começo ao fim.

Agradeço meus amigos de trabalho Regina, Elias e Adilson.

Agradeço à Patrícia e ao Marinho por me darem bons conselhos, ao Eraldo e ao Lucas pela hospedagem, ao Eduardo por aguentar meu mau humor nos últimos tempos, à Ester, por sempre “estar lá” e ao Fernando da informática, por ajudar sempre que precisei.

Agradeço aos coordenadores Laércio e Aurélio, por serem pessoas acessíveis e compreensivas.

Agradeço às professoras Luziane Ferreira de Mendonça, Véra Lucia Rocha Lopes, Valeria Abrão de Podestá e Mariângela Amêndola pelas correções e dicas.

Finalmente, agradeço à Secretaria da Educação do Estado de São Paulo, por financiar esta pesquisa.

Conteúdo

| | |
|--|-----------|
| Lista de Tabelas | ix |
| Introdução | 1 |
| 1 Minimização Irrestrita | 3 |
| 1.1 Estratégias de Globalização | 6 |
| 1.2 Métodos Clássicos para Minimização Irrestrita | 9 |
| 1.2.1 Método de Máxima Descida (MMD) ou do Gradiente | 9 |
| 1.2.2 Método de Newton | 11 |
| 1.2.3 Métodos quase-Newton | 12 |
| 2 Minimização de Quadráticas Sem Restrições | 15 |
| 2.1 Métodos Iterativos | 17 |
| 2.1.1 Método de Máxima Descida (MMD) para Funções Quadráticas | 17 |
| 2.1.2 Método dos Gradientes Conjugados | 19 |
| 2.1.3 Método do Gradiente Espectral | 25 |
| 3 Minimização de Quadráticas em Caixas | 29 |
| 3.1 Algoritmo Geral para Quadráticas em Caixas | 32 |
| 3.2 GENCAN | 35 |
| 4 Pré-condicionadores | 43 |
| 4.1 Pré-condicionando GENCAN | 45 |
| 4.1.1 QNCGNA | 45 |
| 4.1.2 Outro Pré-condicionador para GENCAN | 46 |

| | |
|---|-----------|
| 5 Experimentos Numéricos | 47 |
| 5.1 Análise do Desempenho do GENCAN | 47 |
| 5.2 LANCELOT, QUACAN e GENCAN | 49 |
| Conclusão | 54 |
| A Notações | 56 |
| B Definições | 58 |
| C Tabelas | 60 |
| Bibliografia | 63 |

Lista de Tabelas

| | | |
|-----|--|----|
| 5.1 | Comparação GENCAN com e sem pré-condicionador | 51 |
| 5.2 | Gráficos comparativos GENCAN com e sem pré-condicionador | 51 |
| 5.3 | LANCELOT sem pré-condicionador | 52 |
| 5.4 | LANCELOT com pré-condicionador | 52 |
| 5.5 | QUACAN | 52 |
| 5.6 | Comparação entre os algoritmos sem pré-condicionamento | 53 |
| 5.7 | Comparação entre os algoritmos com pré-condicionamento | 53 |
| 5.8 | Gráficos comparativos GENCAN, LANCELOT e QUACAN | 53 |
| C.1 | GENCAN sem pré-condicionadores | 61 |
| C.2 | GENCAN com QNCGNA | 62 |

Introdução

A técnica de otimização tem um grande (e crescente) papel tanto em diversas áreas de pesquisa científica como, por exemplo, nas distintas áreas das engenharias e economias, bem como em aplicações na indústria sendo, por isso, interessante que entendamos o desenvolvimento dos algoritmos associados a essa técnica, para que saibamos aplicá-los em situações-problemas reais.

Decidimos trabalhar com funções quadráticas pelo grande número de problemas que recaem nesse formato, como por exemplo, minimização dos quadrados dos erros em uma aproximação de parâmetros lineares para uma curva da qual conhecemos parcialmente o comportamento, ou ainda subproblemas para auxiliar a resolução de problemas mais complexos; enfim, a minimização de quadráticas é frequentemente usada em algoritmos de otimização, por ter uma estrutura relativamente simples.

Na década de 90, os algoritmos denominados LANCELOT e QUACAN [11, 16], eram certamente os mais aplicados para minimização de quadráticas em caixas. Ambos baseiam-se em estratégias de região de confiança, com o princípio de restrições ativas, usando o método de gradientes conjugados nas faces da caixa e uma direção baseada no gradiente projetado para deixar a face.

Em 2001, foi proposto o GENCAN [6, 1, 2], um método de restrições ativas com estratégias de busca linear, que ainda usa gradientes conjugados dentro das faces, mas o método do gradiente espectral para sair da face.

Assim, nossos principais objetivos neste trabalho foram:

- estudar com detalhes os conceitos e os métodos envolvidos na minimização de quadráticas em caixas,
- analisar o desempenho do GENCAN, comparando-o com LANCELOT e QUACAN. Nesse sentido, pretendemos contrapor busca linear e região de confiança;

projeção do gradiente espectral e do gradiente. Tudo isso no contexto de minimização de quadráticas em caixas.

No capítulo 1, tratamos primeiramente de conceitos fundamentais na minimização irrestrita de funções, para caracterizarmos os tipos de soluções que podemos encontrar e as condições para que elas existam, além de mostrarmos algumas características que um algoritmo de minimização irrestrita pode ter. No mesmo capítulo descrevemos métodos clássicos para minimizar funções, que também podem ser aplicados a quadráticas: Método do gradiente, Newton e quase-Newton.

No capítulo 2, iniciamos o estudo sobre minimização de quadráticas sem restrições. Estudamos métodos clássicos essencialmente no caso de minimização de quadráticas: os métodos do gradiente e dos gradientes conjugados e também um método mais recente: o método do gradiente espectral.

No capítulo 3, estudamos minimização de quadráticas em caixas, objetivo principal de nossa dissertação. Mostramos um algoritmo geral que utiliza o princípio das restrições ativas para que todas as faces da caixa onde a quadrática se encontra possam ser exploradas e encontremos o mínimo na caixa ou uma direção em que a quadrática sempre decresça. Em seguida descrevemos os princípios do algoritmo GENCAN que minimiza funções em caixas.

No capítulo 4 tratamos de pré-condicionamento de algoritmos, para melhorar o desempenho do método dos gradientes conjugados e consequentemente do GENCAN.

No capítulo 5 mostramos os experimentos numéricos que realizamos com os problemas do CUTE [8] e os comparamos com resultados obtidos com os mesmos problemas, mas com os algoritmos LANCELOT e QUACAN. No capítulo 6, fazemos algumas considerações e a conclusão.

Ao final, temos o apêndice A com as notações utilizadas no texto; o apêndice B com algumas definições que não cabiam no corpo do trabalho, mas que são necessárias para sua compreensão e o apêndice C, com as tabelas completas dos problemas que analisamos.

Capítulo 1

Minimização Irrestrita

Este capítulo tem como objetivo introduzir os conceitos fundamentais que são usados em qualquer problema de otimização sem restrições. Na próxima seção trataremos especificamente de funções quadráticas.

Num problema geral de minimização, teoricamente procuramos um minimizador global. Porém, essa exigência é muito forte, além de ser difícil de reconhecer se um ponto é um minimizador global. Na prática, analisaremos propriedades mais fracas, mas que são satisfeitas pelo minimizador global, que são as *condições necessárias de otimalidade*.

Estamos então tratando do problema:

$$\text{Minimizar } f(x), \tag{1.1}$$

onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$.

Sendo assim, buscamos um vetor $x^* \in \mathbb{R}^n$ que soluciona o problema (1.1). Tal vetor é denominado *minimizador* de $f(x)$. Dois tipos de solução serão consideradas, o minimizador local e o global, de acordo com as definições que seguem:

- Um ponto $x^* \in \mathbb{R}^n$ é um minimizador local de f se existe $\varepsilon > 0$ tal que $f(x) \geq f(x^*)$ para todo $x \in \mathbb{R}^n$ que satisfaz $\|x - x^*\| < \varepsilon$.
- Um ponto $x^* \in \mathbb{R}^n$ é um minimizador global de f se $f(x) \geq f(x^*), \forall x \in \mathbb{R}^n$.

Temos como consequência direta das definições acima que um minimizador global é também um minimizador local.

Vamos agora expor as condições de otimalidade necessárias de primeira e segunda ordem, e as condições suficientes de segunda ordem, para que x^* seja um minimizador local de f [18].

Proposição 1. *Condições necessárias de primeira ordem*

Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $f \in C^1$. Se x^* é um minimizador local de f em \mathbb{R}^n , então $\nabla f(x^*) = 0$.

Demonstração. Considere $d \in \mathbb{R}^n$ arbitrário e seja a função $\varphi : \mathbb{R} \rightarrow \mathbb{R}$ definida por:

$$\varphi(\lambda) = f(x^* + \lambda d).$$

Se x^* é um minimizador local de f , então $\lambda = 0$ é um minimizador local de φ ($\varphi(0) = f(x^*)$). Assim, supondo conhecido o resultado para uma variável de que se x^* é um minimizador local de uma função $g : \mathbb{R} \rightarrow \mathbb{R}$, $g \in C^1$, então $g'(x^*) = 0$, temos que $\varphi'(0) = 0$.

Usando a regra da cadeia obtemos $\varphi'(\lambda) = \nabla^T f(x^* + \lambda d)d$.

Substituindo para $\lambda = 0$, resulta que $\varphi'(0) = 0 = \nabla^T f(x^*)d$.

Como tomamos $d \in \mathbb{R}^n$ arbitrário, a igualdade acima significa que $\nabla f(x^*)$ é ortogonal a todos os vetores do espaço, portanto $\nabla f(x^*) = 0$. \square

Dizemos que um ponto onde o gradiente da função se anula é estacionário de uma função.

Proposição 2. *Condições necessárias de segunda ordem*

Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $f \in C^2$. Se x^* é um minimizador local de f em \mathbb{R}^n , então:

- $\nabla f(x^*) = 0$;
- $\nabla^2 f(x^*)$ é semidefinida positiva.

Demonstração. A primeira parte da proposição já está demonstrada.

Para demonstrar a segunda parte, vamos supor conhecido o resultado para uma variável:

Seja $g : \mathbb{R} \rightarrow \mathbb{R}, g \in C^2$. Se x^* é um minimizador local de g em \mathbb{R} , então $g'(x^*) = 0$ e $g''(x^*) \geq 0$.

Considerando $\varphi(\lambda)$ como na proposição 1, esse resultado nos diz que $\varphi''(0) \geq 0$. Usando a regra da cadeia temos $\varphi''(\lambda) = d^T \nabla^2 f(x^* + \lambda d)d$, logo $\varphi''(0) = d^T \nabla^2 f(x^*)d \geq 0$.

Como $d \in \mathbb{R}$ é arbitrário temos que $\nabla^2 f(x^*)$ é semidefinida positiva. \square

Proposição 3. Condições suficientes de segunda ordem

Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}, f \in C^2$. Se $x^* \in \mathbb{R}^n$, $\nabla f(x^*) = 0$ e $\nabla^2 f(x^*) > 0$, então x^* é um minimizador local estrito de f em \mathbb{R}^n .

Demonstração. Seja $B = \{h \in \mathbb{R}^n / \|h\| = 1\}$. Consideremos a função $\Gamma : B \rightarrow \mathbb{R}$ dada por

$$\Gamma(h) = h^T \nabla^2 f(x^*)h.$$

Γ é uma função contínua e B é um conjunto fechado e limitado, portanto Γ atinge um valor máximo e um valor mínimo em B . Chamemos a o valor mínimo, então

$$\Gamma(h) \geq a > 0 \text{ para todo } h \in B.$$

Agora, consideremos $d \in \mathbb{R}^n$ arbitrário, não nulo. Como $d/\|d\| \in B$, temos que

$$d^T \nabla^2 f(x^*)d \geq a\|d\|^2.$$

Desenvolvendo f em série de Taylor em torno de x^* , temos

$$f(x^* + d) - f(x^*) = \nabla^T f(x^*)d + \frac{1}{2}d^T \nabla^2 f(x^*)d + o(\|d\|^2). \quad (1.2)$$

Desde que, por hipótese, $\nabla f(x^*) = 0$, (1.2) implica que

$$f(x^* + d) - f(x^*) \geq \frac{a}{2}\|d\|^2 + o(\|d\|^2).$$

Então, para todo d tal que $\|d\|$ é suficientemente pequena, o primeiro termo do membro direito da desigualdade define o sinal desse lado. Mas

$$\frac{a}{2}\|d\|^2 > 0$$

Portanto, para $\|d\|$ suficientemente pequena e não nula ($0 < \|d\| < \epsilon$),

$$f(x^* + d) - f(x^*) > 0 \Rightarrow f(x^* + d) > f(x^*).$$

Assim, para todo $x \in B(x^*, \epsilon)$, $x \neq x^*$, temos que $f(x) > f(x^*)$. Logo, x^* é um minimizador local estrito de f . \square

Apesar das Proposições 1-3 terem fundamental importância na classificação de possíveis soluções de (1.1), a maioria das demonstrações de convergência de algoritmos busca apenas provar que os pontos limites das sequências geradas são pontos estacionários da função objetivo f . Usaremos então as palavras “convergir globalmente” para algoritmos que convergem para um ponto estacionário da função objetivo ou para alguma solução de um sistema linear. Diremos, neste caso, que o algoritmo é globalmente convergente.

1.1 Estratégias de Globalização

Atualmente existe um grande número de algoritmos eficientes para minimização irrestrita de funções, nos quais buscamos um ponto estacionário do problema (gostaríamos de um minimizador, mas nem sempre é possível). Todos esses algoritmos requerem que forneçamos uma estimativa inicial, que será denotada por x_0 . Começando em x_0 , algoritmos de otimização geram uma sequência de iterandos $\{x_k\}_{k=0}^{\infty}$, até quando nenhum progresso pode ser feito, ou quando parece que a solução foi aproximada com eficiência. Para decidir como passar de um iterando x_k para outro, x_{k+1} , os algoritmos usam informações sobre a função em x_k , e possivelmente algumas informações sobre os iterandos anteriores x_0, x_1, \dots, x_{k-1} .

Quando conhecemos bem o problema, somos capazes de fornecer um valor razoável para x_0 , próximo da solução. Nesta situação, podemos aplicar métodos dotados

de resultados de convergência local, ou seja, que geram a sequência $\{x_k\}$ tal que $\lim_{k \rightarrow \infty} x_k = x^*$, como os métodos clássicos que descreveremos na próxima seção.

Em outros casos, a estimativa inicial tem que ser escolhida de maneira arbitrária. Então, para obtermos resultados de convergência devemos encontrar um iterando x_{k+1} com um valor menor para a função que em x_k .

Existem algoritmos que são não-monótonos, ou seja, não insistem para que o valor da função decresça a cada iteração, mas exigem que a função decresça após um número fixo m de iterações: $f(x_k) < f(x_{k-m})$.

Existem duas estratégias de globalização fundamentais para passar de um iterando para o outro, que garantem um decréscimo na função objetivo, que são: *busca linear* e *região de confiança*.

Descreveremos brevemente estas estratégias, a seguir. Mais detalhes podem ser encontradas em [13, 26].

Busca Linear

Na estratégia de busca linear, escolhemos, a partir de x_k uma direção de busca d_k e determinamos, ao longo dessa direção, o próximo iterando x_{k+1} , que tenha um valor menor de função. Encontramos a distância que andaremos na direção d_k resolvendo aproximadamente o seguinte problema de uma dimensão

$$\min_{\alpha > 0} f(x_k + \alpha d_k), \quad (1.3)$$

determinando o que chamamos de *tamanho do passo* α .

Quando resolvemos (1.3) exatamente, encontramos o máximo de benefício na direção d_k , porém uma minimização exata é computacionalmente cara e além disso desnecessária. Ao invés disto, um algoritmo de busca linear gera um número limitado de pontos, até que encontramos um que aproximadamente minimiza (1.3), ou seja, fazemos uma busca linear inexata, com algumas condições dentre as quais destacamos duas: condições de Wolfe e *backtracking*.

No novo iterando, uma nova direção de busca e um novo tamanho de passo são calculados, e o processo se repete até encontrarmos um ponto estacionário da função objetivo, ou detectarmos que o problema não tem solução.

Condições de Wolfe

Uma busca linear inexata bastante popular estipula que α_k deve primeiramente satisfazer um decréscimo suficiente na função objetivo f , como mostramos a seguir:

$$f(x_k + \alpha d_k) \leq f(x_k) + c_1 \alpha \nabla f(x_k)^T d_k, \quad (1.4)$$

para alguma constante $c_1 \in (0, 1)$. Em outras palavras, a redução de f deve ser proporcional tanto ao tamanho do passo quanto à derivada direcional (ver apêndice B) $\nabla f(x_k)^T d_k$.

A desigualdade (1.4) muitas vezes é chamada de *condição de Armijo*. Esta condição sozinha, porém, não é suficiente para garantir que o algoritmo faça progressos suficientes, pois é aceito para valores bem pequenos de α . Para contornar esse problema, introduzimos uma segunda condição, chamada de *condição de curvatura*, que requer que α_k satisfaça

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq c_2 \nabla f(x_k)^T d_k, \quad (1.5)$$

para alguma constante $c_2 \in (c_1, 1)$.

As desigualdades (1.4) e (1.5) são conhecidas como condições de Wolfe. Um tamanho de passo pode satisfazer as condições de Wolfe sem estar necessariamente próximo de um minimizador de (1.3).

Backtracking

Em um algoritmo de busca linear podemos usar backtracking para escolher um bom candidato para o tamanho do passo. Usando backtracking também podemos dispensar a condição (1.5) e usar somente a condição de decréscimo suficiente para terminar o procedimento de busca linear.

Basicamente o procedimento de backtracking é o seguinte:

- 1 Escolher $\bar{\alpha} > 0$, ρ tal que $0 < \rho_{min} < \rho < \rho_{max} < 1$, $c \in (0, 1)$; fazer $\alpha \leftarrow \bar{\alpha}$;
- 2 Até que $f(x_k + \alpha d_k) \leq f(x_k) + c \alpha \nabla f(x_k)^T d_k$, fazer $\alpha \leftarrow \rho \alpha$
- 3 Terminar com $\alpha_k = \alpha$.

Neste procedimento, um tamanho de passo aceitável será encontrado em um número finito de iterações, pois α_k se tornará, eventualmente, pequeno o suficiente para que a condição de decréscimo suficiente seja aceita.

Região de Confiança

Nesta estratégia, as informações da função são usadas para construir um modelo quadrático q_k cujo comportamento próximo ao ponto x_k é similar ao da função objetivo f . Como o modelo q_k pode não ser uma boa aproximação para f quando x está longe de x_k , nós restringimos a busca por um minimizador de q_k em alguma vizinhança ao redor de x_k . Em outras palavras, nós encontramos o candidato a “passo” d resolvendo aproximadamente o subproblema:

$$\min_d q_k(x_k + d), \quad (1.6)$$

onde $x_k + d$ está na região de confiança.

Se o candidato não produz um decréscimo suficiente na função objetivo, concluímos que a região de confiança está muito grande, então a reduzimos e resolvemos novamente (1.6). Normalmente, dado $\Delta > 0$, a região de confiança é uma bola definida por $\|d\|_2 \leq \Delta$, ou uma caixa dada por $\|d\|_\infty \leq \Delta$. Δ é chamado de *raio da região de confiança*.

1.2 Métodos Clássicos para Minimização Irrestrita

Consideramos o problema 1.1.

1.2.1 Método de Máxima Descida (MMD) ou do Gradiente

Suponha $f \in C^1$, [18].

Antes de iniciarmos a explanação sobre o método, vamos destacar a direção $d = -\frac{\nabla f(x)}{\|\nabla f(x)\|}$. Essa direção é a de máxima descida a partir do ponto x . De fato, dada uma direção unitária qualquer d ($\|d\| = 1$), a derivada direcional (ver apêndice

B) $D_d f(x) = \nabla f(x)^T d$ é tal que:

$$|\nabla f(x)^T d| \leq \|\nabla f(x)\| \|d\| ,$$

então

$$-\|\nabla f(x)\| \|d\| \leq \nabla f(x)^T d \leq \|\nabla f(x)\| \|d\| .$$

Como

$$\frac{-\nabla f(x)^T \nabla f(x)}{\|\nabla f(x)\|} = -\|\nabla f(x)\| ,$$

e

$$\nabla f(x)^T d \geq -\|\nabla f(x)\| ,$$

então, dentre todas as direções unitárias, a determinada por $-\nabla f(x)$ é a que fornece a menor derivada direcional. Portanto, o valor da função objetivo diminuirá se avançarmos nessa direção, e a máxima redução será obtida minimizando, ao longo dela, a função f .

Tendo em mente esse resultado, o MMD é um método iterativo que calcula novas aproximações para a solução ao longo da direção oposta à do gradiente de f :

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

onde λ_k é o passo que vamos tomar na direção de $d_k = -\nabla f(x_k)$.

O escalar λ_k é o passo que fornece o menor valor para a função

$$\varphi(\lambda) = f(x_k + \lambda d_k). \quad (1.7)$$

Algoritmo 1. *MMD*

Se $x_k \in \mathbb{R}^n$ é tal que $\nabla f(x_k) \neq 0$, os passos para determinar x_{k+1} são:

Passo 1. Calcular $d_k = -\nabla f(x_k)$.

Passo 2. Determinar λ_k , minimizador de $f(x_k + \lambda d_k)$ sujeita a $\lambda \geq 0$.

Passo 3. Fazer $x_{k+1} = x_k + \lambda_k d_k$.

Teorema 1. Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $f \in C^2$. Seja $x^* \in \mathbb{R}^n$ um minimizador local de

f , tal que a matriz $\nabla^2 f(x^*)$ é definida positiva. Se o algoritmo do MMD está bem definido para todo $k \in N$ e a sequência $\{x_k\}$ gerada por ele converge a x^* , então a sequência $\{f(x_k)\}$ converge a $f(x^*)$ e:

$$\frac{(K - \kappa)}{(K + \kappa)^2} (f(x_k) - f(x^*)), (\text{convergência linear})$$

onde K e κ são o maior e menor autovalor de $\nabla^2 f(x^*)$, respectivamente.

Demonstração. Ver [23] □

1.2.2 Método de Newton

Suponha $f \in C^2$, [26].

O método de Newton ajusta f em uma aproximação x_k por uma quadrática:

$$q(x_k + d) = f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2 f(x_k) d \quad (1.8)$$

e obtém o ponto $x_+ = x_k + d^N$, onde $\nabla q(x_+) = 0$, condição necessária para que x_+ seja minimizador de q . Assim, $d^N = -(\nabla^2 f(x_k))^{-1} \nabla f(x_k)$, se $\nabla^2 f(x_k)$ for não singular.

Algoritmo 2. *Método de Newton*

Dados $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $f \in C^2$ e $x_0 \in \mathbb{R}^n$, seguir os seguintes passos, a cada iteração k :

Passo 1. Resolver

$$\nabla^2 f(x_k) d_k^N = -\nabla f(x_k),$$

Passo 2. Fazer

$$x_{k+1} = x_k + d_k^N.$$

Se $\nabla^2 f(x_k)$ é definida positiva, podemos garantir que d_k^N é direção de descida, $k \in \mathbb{N}$, e, sob outras hipóteses, temos o teorema a seguir.

Teorema 2. *Convergência do Método de Newton.*

Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $f \in C^3$. Seja x^* um minimizador local de f em \mathbb{R}^n tal que $\nabla^2 f(x^*) > 0$. Então existe $\epsilon > 0$ tal que, se $x_0 \in B(x^*, \epsilon)$, a sequência $\{x_k\}$ gerada pelo algoritmo anterior verifica:

$$(i) \nabla^2 f(x_k) > 0, \forall k = 0, 1, 2, \dots$$

$$(ii) \lim_{k \rightarrow \infty} x_k = x^*$$

$$(iii) \text{ Existe } c > 0 \text{ tal que } \|x_{k+1} - x^*\| \leq c\|x_k - x^*\|^2 \text{ (convergência quadrática).}$$

Demonstração. Ver [23]. □

Lembramos que a ordem de convergência quadrática é obtida devido ao uso de derivadas segundas. Temos então que avaliar $\nabla^2 f(x_k)$ e resolver $\nabla^2 f(x_k)d_k^N = -\nabla f(x_k)$. Portanto, o processo é caro computacionalmente e se o número de variáveis for muito grande, a memória para armazenar esta informação pode ser insuficiente.

1.2.3 Métodos quase-Newton

Suponha $f \in C^1$, [26].

Métodos quase-Newton, assim como o método da máxima descida, requerem apenas que o gradiente da função objetivo seja calculado a cada iteração. Os métodos quase-Newton são mais eficientes que o MMD em relação à taxa de convergência (como veremos no teorema 3), e como não exigem o cálculo de derivadas de segunda ordem, podem se tornar uma boa alternativa ao método de Newton.

Definimos a direção:

$$d_k = -H_k \nabla f(x_k),$$

onde $H_k \in \mathbb{R}^{n \times n}$ é uma matriz simétrica e uma aproximação para a Hessiana verdadeira da função. Se H_k for definida positiva, d_k é uma direção de descida.

É interessante determinarmos matrizes H_k de maneira que, em cada iteração k , o trabalho computacional do método resultante seja menor que o do método de Newton.

Em cada iteração de um método quase-Newton, atualizamos H_k , levando em conta informações adicionais obtidas durante a iteração. As atualizações usam o fato de que mudanças no gradiente nos dão informações sobre a segunda derivada da função ao longo da direção de busca.

Usando no teorema de Taylor (ver apêndice B) a fórmula (B.3) temos,

$$\nabla f(x + d) = \nabla f(x) + \nabla^2 f(x)d + \int_0^1 [\nabla^2 f(x + td) - \nabla^2 f(x)]d \, dt.$$

Como $\nabla f(\cdot)$ é contínuo, o tamanho do termo da integral é $o(\|d\|)$. Fazendo $x = x_k$ e $d = x_{k+1} - x_k$, temos

$$\nabla f(x_{k+1}) = \nabla f(x_k) + \nabla^2 f(x_{k+1})(x_{k+1} - x_k) + o(\|x_{k+1} - x_k\|).$$

Quando x_{k+1} e x_k estão numa vizinhança próxima à solução x^* , na qual ∇f é definida positiva, o termo final nesta expansão é dominado por $\nabla^2 f(x_{k+1})(x_{k+1} - x_k)$, e podemos escrever:

$$\nabla^2 f(x_{k+1})(x_{k+1} - x_k) \approx \nabla f(x_{k+1}) - \nabla f(x_k) \quad (1.9)$$

Nós escolhemos a aproximação da Hessiana H_{k+1} de maneira que satisfaça a propriedade (1.9) da Hessiana verdadeira, ou seja, queremos que satisfaça a seguinte condição, conhecida como *equação secante*

$$H_{k+1}s_k = y_k, \quad (1.10)$$

onde $s_k = x_{k+1} - x_k$ e $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$.

Atualização BFGS

A equação secante (1.10) pode ser satisfeita por um número sem fim de matrizes. Assim, na escolha da matriz H_k , geralmente impomos algumas outras condições, tais como simetria (motivada pela simetria da Hessiana verdadeira), e uma restrição de que a diferença entre duas aproximações sucessivas H_k e H_{k+1} tenha posto pequeno.

Uma fórmula muito usada que contempla esses requisitos é a formula BFGS, proposta por Broyden, Fletcher, Goldfarb e Shanno. É definida por:

$$H_{k+1} = H_k - \frac{H_k s_k s_k^T H_k}{s_k^T H_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}. \quad (1.11)$$

Note que a diferença entre as matrizes H_k e H_{k+1} é uma matriz de posto 2. Podemos mostrar também que a fórmula BFGS gera aproximações definidas positivas, quando a aproximação inicial H_0 é definida positiva e $s_k^T y_k > 0$. Para maiores detalhes ver [26].

Teorema 3. *Teorema de Convergência.*

Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $f \in C^3$ tal que existe x^* um minimizador local de f em \mathbb{R}^n com $\nabla^2 f(x^*) > 0$. Existem $\delta > 0$, $\epsilon > 0$ tais que, se $x_0 \in B(x^*, \delta)$ e $\|H_0 - \nabla^2 f(x^*)\| < \epsilon$, as sequências $\{x_k\}$ e $\{H_k\}$ geradas pelo método BFGS, verificam:

- (i) H_k é definida positiva para todo $k \in \mathbb{N}$;
- (ii) $\lim_{k \rightarrow \infty} x_k = x^*$;
- (iii) $\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0$ convergência superlinear.

Demonstração. Ver [13]. □

Capítulo 2

Minimização de Quadráticas Sem Restrições

Muitas vezes obtemos a solução de um problema mais complexo através da resolução de uma sequência de problemas de minimização de funções quadráticas, pois tratar esse tipo de função é relativamente simples. Além disso, por exemplo, para aproximarmos um conjunto de dados discretos por uma função, com parâmetros lineares, minimizamos a soma dos quadrados dos erros, que nesse caso é uma função quadrática.

Sendo assim, este trabalho está voltado à minimização de quadráticas, que são definidas no que segue.

Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$. f é uma função quadrática quando pode ser expressa como:

$$f(x) = \frac{1}{2}x^T Ax - b^T x + c, \quad (2.1)$$

onde $A \in \mathbb{R}^{n \times n}$ é uma matriz simétrica, $b \in \mathbb{R}^n$ e $c \in \mathbb{R}$.

Lema 1. Se $f(x) = \frac{1}{2}x^T Ax - b^T x + c$, então $\nabla f(x) = Ax - b$ e $\nabla^2 f(x) = A$ para todo $x \in \mathbb{R}^n$.

Neste caso, os pontos estacionários de f são as soluções do sistema linear

$$Ax - b = 0$$

Vamos destacar dois resultados sobre a existência de pontos estacionários de f :

Lema 2. 1. A função f admite algum ponto estacionário se, e somente se, b estiver no espaço coluna de A .

2. A função f admite um único ponto estacionário se, e somente se, A é não singular.

Proposição 4. Se $A \geq 0$ e x^* é um ponto estacionário de f , então x^* é minimizador global de f .

Demonstração. Seja x^* ponto estacionário de f , assim $b = Ax^*$. Considere $d \in \mathbb{R}^n$ arbitrário, logo:

$$\begin{aligned} f(x^* + d) &= \frac{1}{2}(x^* + d)^T A(x^* + d) - b^T(x^* + d) + c \\ &= \frac{1}{2}x^{*T}Ax^* + \frac{1}{2}d^TAd + d^TAx^* - b^Tx^* - b^Td + c \\ &= \frac{1}{2}x^{*T}Ax^* - b^Tx^* + c + \frac{1}{2}d^TAd \\ &\geq f(x^*). \end{aligned}$$

Portanto, $f(x^* + d) \geq f(x^*)$, para todo $d \in \mathbb{R}^n$.

□

Podemos concluir, observando os resultados acima, que se a matriz A é simétrica e definida positiva, o ponto onde o gradiente se anula é o único mínimo da função f .

Sob as mesmas hipóteses da proposição 4:

Corolário 1. Todo minimizador local de f é global.

Corolário 2. Se a matriz A é indefinida, então a quadrática f não tem extremos locais.

Quando não existem minimizadores da função f , dado um x arbitrário pertencente a \mathbb{R}^n , é útil determinar uma direção $d \in \mathbb{R}^n$ tal que

$$\lim_{t \rightarrow \infty} f(x + td) = -\infty. \quad (2.2)$$

Se soubermos achar uma direção que satisfaça (2.2) podemos dizer que sempre somos capazes de resolver

$$\text{minimizar } f(x) = \frac{1}{2}x^T Ax - b^T x + c \quad (2.3)$$

até quando o “mínimo é $-\infty$ ” (e o “minimizador é $x - \infty d$ ”).

2.1 Métodos Iterativos

Podemos minimizar uma função quadrática usando métodos diretos, como a decomposição espectral da matriz A ou utilizando fatorações, como a fatoração de Cholesky. Os métodos diretos (ver [20, 12]) calculam a solução de (2.3) em um único passo, mas através de um processo relativamente trabalhoso.

Os métodos iterativos computam uma sequência de pontos $x_k \in \mathbb{R}^n$, e a passagem de um iterando a outro se faz através de um conjunto de operações geralmente computacionalmente baratas, e a solução é obtida depois de um número finito de passos ou no limite.

O *método dos gradientes conjugados* [25] é o usado mais frequentemente para resolver (2.3), se A é definida positiva. Este método, quando a matriz A é definida positiva, converge em no máximo n iterações - onde n é o número de variáveis do problema. Assim, muitas vezes acaba sendo considerado um método direto. Nós optamos por tratá-lo como um método iterativo, principalmente por não exigirmos a positividade de A . Para motivá-lo, abordaremos o *método de máxima descida* [18].

2.1.1 Método de Máxima Descida (MMD) para Funções Quadráticas

Vamos tratar aqui do MMD especialmente para quadráticas.

Na seção (1.2), vimos que no MMD devemos encontrar λ_k , minimizador de uma

função de uma variável (1.7), o que nos leva a determinar o zero de $\varphi'(\lambda)$:

$$\varphi'(\lambda) = 0$$

$$\nabla f(x_k + \lambda d_k)^T d_k = 0$$

$$(A(x_k + \lambda d_k) - b)^T d_k = 0$$

$$(x_k^T A + \lambda d_k^T A + b^T) d_k = 0$$

$$x_k^T A d_k + \lambda d_k^T A d_k + b^T d_k = 0$$

$$(A x_k + b)^T d_k + \lambda d_k^T A d_k = 0$$

$$\nabla f(x_k)^T d_k + \lambda d_k^T A d_k = 0$$

$$\lambda_k = -\frac{\nabla f(x_k)^T d_k}{d_k^T A d_k}$$

Assim temos que:

$$\lambda_k = ((d_k)^T d_k) / ((d_k)^T A d_k).$$

Algoritmo 3. *MMD para quadráticas*

Considerar $x_0 \in \mathbb{R}^n$ e $\epsilon > 0$ dados;

Seja $r_0 = b - A x_0$, $\delta_0 = r_0^T r_0$, $k = 0$;

Seguir os seguintes passos enquanto $\delta_k > \epsilon$:

Passo 1. $d_k = -\nabla f(x_k)$;

Passo 2. $\lambda_k = \frac{\delta_k}{d_k^T A d_k}$;

Passo 3. $x_{k+1} = x_k + \lambda_k d_k$;

Passo 4. $r_{k+1} = -\nabla f(x_{k+1})$;

Passo 5. $\delta_{k+1} = r_{k+1}^T r_{k+1}$; $k = k + 1$

Teorema 4. *Teorema de Convergência.*

Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função quadrática com matriz Hessiana A definida positiva.

Seja x^* o minimizador global de f .

Dado $x_0 \in \mathbb{R}^n$, arbitrário, o algoritmo do MMD gera uma sequência $\{x_k\}$ tal que:

1. $\lim_{k \rightarrow \infty} x_k = x^*$,
2. $\lim_{k \rightarrow \infty} f(x_k) = f(x^*)$ e $f(x_{k+1}) - f(x^*) \leq ((K - \kappa)/(K + \kappa))^2 (f(x_k) - f(x^*))$,

onde K e κ são o maior e menor autovalor de A , respectivamente.

Demonstração. Ver [23]. □

Esse resultado nos garante que dado $x_0 \in \mathbb{R}^n$ arbitrário, se A for simétrica e definida positiva, o MMD converge ao único ponto de mínimo de f .

No entanto, o MMD pode se tornar muito lento, pois as direções geradas consecutivamente são **ortogonais**:

$$\varphi'(\lambda_k) = 0 \iff \nabla f(x_k + \lambda_k d_k)^T d_k = 0 \iff -\nabla f(x_{k+1})^T \nabla f(x_k) = 0,$$

podendo levar a um movimento de zig-zag mesmo perto da solução. Este comportamento se torna desfavorável à medida que as superfícies de nível de f se tornam mais alongadas, o que corresponde a um número de condição grande da matriz A . Outros métodos, trabalhando com diferentes direções, podem acelerar o processo de convergência. Dessa maneira, vamos introduzir o método dos gradientes conjugados, como uma espécie de “MMD com memória”.

2.1.2 Método dos Gradientes Conjugados

Estamos resolvendo (2.3). A solução desse problema é o vetor que resolve $Ax = b$. Sabemos que $\nabla f(x) = Ax - b$ e esse vetor equivale ao oposto do resíduo do sistema

linear: $r(x) = b - Ax$.

Um dos inconvenientes do método de máxima descida é tomar as direções ortogonais. O Método dos Gradientes Conjugados resolve essa questão usando direções A -conjugadas (definidas no próximo teorema). Esse método foi originalmente proposto por Hestenes e Stiefel, em 1952, ver [21].

O próximo teorema estabelece a propriedade da A -conjugação, ou A -ortogonalidade, satisfeita pelos incrementos s_k .

Teorema 5. *Se $\{x_k\}$ é uma sequência gerada pelo Algoritmo 2, os incrementos $s_k = x_{k+1} - x_k$, $k = 0, 1, \dots$ são A -conjugados, isto é, para todo $k \geq 1$ vale*

$$s_j^T A s_k = 0, \quad j = 0, 1, \dots, k-1. \quad (2.4)$$

E, se $\nabla f(x_0), \nabla f(x_1), \dots, \nabla f(x_{k-1})$ são não nulos e x_k está bem definido, então

$$s_j^T A s_j > 0 \text{ para todo } j = 0, 1, \dots, k-1. \quad (2.5)$$

Demonstração. Já vimos que $\nabla f(x_{k+1}) \perp S_k$, onde

$$S_k = \text{span} \{ \nabla f(x_0), \dots, \nabla f(x_k) \} = \text{span} \{ s_0, \dots, s_k \}.$$

Então,

$$\nabla f(x_{k+1}) \perp s_j, \quad j = 0, 1, \dots, k. \quad (2.6)$$

Pela definição de s_k , e por cálculos elementares,

$$\nabla f(x_{k+1}) = \nabla f(x_k) + A s_k. \quad (2.7)$$

Pré-multiplicando (2.7) por s_j^T , para $j = 0, 1, \dots, k-1$, por (2.6) segue (2.4).

Provaremos agora (2.5).

Se $\nabla f(x_j) \neq 0$, temos que x_{j+1} está bem definido, portanto $s_j \neq 0$ e, por consequência, $\nabla f(x_j)^T s_j < 0$. Mas, pela definição de x_{j+1} , $t = 1$ deve ser minimizador de $f(x_j + ts_j)$. Como essa função de t é uma parábola, para que exista um minimizador há duas possibilidades: ou a parábola é constante, ou o coeficiente de segunda ordem é maior que zero. Sabemos que $\frac{d}{dt}f(x_j + ts_j) = \nabla f(x_j)^T s_j < 0$ em $t = 0$, portanto a parábola não é constante. Como o coeficiente de segunda ordem é $\frac{s_j^T A s_j}{2}$, segue (2.5). \square

No Método dos Gradientes Conjugados, a primeira iteração será dada pelo MMD, ou seja,

$$x_1 = x_0 + \alpha_0 d_0 = x_0 - \alpha_0 \nabla f(x_0),$$

onde α_0 é escolhido de modo a minimizar (2.1) na direção $d_0 = -\nabla f(x_0)$.

Geraremos então uma sequência x_k para o algoritmo:

Algoritmo 4. *Começamos o algoritmo com $x_0 \in \mathbb{R}^n$ arbitrário. Dado $x_k \in \mathbb{R}^n$, definimos:*

$$S_k = \text{span} \{-\nabla f(x_0), \dots, -\nabla f(x_k)\}$$

e

$$V_k = x_0 + S_k = \{v \in \mathbb{R}^n / v = x_0 + w, \text{ com } w \in S_k\}.$$

Consideramos o problema

$$\min f(x) \text{ sujeita a } x \in V_k. \quad (2.8)$$

Se (2.8) não tem solução, o algoritmo pára por inexistência de mínimo. Caso contrário, definimos x_{k+1} como uma das soluções de (2.8). (De fato, (2.8) não pode ter mais de uma solução).

A cada iteração k , calculamos

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

onde s_k é o passo na iteração k .

Por construção, $s_k \in S_k$. Além disso, $s_{k+1} \notin S_k$. Podemos concluir que $S_k = \text{span}\{s_0, s_1, \dots, s_k\}$. Para maiores detalhes ver [25].

Agora veremos que um algoritmo para resolução de (2.8) possui duas possibilidades: que em algum momento ele pare por inexistência do minimizador de $f(x)$ em V_k ou que, em um número finito de passos (menor ou igual a n), encontre uma solução do sistema linear $Ax - b = 0$. Quando A é definida positiva ou semidefinida positiva, e $b \in R(A)$, os minimizadores do problema (2.8) sempre existem. Portanto, nesses casos, o algoritmo termina com uma solução de $Ax - b = 0$, que necessariamente é minimizador global de (2.3). Se $b \notin R(A)$, não existem soluções de $Ax - b = 0$. Logo, nesse caso, o algoritmo pára por inexistência de mínimo de (2.8) em alguma iteração k . Agora, se $b \in R(A)$, mas A tem algum autovalor negativo, as duas possibilidades permanecem: que seja encontrada uma iteração que resolva $Ax - b = 0$ (ponto crítico de (2.3)) ou que o algoritmo pare por inexistência de mínimo. Observemos o seguinte teorema:

Teorema 6. *Se o algoritmo 2 não pára por inexistência de mínimo, então existe $k \leq n$ tal que x_k é uma solução do sistema $Ax - b = 0$ (ponto estacionário de (2.3)).*

Demonstração. Suponhamos que o algoritmo 2 não pare por inexistência de mínimo. Então, para cada iteração k em que $\nabla f(x_{k+1})$ é não nulo, temos que

$$\dim(V_{k+1}) = \dim(V_k) + 1.$$

Assim, se chegarmos a completar n iterações com gradientes não nulos, teremos $\dim(V_{n-1}) = n$. Isso implica que $V_{n-1} = \mathbb{R}^n$ e, portanto, x_n é solução de (2.3). □

Mostraremos agora como calcular as direções de busca, de uma maneira eficiente, para que elas sejam mutuamente A -conjugadas.

Sabemos que $S_k = \text{span}\{s_0, s_1, \dots, s_{k-1}, -\nabla f(x_k)\}$, de modo que:

$$s_k = \lambda_0 s_0 + \lambda_1 s_1 + \dots + \lambda_{k-1} s_{k-1} - \lambda_k \nabla f(x_k).$$

Defina

$$s_k = \alpha_k d_k,$$

onde d_k é o vetor direção (w_0, \dots, w_{k-1} são escalares)

$$d_k = w_0 s_0 + w_1 s_1 + \dots + w_{k-1} s_{k-1} - \nabla f(x_k).$$

Pré-multiplicando os membros por $s_j^T A, j = 0, 1, \dots, k-1$, e usando a A-conjugação dos s_j , obtemos:

$$0 = s_j^T A d_k = -s_j^T A \nabla f(x_k) + w_j s_j^T A s_j.$$

Agora, usando que $s_j^T A s_j > 0$,

$$w_j = \frac{\nabla f(x_k)^T A s_j}{s_j^T A s_j}, \text{ para } j = 0, 1, \dots, k-1.$$

Como $A s_j = \nabla f(x_{j+1}) - \nabla f(x_j)$, temos que $\nabla f(x_k)^T A s_j = 0$ para $j = 0, 1, \dots, k-2$. Logo, $w_j = 0$ para $j = 0, 1, \dots, k-2$ e, consequentemente:

$$d_k = -\nabla f(x_k) + w_{k-1} s_{k-1} = -\nabla f(x_k) + \frac{\nabla f(x_k)^T A s_{k-1}}{s_{k-1}^T A s_{k-1}} s_{k-1}. \quad (2.9)$$

Lembrando que $A s_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1})$, e $\nabla f(x_k) \perp \nabla f(x_{k-1})$, da expressão (2.9) se deduz que:

$$d_k = -\nabla f(x_k) - \frac{\nabla f(x_k)^T \nabla f(x_k)}{s_{k-1}^T \nabla f(x_{k-1})} s_{k-1} = -\nabla f(x_k) - \frac{\nabla f(x_k)^T \nabla f(x_k)}{d_{k-1}^T \nabla f(x_{k-1})} d_{k-1}.$$

Além disso, como d_{k-1} é a soma de $-\nabla f(x_{k-1})$ mais uma combinação linear dos gradientes anteriores, e esses gradientes são ortogonais a $\nabla f(x_{k-1})$, a expressão anterior toma a forma:

$$d_k = -\nabla f(x_k) + \beta_{k-1} d_{k-1}, \text{ onde } \beta_{k-1} = \frac{\nabla f(x_k)^T \nabla f(x_k)}{\nabla f(x_{k-1})^T \nabla f(x_{k-1})}. \quad (2.10)$$

Finalmente, sabemos que x_{k+1} deve ser o minimizador de f ao longo da reta que passa por x_k , com direção d_k , obtemos então:

$$x_{k+1} = x_k + \alpha_k d_k \text{ onde } \alpha_k = \frac{\nabla f(x_k)^T \nabla f(x_k)}{d_k^T A d_k}. \quad (2.11)$$

e,

$$x_{k+1} - x_k = s_k = \alpha_k d_k = \frac{d_k^T \nabla f(x_k)}{d_k^T A d_k} d_k.$$

As expressões (2.10) e (2.11) descrevem o algoritmo de gradientes conjugados de maneira mais operativa. Observando as expressões a partir de (2.9), o fato mais relevante mostrado é que o cálculo de x_{k+1} , quando esse ponto está bem definido, depende apenas do incremento anterior s_{k-1} e do gradiente atual $\nabla f(x_k)$. Ou seja, a minimização de f na variedade V_k pode ser efetuada com trabalho e memória mínimos.

Algoritmo 5. *MGC*

Considerar $x_0 \in \mathbb{R}^n$ e $\epsilon > 0$ dados;

Seja $d_0 = r_0 = b - Ax_0$ e $\delta_0 = r_0^T r_0$, $k = 0$;

Seguir os seguintes passos, enquanto $\delta_0 > \epsilon$

Passo 1. $h_k = A d_k$;

Passo 2. $\alpha_k = \frac{\delta_0}{d_k^T h_k}$;

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

Passo 4. $r_{k+1} = r_k + \alpha_k h_k$;

Passo 5. $\delta_1 = r_{k+1}^T r_{k+1}$;

Passo 6. se $\delta_1 < \epsilon$, saída com $x^* \cong x_{k+1}$;

Passo 7. $\beta_k = \frac{\delta_1}{\delta_0}$;

Passo 8. $\delta_0 = \delta_1$;

Passo 9. $d_{k+1} = r_{k+1} + \beta_k d_k$, $k = k + 1$.

Em cada iteração do algoritmo acima, o ponto x_k encontrado é o mínimo da quadrática restrita à variedade linear $x_0 + K_k$, onde K_k é o conjunto dado por

$$K_k = \text{span} \{ \nabla f(x_0), A \nabla f(x_0), \dots, A^{k-1} \nabla f(x_0) \},$$

conhecido como o k -ésimo **subespaço de Krylov** da matriz A .

É importante observar que a dimensão de tal subespaço é dada pelo número de autovalores distintos da matriz A , o que indica que o método dos gradientes conjugados encontra a solução exata para o problema de minimização ou calcula uma direção ao longo da qual a quadrática tende a $-\infty$ em, no máximo, p iterações, onde p é o número de autovalores distintos de A [25].

Apesar desse resultado, o método dos gradientes conjugados pode se tornar extremamente lento para problemas de grande porte em duas situações: quando os autovalores distintos de A são muitos e quando o número de condição da matriz A é grande.

Voltando à memória ocupada: são necessários vetores de dimensão n para armazenar h , x , r e d , e uma matriz de dimensão $n \times n$ para armazenar A (numa implementação para sistemas esparsos, o armazenamento de A depende do seu número de elementos não nulos e da estrutura de dados usada).

Quanto ao esforço computacional gasto: temos, por iteração, um produto de matriz por vetor (Ad_k) e dois produtos internos, além das multiplicações por escalar e adições envolvidas nas fórmulas recursivas. Logo, o número de operações, por iteração, é $O(n^2)$.

2.1.3 Método do Gradiente Espectral

Barzilai e Borwein [4] propuseram, em 1988, um método do tipo gradiente que requer pouco esforço computacional, e que parece ser menos sensível ao mau-condicionamento. Este método não requer armazenamento de matrizes ou buscas lineares. A direção de busca é sempre o oposto ao gradiente, mas a escolha do tamanho do passo não é a mesma do MMD. Na realidade, Barzilai e Borwein observaram que a nova escolha do tamanho do passo acelerava a convergência do MMD para quadráticas.

Adotaremos a notação simplificada $g(x_k) = \nabla f(x_k)$, em alguns pontos do texto.

Estamos tratando da iteração $x_{k+1} = x_k - B_k \nabla f(x_k)$, onde $B_k = \frac{1}{\sigma_k} I$, e σ_k minimiza $\|s_{k-1} - \sigma y_{k-1}\|^2$, onde $s_{k-1} = x_k - x_{k-1}$ e $y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1})$. A motivação para essa escolha é que ela traz uma aproximação para a equação secante $B_k s_{k-1} = y_{k-1}$. Temos então a iteração:

$$x_{k+1} = x_k - \frac{1}{\sigma_k} g(x_k)$$

onde σ_k , o passo espectral, é dado por:

$$\sigma_k = \frac{\langle y_{k-1}, y_{k-1} \rangle}{\langle s_{k-1}, y_{k-1} \rangle}. \quad (2.12)$$

Por simetria, podemos minimizar $\|\sigma s_{k-1} - y_{k-1}\|^2$, com respeito a σ e teremos então:

$$\sigma_k = \frac{\langle s_{k-1}, y_{k-1} \rangle}{\langle s_{k-1}, s_{k-1} \rangle} \quad (2.13)$$

Como estamos trabalhando com funções quadráticas, as equações (2.12) e (2.13) se tornam, respectivamente [29]:

$$\sigma_k = \frac{s_{k-1}^T A^2 s_{k-1}}{s_{k-1}^T A s_{k-1}} \quad (2.14)$$

e

$$\sigma_k = \frac{s_{k-1}^T A s_{k-1}}{s_{k-1}^T s_{k-1}} \quad (2.15)$$

Nesse caso, σ_k é o quociente de Rayleigh de A , em consequência:

$$0 < \lambda_{min} \leq \sigma_k \leq \lambda_{max},$$

onde λ_{min} e λ_{max} são respectivamente o menor e o maior autovalores de A .

Barzilai e Borwein [4] apresentaram a convergência do método para quadráticas em duas dimensões. Raydan [29] estendeu a aplicação do método para funções quadráticas estritamente convexas (ver apêndice B) em \mathbb{R}^n com resultados de convergência global. Este resultado foi estendido para quadráticas convexas em \mathbb{R}^n (não

necessariamente estritamente), com restrições de caixas, em [19].

O algoritmo que apresentaremos aqui, proposto em [30], insere o método de Barzilai e Borwein em uma estratégia de globalização, que aceita o passo espectral sempre que possível, para que possa ser aplicado para qualquer tipo de função. Essa estratégia é baseada em uma busca linear não monótona.

Aplicaremos a (2.3) o Método do Gradiente Espectral (MGE), que irá gerar uma sequência de pontos factíveis x_k , $k = 0, 1, 2, \dots$, onde x_0 é dado e $\sigma_0 = 1$. Daremos a seguir os passos do algoritmo do MGE, no qual usaremos os parâmetros:

- $M \geq 0$, $\delta \geq 0$;
- $\gamma \in (0, 1)$, o parâmetro da busca linear não monótona, e
- $0 < \alpha_1 < \alpha_2 < 1$, $0 < \epsilon < 1$ parâmetros de salvaguarda.

Algoritmo 6. *MGE*

Seja $k = 0$.

Passo 1. *Se $\|g(x_k)\| = 0$, parar;*

Passo 2. *Se $\sigma_k \leq \epsilon$ ou $\sigma_k \geq \frac{1}{\epsilon}$, faça $\sigma_k = \delta$;*

Passo 3. *Faça $\lambda = \frac{1}{\sigma_k}$;*

Passo 4. *Busca linear não monótona*

Considere $0 \leq j \leq \min(k, M)$. Se

$$f(x_k - \lambda g(x_k)) \leq \max_j \{(f_{k-j}) - \gamma \lambda g(x_k)^T g(x_k)\}.$$

Então faça

$$\lambda_k = \lambda,$$

$$x_{k+1} = x_k - \lambda_k g(x_k);$$

Caso contrário, vá para o passo 5.

Passo 5. *Escolha $\sigma \in [\alpha_1, \alpha_2]$, faça $\lambda = \sigma \lambda$ e vá para o passo 4;*

Passo 6. *Faça $\sigma_{k+1} = -\frac{g(x_k)^T y_k}{\lambda_k g(x_k)^T g(x_k)}$, $k = k + 1$ e vá para o passo 1.*

Observações:

- Desde que $s_k = -\lambda_k g(x_k)$, então a definição de σ_{k+1} é equivalente a (2.15).
- Para $k = 0$ a condição do passo 4 se reduz à condição de Armijo. Para $k > 0$ a função objetivo poderá aumentar de valor em algumas iterações, com respeito às iterações anteriores. Porém, $f(x_k) \leq f(x_0)$, $\forall k$, então o conjunto $\{x | f(x) \leq f(x_0)\}$ contém a sequência toda de iterandos $\{x_k\}$.

As propriedades de convergência do MGE são enunciadas no seguinte teorema:

Teorema 7. *Assuma que $\Omega_0 = \{x | f(x) \leq f(x_0)\}$ é um conjunto limitado. Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ continuamente diferenciável em alguma vizinhança N de Ω_0 . Seja $\{x_k\}$ a sequência gerada pelo algoritmo MGE. Então $g(x_j) = 0$ para algum j finito, ou as seguintes propriedades acontecem:*

- (i) $\lim_{k \rightarrow \infty} \|g(x_k)\| = 0$;
- (ii) *nenhum ponto de $\{x_k\}$ é um máximo local de f ;*
- (iii) *se o número de pontos estacionários de f em Ω_0 é finito, então a sequência $\{x_k\}$ converge.*

Demonstração. Ver [30]. □

Para outras consequências e resultados do algoritmo acima, ver [7].

Capítulo 3

Minimização de Quadráticas em Caixas

Quando tratamos de problemas em que os dados só nos interessam em uma determinada região do espaço, podemos falar de um problema de minimização com restrições. O caso mais comum é quando cada parâmetro não pode ser inferior a uma determinada cota e nem superior a outra. Assim, o conjunto de restrições toma a forma:

$$l_i \leq x_i \leq u_i$$

ou, considerando os vetores l , x e u :

$$l \leq x \leq u.$$

O conjunto $\Omega \subset \mathbb{R}^n$ formado pelos pontos que satisfazem essas restrições é chamado de caixa de \mathbb{R}^n .

Nosso problema será agora:

$$\min f(x) = \frac{1}{2}x^T Ax - b^T x + c \tag{3.1}$$

$$\textit{sujeita a } x \in \Omega,$$

onde $\Omega = \{x \in \mathbb{R}^n | l \leq x \leq u, l < u\}$.

É conveniente admitir os valores $-\infty$ para l_i e $+\infty$ para u_i , já que, às vezes, apenas algumas variáveis estão naturalmente limitadas, e para outras, a limitação é apenas superior, ou inferior.

Dado um vetor $x \in \Omega$, dizemos que a restrição i está ativa se $x_i = l_i$ ou $x_i = u_i$. Assim, montamos o subconjunto $I(x)$ de $\{1, 2, \dots, 2n\}$, de maneira que:

$$i \in I(x) \iff x_i = l_i, n + i \in I(x) \iff x_i = u_i.$$

Podemos então definir uma face aberta F_I como:

$$F_I = \{x \in \Omega \mid x_i = l_i, \text{ se } i \in I, x_i = u_i, \text{ se } n + i \in I, l_i < x_i < u_i \text{ nos outros casos}\}$$

Chamaremos \bar{F}_I o fecho de cada face aberta e $V(F_I)$ a menor variedade afim que contém F_I .

Para cada $x \in \Omega$ definimos o gradiente projetado negativo, ou “vetor de Cauchy” $g_p(x) \in \mathbb{R}^n$ como

$$g_p(x)_i = \begin{cases} 0 & \text{se } x_i = l_i \text{ e } [\nabla f(x)]_i > 0 \\ 0 & \text{se } x_i = u_i \text{ e } [\nabla f(x)]_i < 0 \\ -[\nabla f(x)]_i & \text{nos outros casos} \end{cases}$$

Podemos verificar que se x é minimizador local ou global de (3.1), teremos $g_p(x) = 0$. Se $A \geq 0$, a quadrática é convexa e esse resultado passa a ser uma condição suficiente para minimizador global.

Quando restringimos a função quadrática a uma face aberta F_I , as variáveis livres são apenas as que se encontram estritamente entre os limites definidos pelo conjunto I . O vetor definido a seguir é o inverso aditivo em relação a essas variáveis livres. Assim, para cada $x \in F_I$ definimos $g_I(x) \in \mathbb{R}^n$ (gradiente interno negativo), como:

$$g_I(x)_i = \begin{cases} 0 & \text{se } i \in I \text{ ou } n + i \in I \\ -[\nabla f(x)]_i & \text{nos outros casos} \end{cases}$$

A outra componente de $g_p(x)$ será chamada de gradiente chopado e é definida por:

$$g_I^C(x) = \begin{cases} 0 & \text{se } i \notin I \text{ e } n+i \notin I \\ 0 & \text{se } i \in I \text{ e } [\nabla f(x)]_i > 0 \\ 0 & \text{se } n+i \in I \text{ e } [\nabla f(x)]_i < 0 \\ -[\nabla f(x)]_i & \text{nos outros casos} \end{cases}$$

Podemos verificar que, para todo $x \in F_I$, o gradiente interno é ortogonal ao gradiente chopado, e:

$$g_p(x) = g_I(x) + g_I^C(x).$$

A figura 1 nos mostra um exemplo onde o gradiente chopado é menor que o gradiente projetado.

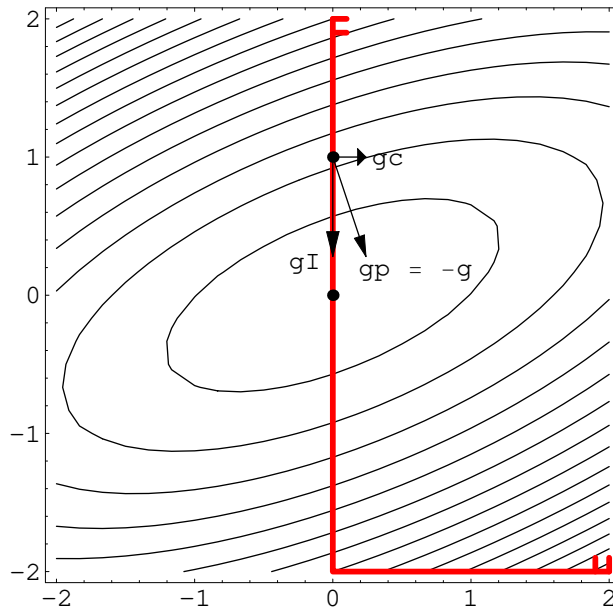


Figura 1

Por outro lado, na figura 2 a seguir, a norma do gradiente chopado é somente um pouco menor que a do gradiente projetado.

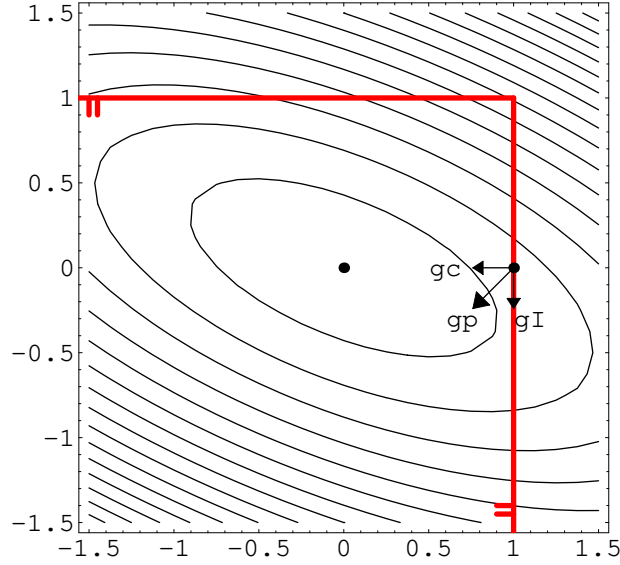


Figura 2

3.1 Algoritmo Geral para Quadráticas em Caixas

Apresentaremos um algoritmo para minimizar quadráticas em caixas que produz uma sequência $\{x_k\}$ de aproximações da solução de (3.1) baseado na minimização parcial da quadrática nas diferentes faces visitadas [25]. Quando x_k pertence a uma face F_I , um “algoritmo interno” para minimização de quadráticas sem restrições será acionado, trabalhando apenas com as variáveis livres da face. Esse “algoritmo interno” encontra um ponto estacionário restrito a F_I ou viola as restrições inativas dessa face.

Em cada passo do “algoritmo interno”, verificamos se ele já está bastante perto de um ponto estacionário em F_I , comparando o tamanho do gradiente chopado com o tamanho do gradiente projetado. Se o quociente entre ambos for grande (o valor máximo é 1), significa que o gradiente interno é pequeno em relação ao gradiente chopado e, portanto, continuar explorando a face F_I é pouco econômico, ou seja, abandonar a face em que estamos parece razoável. Para isso utilizaremos a direção do gradiente chopado.

Diremos que um algoritmo para minimizar $f(x)$ em $V(F_I)$, problema essencialmente irrestrito, tem as “propriedades boas para a minimização em caixas” quando produz uma sequência (talvez finita) $\{z_0, z_1, z_2, \dots\} \subset V(F_I)$, $z_0 \in F_I$ que cumpre:

1. Se z_k e z_{k+1} estão definidos, então $f(z_{k+1}) < f(z_k)$.
2. Se z_{k+1} não está definido (a sequência termina em z_k), pode ser por dois motivos: z_k é um ponto estacionário da minimização de $f(x)$ em $V(F_I)$ ou foi encontrada uma direção d_K tal que $\lim_{t \rightarrow \infty} f(z_k + td_k) = -\infty$.

Neste caso, se $z_k + td_k \in \Omega$ para todo t , há inexistência de solução de (3.1). Se $z_k + td_k \notin \Omega$ para t suficientemente grande, escolhe-se o último $z_{k+1} = z_k + td_k \in \Omega$ tal que $f(z_{k+1}) < f(z_k)$ e dá-se por terminada a sequência gerada pelo algoritmo interno em z_{k+1} .

3. Se a sequência $\{z_k\}$ é infinita, então todo ponto limite desta é um ponto estacionário de f sujeita a $V(F_I)$. Se não existem pontos limite deve-se satisfazer $\lim_{k \rightarrow \infty} f(z_k) = -\infty$

Vimos que o MGC termina em um ponto estacionário em um número finito de passos ou gera uma direção ao longo da qual a quadrática tende a $-\infty$, satisfazendo, portanto, as condições 1, 2 e 3. Em [5] são estudados outros métodos iterativos que satisfazem essas condições em determinadas circunstâncias.

Algoritmo 7. *Minimização de Quadráticas em Caixas*

Seja $\eta \in (0, 1)$ dado independentemente de k , e $x_0 \in \Omega$ um ponto inicial arbitrário. O algoritmo define uma sequência $\{x_k\}$ em Ω e pára se $\|g_p(x_k)\| = 0$. Suponhamos que $x_k \in F_I$ é tal que $\|g_p(x_k)\| \neq 0$. Seja $I = I(x_k)$ tal que $x_k \in F_I$. Chamemos $\Psi(x) \in \Omega$ ao minimizador de f ao longo do segmento (talvez semi-reta) $\{x \in \Omega | x = x_k + tg_I^C(x_k), t \geq 0\}$.

Os seguintes passos definem o procedimento para encontrar x_{k+1} .

Passo 1.

Começamos com $z_0 = x_k$ e usamos um método com as propriedades 1, 2, 3 vistas acima para minimizar quadráticas, aplicado essencialmente ao problema irrestrito de minimizar $f(x)$ em $V(F_I)$, obtendo assim $z_0 = x_k$, $z_1 = x_{k+1}$, e assim por diante.

Interromper esse método quando x_k satisfaz uma das seguintes condições:

(i)

$$x_k \in \Omega \text{ e } \|g_p(x_k)\| = 0. \quad (3.2)$$

(ii) O método interno detectou que (3.1) é ilimitado inferiormente.

(iii)

$$\|g_I^C(x_k)\| > \eta \|g_p(x_k)\|. \quad (3.3)$$

(iv)

$$z_{r+1} \notin \Omega, \text{ onde } r = 0, 1, 2, \dots \quad (3.4)$$

Passo 2.

Se o algoritmo interno foi interrompido por (3.2), parar, pois x_k é um ponto estacionário de (3.1).

Se o algoritmo interno detecta que (3.1) não tem solução, o algoritmo principal é interrompido com este diagnóstico.

Passo 3.

Se o teste (3.3) foi satisfeito em x_k , e f não é limitada inferiormente no segmento (nesse caso necessariamente semi-reta) $\{x_k + tg_I^C(x_k), t \geq 0\}$ o problema (3.1) não tem solução. Nesse caso, parar.

Caso contrário, calcular $x_{k+1} = \Psi(x_k)$.

Passo 4.

Se $x_k = z_r$ e z_{r+1} viola os limites de \bar{F}_I (condição (3.4)), encontrar x_{k+1} na fronteira de F_I ($\bar{F}_I - F_I$) tal que $f(x_{k+1}) < f(x_k)$ ou detectar que o problema (3.1) não tem solução.

Observando a figura 1 na seção anterior e a desigualdade (3.3) vemos que devemos continuar na mesma face, onde está localizada a solução do problema. E na figura 2 temos um caso em que saímos da face corrente, na direção do gradiente chopado, que neste caso aponta para a solução.

O algoritmo de minimização de quadráticas em caixas pára, isto é, gera uma sequência finita quando encontra um ponto estacionário x_k de (3.1) ou quando detecta que o problema é ilimitado inferiormente, e, portanto, sem solução. Porém, se o algoritmo gera uma sequência infinita, haverá, essencialmente, as mesmas duas possibilidades: encontraremos um gradiente projetado arbitrariamente pequeno, ou a sequência dos valores da função em x_k tenderá a $-\infty$.

Para demonstrações de convergência e maiores detalhes sobre o algoritmo, vide [25].

3.2 GENCAN

GENCAN é um método de otimização proposto em 2001 por Ernesto G. Birgin e José Mario Martinez, em [6], que significa **G**eneralização do **Q**UACAN, ou seja, é um método para funções gerais canalizadas. O método resolve problemas do tipo:

$$\min f(x)$$

$$\text{sujeita a } x \in \Omega,$$

onde $\Omega = \{x \in \mathbb{R}^n | l \leq x \leq u, l < u\}$.

Nosso interesse é aplicar o GENCAN nos casos onde f é uma função quadrática.

O esquema básico do GENCAN consiste em usar uma estratégia de restrições

ativas, conforme explicado na seção anterior. Para a minimização interna nas faces é introduzido um novo algoritmo com uma busca linear que combina backtracking e extrapolação. Para abandonar a face, usamos o gradiente espectral projetado.

Para o algoritmo interno, GENCAN utiliza um procedimento do tipo Newton-Truncado. Isso significa que a direção de busca será, para uma função objetivo qualquer, uma aproximação do minimizador da função quadrática na face corrente. GENCAN usa o Método dos Gradientes Conjugados para encontrar tal direção, de modo que a primeira iteração é obviamente uma direção de descida e esta propriedade é monitorada nas iterações seguintes por sucessivos passos de gradientes conjugados.

Utilizamos aqui a notação $s_k = x_k - x_{k-1}$, $g(x_k) = \nabla f(x_k)$ e $y_k = g(x_k) - g(x_{k-1})$. Temos o seguinte esquema:

1. Verificamos se o iterando x_k deve abandonar a face F_I ou se deve permanecer no fecho dessa face (\bar{F}_I), descrito pelo algoritmo 8 a seguir.
2. Caso deva abandonar a face, fazemos uma iteração do Método do Gradiente Espectral Projetado.
3. Caso contrário fazemos uma iteração do Algoritmo de Busca Linear.

Algoritmo 8. *GENCAN*

Seja $x_0 \in \Omega$ um ponto inicial arbitrário, $\eta \in (0, 1)$, $0 < \sigma_{\min} \leq \sigma_{\max} \leq \infty$, $\sigma_k = \max \left\{ 1, \frac{\|x_k\|}{\|g_p(x_k)\|} \right\}$. Considere F_I a face que contém o atual iterando x_k . Assumimos que $g_p(x_k) \neq 0$, pois caso contrário o algoritmo pára.

Passo 1. *Teste para abandonar ou não a face:*

$$\|g_I(x_k)\| \geq \eta \|g_p(x_k)\|. \quad (3.5)$$

Se (3.5) ocorre, o novo iterando vai permanecer no fecho da face corrente; então calculamos x_{k+1} realizando uma iteração do Algoritmo de Busca Linear, relativo às variáveis livres na face.

Se (3.5) não ocorre, julgamos que alguma restrição deve ser abandonada, e então o novo iterando x_{k+1} é calculado a partir de uma iteração do Método do Gradiente Espectral. Nesse caso, antes de calcularmos x_{k+1} , obtemos o coeficiente espectral da seguinte maneira:

Passo 2. Se $s_k^T y_k \leq 0$, então

$$\sigma_k = \max \left\{ 1, \frac{\|x_k\|}{\|g_p(x_k)\|} \right\}.$$

Caso contrário, definimos:

$$\bar{\sigma}_k = \frac{s_k^T y_k}{s_k^T s_k}$$

e

$$\sigma_k = \min \{ \sigma_{\max}, \max \{ \sigma_{\min}, \bar{\sigma}_k \} \}.$$

A seguir descrevemos o algoritmo do gradiente espectral projetado. Neste método, o iterando x_{k+1} é obtido usando o passo espectral σ_k e um critério monótono de decréscimo suficiente.

Algoritmo 9. *Método do Gradiente Espectral Projetado*

Dados σ_k obtido pelo algoritmo 8 e $\gamma \in (0, 1)$.

Passo 1. Cálculo da direção de busca.

$$d_k = P_{\Omega}(x_k - \sigma_k g(x_k)) - x_k$$

Passo 2. Teste de decréscimo suficiente.

Calculamos $x_{k+1} = x_k + \alpha_k d_k$ e testamos:

$$f(x_{k+1}) \leq f(x_k) + \gamma \alpha_k g(x_k)^T d_k \quad (3.6)$$

Passo 2.1. $\alpha = 1$.

Passo 2.2. Se (3.6) foi aceito, fazer $\alpha_k = \alpha$. Senão reduzir α por um procedimento de interpolação quadrática e repetir o teste (3.6).

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

Observe que se x_{k+1} foi calculado com este algoritmo e $x_k \in F_I$, então $x_{k+1} \notin \bar{F}_I$.

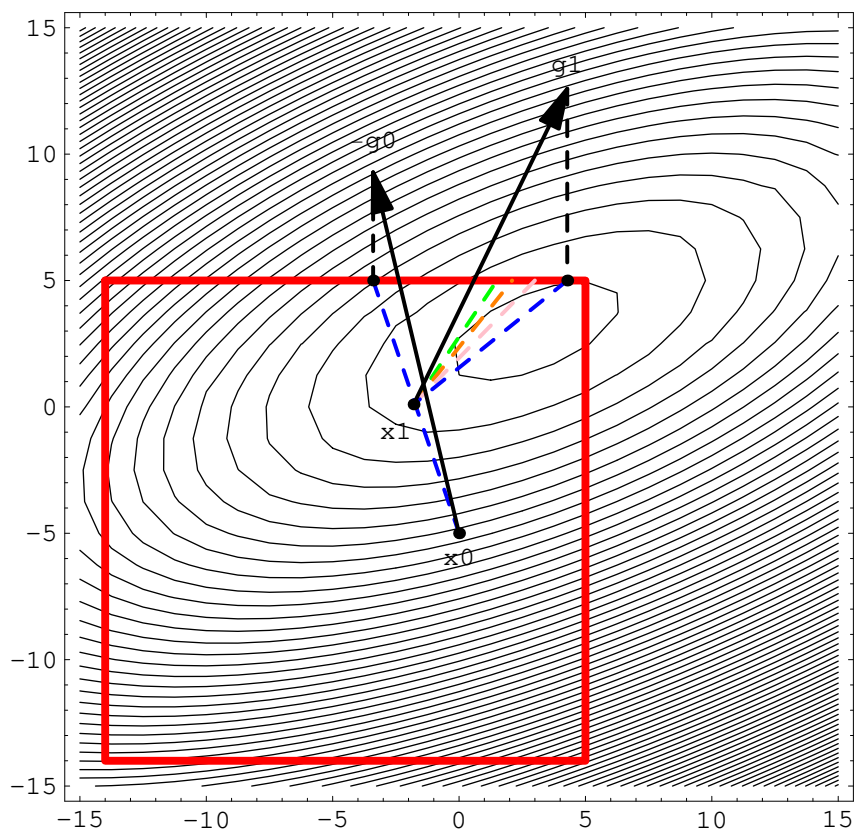


Figura 3

Para termos uma idéia melhor da direção do gradiente espectral projetado, observe a figura 3. Podemos ver que quando partimos de x_0 , fazemos uma iteração da técnica do gradiente projetado, ou seja, andamos na direção oposta ao gradiente em x_0 ($-g_0$), então projetamos $-g_0$ na caixa, minimizando a função objetivo no segmento que une x_0 e a projeção de $-g_0$, encontrando x_1 .

No próximo passo, partimos de x_1 e encontramos a direção oposta ao gradiente, g_1 ; então ao invés de projetarmos esta direção diretamente na caixa, calculamos o passo espectral ao longo de g_1 , para então projetarmos na caixa e encontrarmos o próximo iterando minimizando no segmento que liga x_1 a esta projeção.

O algoritmo a seguir será usado independentemente das faces, como se fosse um algoritmo para uma minimização sem restrições. É baseado em buscas lineares com condições de decréscimo suficiente e extrapolação. Se o critério de decréscimo suficiente não se cumpre, fazemos backtracking. Caso contrário, fazemos a extrapolação. As iterações se repetem até que $\|g(x_k)\| = 0$.

Aqui M é equivalente a \bar{F}_I , $P(x_k) = P_\Omega(x_k)$ e $N > 1$.

Algoritmo 10. *Busca Linear com Backtracking e Extrapolação*

Passo 1. Cálculo da direção de busca

1.1 Obter $d_k \in \mathbb{R}^n$ tal que

$$g(x_k)^T d_k \leq -\theta \|g(x_k)\| \|d_k\|. \quad (3.7)$$

Passo 2. Decisões de Busca Linear;

2.1 Computar $\alpha_{max} \leftarrow \max \{\alpha \geq 0 \mid [x_k, x_k + \alpha d_k] \subset M\}$ e assumir

$\alpha \leftarrow \min \{\alpha_{max}, 1\}$;

se $(\alpha_{max} > 1)$ **então**

| $x_k + d_k \in \text{Int}(M)$, então vá para 2.2.;

senão

| vá para 2.3;

fim se

2.2 Aqui temos $x_k + d_k \in \text{Int}(M)$

se

$$f(x_k + d_k) \leq f(x_k) + \gamma g(x_k)^T d_k, \quad (3.8)$$

então

se

$$d_k^T g(x_k + d_k) \geq \beta g(x_k)^T d_k, \quad (3.9)$$

então

| Tomar $\alpha_k = 1$, $x_{k+1} = x_k + d_k$ e vá para o passo 5.

senão

| vá para o passo 3 (extrapolação).

fim se

senão

| vá para o passo 4 (backtracking).

fim se

2.3 Aqui temos $x_k + d_k \notin \text{Int}(M)$;

se $f(x_k + \alpha_{max} d_k) < f(x_k)$ **então**

| Tome $\alpha_k \geq \alpha_{max}$ e $x_{k+1} = P(x_k + \alpha_k d_k) \in M - \text{Int}(M)$ (ou seja, pertence à fronteira de M) tal que $f(x_{k+1}) \leq f(x_k + \alpha_{max} d_k)$ e vá para o passo 5.

senão

| vá para o passo 4 (Backtracking), para que a condição de Armijo seja satisfeita.

fim se

Passo 3. Extrapolação

3.1 se $[\alpha < \alpha_{max} \text{ e } N\alpha > \alpha_{max}]$ **então**

| *assume* $\alpha_{teste} \leftarrow \alpha_{max}$

senão

| *assume* $\alpha_{teste} \leftarrow N\alpha$.

fim se

3.2 se $[\alpha \geq \alpha_{max} \text{ e } \|P(x_k + \alpha d_k)\|_\infty < \max\{\epsilon_{abs}, \epsilon_{rel}\|P(x_k + \alpha d_k)\|_\infty\}]$

então

| Tome $\alpha_k = \alpha$, $x_{k+1} = P(x_k + \alpha d_k)$ e termine a execução deste algoritmo.

senão

| **3.3**

fim se

se $f[P(x_k + \alpha_{teste} d_k)] \geq f[P(x_k + \alpha d_k)]$ **então**

| Tome $\alpha_k = \alpha$, $x_{k+1} = P(x_k + \alpha d_k)$ e vá para o passo 5.

senão

| *assume* $\alpha \leftarrow \alpha_{teste}$ e vá para 3.1.

fim se

Passo 4. Backtracking

4.1 Calcule $\alpha_{novo} \in [\alpha\sigma_1, \alpha\sigma_2]$ e *assume* $\alpha \leftarrow \alpha_{novo}$.

4.2 se $f(x_k + \alpha d_k) \leq f(x_k) + \gamma g(x_k)^T d_k$ **então**

| Tome $\alpha_k = \alpha$, $x_{k+1} = x_k + \alpha d_k$ e vá para o passo 5.

senão

| vá para 4.1.

fim se

Passo 5. Atualizações

se $\alpha_k \geq \alpha_{max}$ **então**

| pare o algoritmo.

senão

| Caso contrário *assume* $k \leftarrow k + 1$ e vá para o passo 1.

fim se

No passo 2.2 testamos a condição de decréscimo suficiente e a condição do tamanho da derivada direcional. Este algoritmo realiza buscas lineares ao longo de direções que satisfazem a condição do ângulo (3.7). Em geral, essa direção vai possuir alguma informação de segunda ordem, então o passo com $\alpha = 1$ deverá

ser inicialmente testado e aceito se as condições de decréscimo suficiente (3.8) e da derivada direcional (3.9) são satisfeitas. Se a condição de decréscimo suficiente for aceita, mas a de derivada direcional não, devemos tentar passos maiores ao longo da direção d_k (extrapolação). Estes passos provavelmente produzirão um maior decréscimo no valor da função objetivo. Se a condição de decréscimo suficiente não for aceita, então o backtracking pára quando a condição de Armijo no passo (4.2) for satisfeita.

No passo 2.3, como $f(x_k + \alpha_{max}d_k) < f(x_k)$, tentamos obter um menor valor para a função objetivo multiplicando o passo por um fator fixo e o projetamos na caixa. Na prática, esse ponto é obtido pela extrapolação (passo 3).

Na extrapolação, tentamos sucessivas projeções de $x_k + \alpha d_k$ na caixa, com valores crescentes de α . Se o ponto de entrada $x_k + \alpha d_k$ é interior, mas $x_k + N\alpha d_k$ não é, temos a certeza de que o ponto $x_k + \alpha_{max}d_k$ será testado em primeiro lugar. A extrapolação termina quando o decréscimo do valor da função objetivo não é mais alcançado, ou quando a distância entre a projeção de dois pontos de teste é insignificante.

A iteração termina no passo 5. Se o iterando correspondente x_{k+1} está na fronteira de M , o algoritmo pára, tendo encontrado um valor na fronteira, onde o valor da função objetivo decresceu com respeito a todos os anteriores. Se x_{k+1} está no interior de M , a execução do algoritmo continua, aumentando o número de iterações.

Capítulo 4

Pré-condicionadores

Sabemos que o método dos gradientes conjugados possui muitas vantagens, uma sendo sua fácil implementação: nenhum parâmetro precisa ser estimado e o método consiste basicamente de multiplicações de matrizes e vetores e de operações com vetores. Entretanto, o método pode se tornar muito lento, dependendo do resíduo inicial e da distribuição dos autovalores da matriz. Neste capítulo, mostraremos uma versão de pré-condicionamento para o método dos gradientes conjugados usado em GENCAN, com o objetivo de tentar contornar esse último problema, acelerando a convergência do método.

Estamos resolvendo o problema (3.1).

Para reduzir o número de condição da matriz Hessiana, e assim melhorar o rendimento da iteração, iremos utilizar um produto interno definido pela matriz de pré-condicionamento (C).

Se C é simétrica definida positiva (na prática uma aproximação para A), usaremos o produto interno definido naturalmente por C , ou seja,

$$\langle x, y \rangle = x^T C y,$$

e considerar o pseudo-resíduo $h_k = C^{-1}(b - Ax_k)$. Note que $C^{-1}A$ é simétrica

definida positiva com respeito a esse produto interno. Observamos que se $C \approx A \Rightarrow \text{cond}(C^{-1}A) \approx 1$ então o novo sistema linear tende a ser melhor condicionado que o original. Substituindo, então, A no algoritmo anteriormente proposto para o MGC, ficamos com:

$$\begin{aligned}
\alpha_k &= h_k^T C h_k / d_k^T A d_k, \\
x_{k+1} &= x_k + \alpha_k d_k, \\
h_{k+1} &= h_k + \alpha_k C^{-1} A d_k, \\
\beta_k &= h_{k+1} C h_{k+1} / h_k^T C h_k, \\
d_{k+1} &= -h_{k+1} + \beta_k d_k,
\end{aligned} \tag{4.1}$$

Teremos então o algoritmo a seguir:

Algoritmo 11. *MGC Pré-condicionado*

Considerar $x = x_0$, $r_0 = b - Ax_0$, $h_0 = C^{-1}r_0$, $d_0 = -h_0$, $\delta_0 = r_0^T h_0$, $k = 0$.

Se $\delta_0 \leq \epsilon$ parar. Caso contrário fazer:

Passo 1. $h_k = A d_k$;

Passo 2. $\alpha_k = \delta_0 / d_k^T h_k$;

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

Passo 4. $r_{k+1} = r_k + \alpha_k h_k$;

Passo 5. $h_{k+1} = C^{-1}r_{k+1}$;

Passo 6. $\delta_1 = r_{k+1}^T h_{k+1}$;

Passo 7. Se $\delta_1 \leq \epsilon$ parar;

Passo 8. $\beta_k = \delta_1 / \delta_0$;

Passo 9. $\delta_0 = \delta_1$;

Passo 10. $d_{k+1} = -h_{k+1} + \beta_k d_k$;

Passo 11. $k = k + 1$.

Sabemos que o Método dos Gradientes Conjugados resolve o sistema linear $Ax = b$, que é o equivalente a encontrar um minimizador de uma função quadrática. Podemos, no entanto, observar que é o equivalente a minimizar o funcional $f(x) = \frac{1}{2} \langle r, A^{-1}r \rangle$, com $r = b - Ax$. Agora, no método pré-condicionado, $f(x)$ toma a forma:

$$f(x) = \frac{1}{2} \langle h, (C^{-1}A)^{-1}h \rangle = \frac{1}{2} (C^{-1}r)^T C (C^{-1}A)^{-1} C^{-1}r = \frac{1}{2} r^T A^{-1}r$$

o que nos mostra que estamos minimizando o mesmo funcional, porém no subespaço de Krylov $\{r_0, AC^{-1}r_0, \dots, (AC^{-1})_k r_0\}$. Com o pré-condicionador apropriado, este subespaço de Krylov pode gerar vetores para minimizar $f(x)$ muito mais rápido que o subespaço não pré-condicionado.

Note também que:

$$d_k^T A d_j = d_k^T (C^{-1}A) d_j = 0, 0 \leq j < k.$$

4.1 Pré-condicionando GENCAN

4.1.1 QNCGNA

O pré-condicionamento proposto em [24] para o algoritmo interno do GENCAN trata de criar uma fórmula Quase-Newton adequada à forma da Hessiana de f ; lembrando que GENCAN se aplica, originalmente, à minimização de uma função f qualquer, restrita a uma caixa. Este pré-condicionamento recebe o nome de QNCGNA que significa **Q**uasi-Newton **C**orrection of a **G**auss-Newton **A**proximation of the Hessian matrix.

No nosso caso, onde f é uma função quadrática, o pré-condicionamento proposto em [24] se limita a aplicar uma correção do tipo BFGS ([26, 13]) à matriz A , com o objetivo de melhorar seu condicionamento e, assim, acelerar a convergência do método dos gradientes conjugados.

Descrevemos, a seguir, o pré-condicionamento proposto para GENCAN, adaptado ao caso quadrático, chamando de C_p a matriz condicionadora.

Definindo $s = x_k - x_{k-1}$ e $y = \nabla f(x_k) - \nabla f(x_{k-1})$, podemos obter uma aproximação barata e razoável da Hessiana da seguinte maneira:

Passo 1 Definimos:

$$A_+ = \sigma I.$$

σ é obtido seguindo a filosofia do Método do Gradiente Espectral, ou seja:

$$\sigma_{spc} = \operatorname{argmin} \|\sigma s - y\|^2,$$

o que implica:

$$\sigma_{spc} = \frac{y^T s}{s^T s}$$

e, por segurança, corrigimos:

$$\sigma = \max \{ \sigma_{min}, \min \{ \sigma_{max}, \sigma_{spc} \} \}.$$

Passo 2 Se $s^T y \leq 10^{-8} \|s\| \|y\|$, definimos $C_p = A_+$. Caso contrário, corrigimos A_+ de maneira a satisfazer a equação secante $C_p s = y$ mantendo a característica de C_p ser definida positiva. Como A_+ é definida positiva, é natural usarmos BFGS. Assim:

$$C_p = A_+ + \frac{yy^T}{s^T y} - \frac{A_+ s s^T A_+}{s^T A_+ s}$$

o que se torna:

$$C_p = \sigma I + \frac{yy^T}{s^T y} - \frac{\sigma s s^T}{s^T s}. \quad (4.2)$$

4.1.2 Outro Pré-condicionador para GENCAN

Um pré-condicionador simples e que exige pouco esforço computacional por iteração é o pré-condicionador diagonal. Nesta análise se poderia pensar em substituir a matriz C_p da seção anterior pela diagonal extraída de (4.2). Dessa forma temos:

$$C = \operatorname{diag} \left(\sigma I + \frac{yy^T}{s^T y} - \frac{\sigma s s^T}{s^T s} \right). \quad (4.3)$$

Observamos que C é definida positiva e pode se constituir num pré-condicionador computacionalmente barato.

Capítulo 5

Experimentos Numéricos

Utilizamos o programa GENCAN, que foi desenvolvido dentro do projeto TANGO (Trustable Algorithms for Nonlinear General Optimization), juntamente com outras rotinas para Otimização, em Fortran. Esse projeto foi realizado no Departamento de Matemática Aplicada (DMA) da UNICAMP e no Departamento de Ciências da Computação da USP, sob a coordenação do professor José Mario Martinez. Os softwares são livres para usos não comerciais, podendo ser modificados ou redistribuídos de acordo com as normas do GNU (General Public License), conforme publicado pela Fundação para o Software Livre (Free Software Foundation), e estão sob constante atualização.

O GENCAN é um código em Fortran que não utiliza operações com matrizes, baseado nos algoritmos descritos na seção anterior, sendo portanto muito útil para resolver problemas de grande porte com restrições em caixas, em um tempo razoável. Para maiores detalhes sobre o GENCAN, consultar [6, 24] e o site do projeto TANGO(<http://www.ime.usp.br/~egbirgin/tango/>).

5.1 Análise do Desempenho do GENCAN

Com o objetivo de analisar o desempenho do GENCAN, com e sem pré-condicionamento, trabalhamos com um conjunto de 39 problemas extraídos da coleção CUTE - **C**onstrained and **U**nconstrained **T**esting **E**nvironment [8]. Este conjunto se refere

a todos os problemas de minimização de quadráticas em caixas desta coleção, que é composta por centenas de testes. Os nossos testes foram executados em um Pentium D, CPU 3.4 GHz e 2 GB de RAM, e os resultados obtidos estão registrados nas tabelas C.1 e C.2, no apêndice C, cuja notação é descrita a seguir.

Dados analisados:

- **N** é o número de variáveis do problema.
- **T** é o tempo gasto pelo GENCAN, medido em segundos.
- **I** igual a 1 significa que GENCAN convergiu com a norma do sup do critério de parada do gradiente projetado contínuo, **I** igual a 0 significa o mesmo, mas com a norma Euclidiana e **I** igual a 6 significa que o GENCAN parou, mas não por um critério de parada bem sucedido ($SupN > 1E - 04$).
- **ITG** é o número de iterações do GENCAN.
- **ITGC** é o número de iterações do método dos gradientes conjugados.
- **F** é o valor da função objetivo na solução.
- **SupN** é a norma do sup do gradiente projetado da função na solução, definida por $\|g\| = \max_{t \in J} |g(t)|$, onde J é um intervalo fechado. Como tolerância usamos 1E-04.

A tabela 5.1 nos mostra a média aritmética dos resultados do GENCAN com e sem pré-condicionamento, lembrando que os testes foram realizados com o pré-condicionar QNCGNA. Os gráficos que estão na tabela 5.2 comparam o tempo e

o número de iterações dos gradientes conjugados nestes dois algoritmos. Podemos observar que o tempo gasto pelo GENCAN sem pré-condicionador é inferior ao gasto pelo GENCAN com QNCGNA, porém o número de iterações dos gradientes conjugados é relativamente pouco menor, o que nos leva a concluir que as iterações do pré-condicionador são caras. Das tabelas C1 e C2, observamos que o problema NCVXBQP3 não convergiu com o critério de parada estabelecido, quando usamos o GENCAN sem pré-condicionamento, o que não ocorreu quando pré-condicionamos o algoritmo. Vemos então que o pré-condicionamento tornou o método mais robusto.

Podemos observar também a coluna que se refere ao número total de iterações do GENCAN, nas tabelas C1 e C2. Se calcularmos a média aritmética de ITG, referente à tabela C1, não considerando o resultado do problema NCVXBQP3, obtemos o valor de 206,39. Os mesmos cálculos realizados com os resultados da tabela C2, incluindo agora os obtidos em NCVXBQP3, levam ao valor de 203,03. Percebemos que o uso desse pré-condicionamento diminui muito pouco o número total de iterações, que incluem as iterações do algoritmo de busca linear e do método do gradiente espectral.

5.2 LANCELOT, QUACAN e GENCAN

Como citamos na introdução deste trabalho, na década de 90, os algoritmos LANCELOT (**L**arge **A**nd **N**onlinear **C**onstrained **E**xtended **L**agrangian **O**ptimization **T**echniques) e QUACAN (**Q**uadráticas **C**analizadas) eram os mais aplicados para minimização de quadráticas em caixas. Ambos baseiam-se em estratégias de região de confiança, com o princípio de restrições ativas, usando o método de gradientes conjugados nas faces da caixa e uma direção baseada no gradiente projetado para deixar a face. Para mais detalhes sobre o Lancelot, ver [9, 10, 11] e sobre o QUACAN ver [16].

A partir dos resultados publicados em [16, 15], obtidos com os softwares LANCELOT e QUACAN, selecionamos os problemas que poderiam ser comparados com os testes que realizamos para o GENCAN. LANCELOT foi analisado sem e com pré-condicionamento. Neste caso o pré-condicionador usado é de banda pentadiagonal. Não há versão pré-condicionada de QUACAN.

Todos os problemas escolhidos são de grande porte. Todos os parâmetros usados nos testes são os padrões tanto do LANCELOT como do QUACAN, que estão em [11] e [16]. Ver os resultados nas tabelas 5.3, 5.4 e 5.5. A notação usada é a mesma das tabelas anteriores.

Estes algoritmos foram desenvolvidos em Fortran 77 e executados em uma estação SUN Ultra 1 Creator.

As tabelas 5.6 e 5.7 são um resumo do desempenho dos algoritmos, com a média aritmética dos resultados.

O GENCAN é o que realiza maior número de iterações do método dos gradientes conjugados, sem ou com pré-condicionamento. No entanto, é importante salientar que GENCAN aplica o método dos gradientes conjugados para obter uma direção de busca para o algoritmo de busca linear (algoritmo 10 - seção 3.2). Assim, as iterações de gradientes conjugados são, para GENCAN, iterações de um algoritmo interno, e o número de iterações de fato efetuadas pelo algoritmo está registrado, para cada problema, nas colunas “ITG” das tabelas C1 e C2.

Por outro lado, tanto LANCELOT como QUACAN minimizam a função objetivo quadrática na caixa, sujeita a uma região de confiança. Este processo de minimização é realizado pelo método de gradientes conjugados, seguido da atualização do raio da região de confiança. Assim, as iterações efetuadas por ambos os métodos são predominantemente de gradientes conjugados. Se analisarmos sob este ponto de vista, a média de iterações realizadas por GENCAN sem e com pré-condicionamento é muito menor que as observadas em LANCELOT e QUACAN.

Fazendo uma rápida comparação entre o LANCELOT com e sem pré-condicionador, usando os resultados das tabelas 5.6 e 5.7, observamos que tanto o tempo total de execução como o número de iterações do método dos gradientes conjugados diminuíram.

Os gráficos na tabela 5.8 mostram a comparação entre todos os algoritmos.

| Algoritmo | T | ITGC |
|-------------------|-------|----------|
| GENCAN | 52,18 | 86081,37 |
| GENCAN com QNCGNA | 70,65 | 86460,39 |

Tabela 5.1: Comparação GENCAN com e sem pré-condicionador

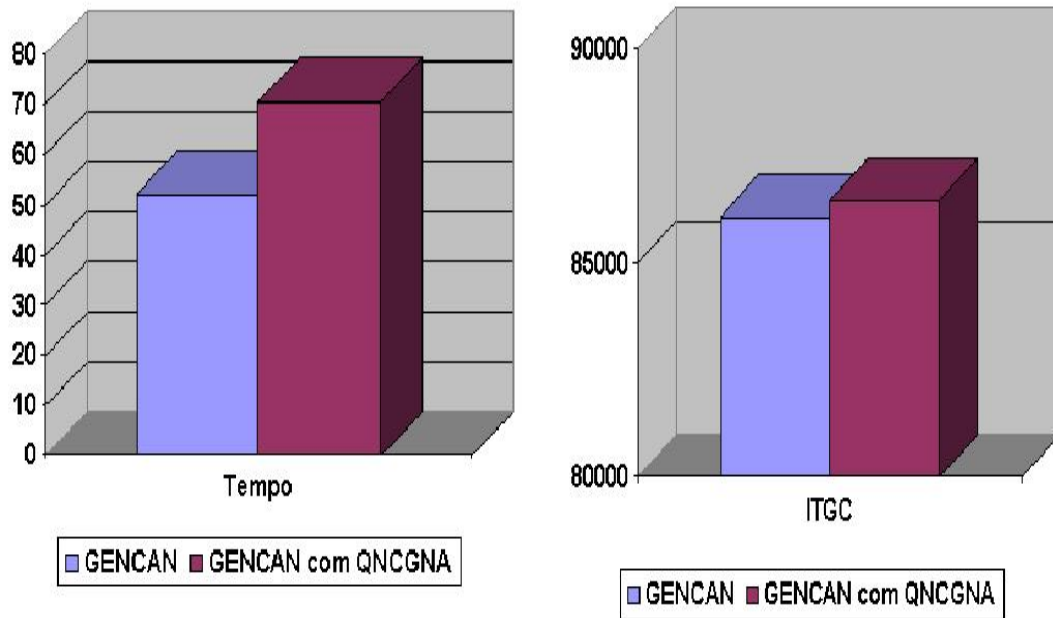


Tabela 5.2: Gráficos comparativos GENCAN com e sem pré-condicionador

| PROBLEMA | N | T | ITGC |
|----------|-------|---------|-------|
| BQPGAUSS | 2003 | 117,10 | 9511 |
| CVXBQP1 | 10000 | 1,20 | 1 |
| NCVXBQP1 | 10000 | 1,20 | 0 |
| NCVXBQP2 | 10000 | 3,70 | 435 |
| NCVXBQP3 | 10000 | 3,60 | 366 |
| ODNAMUR | 11130 | 1224,10 | 51556 |

Tabela 5.3: LANCELOT sem pré-condicionador

| PROBLEMA | N | T | ITGC |
|----------|-------|---------|-------|
| BQPGAUSS | 2003 | 46,40 | 2928 |
| CVXBQP1 | 10000 | 1,20 | 1 |
| NCVXBQP1 | 10000 | 1,20 | 0 |
| NCVXBQP2 | 10000 | 4,30 | 407 |
| NCVXBQP3 | 10000 | 4,10 | 359 |
| ODNAMUR | 11130 | 1377,70 | 30006 |

Tabela 5.4: LANCELOT com pré-condicionador

| PROBLEMA | N | T | ITGC |
|----------|-------|---------|-------|
| BQPGAUSS | 2003 | 84,62 | 6363 |
| CVXBQP1 | 10000 | 0,47 | 1 |
| NCVXBQP1 | 10000 | 0,34 | 1 |
| NCVXBQP2 | 10000 | 1,87 | 45 |
| NCVXBQP3 | 10000 | 2,26 | 59 |
| ODNAMUR | 11130 | 1540,64 | 37778 |

Tabela 5.5: QUACAN

| Algoritmo | T | ITGC |
|-----------|--------|----------|
| GENCAN | 28,53 | 15115,00 |
| LANCELOT | 269,46 | 12302,00 |
| QUACAN | 325,59 | 8837,60 |

Tabela 5.6: Comparação entre os algoritmos sem pré-condicionamento

| Algoritmo | T | ITGC |
|------------------------|--------|----------|
| GENCAN com QNCGNA | 31,37 | 14467,50 |
| LANCELOT com pré-cond. | 239,15 | 5616,83 |

Tabela 5.7: Comparação entre os algoritmos com pré-condicionamento

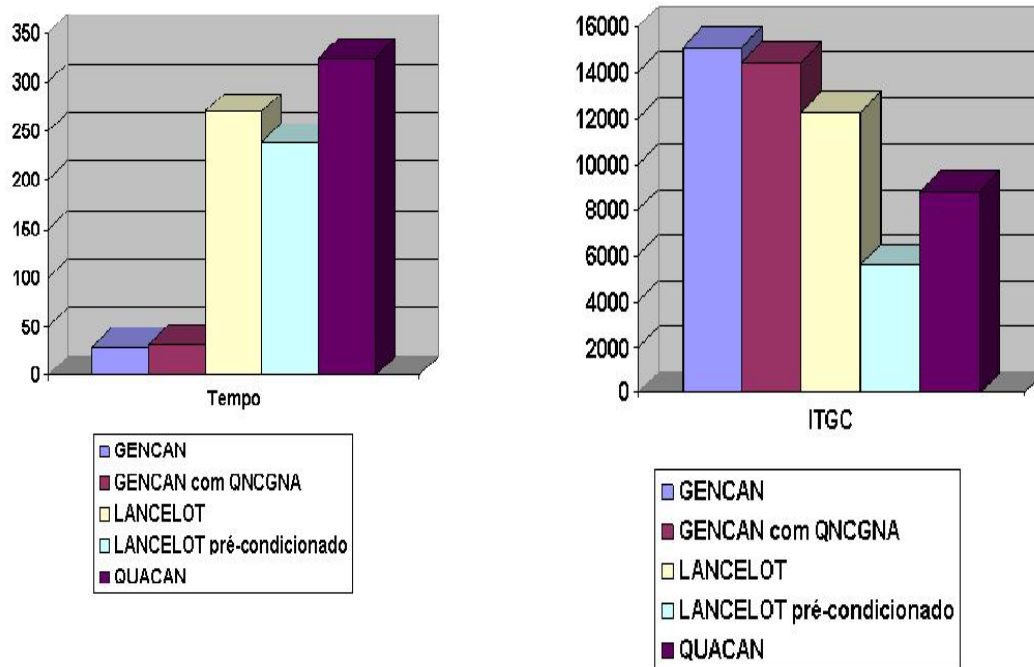


Tabela 5.8: Gráficos comparativos GENCAN, LANCELOT e QUACAN

Conclusão

Nosso objetivo neste trabalho foi o estudo com detalhes de métodos para minimização de quadráticas em caixas, pelo grande número de aplicações desse tipo de problema. Para isso estudamos também alguns métodos de minimização irrestrita e estratégias de globalização, como também procedimentos para minimização de quadráticas sem restrições. Estudamos a fundo métodos clássicos e métodos mais recentes.

Nosso trabalho culminou com a análise do desempenho do GENCAN, um método para minimização de funções gerais em caixas, que utiliza estratégias de busca linear e a direção do gradiente espectral projetado para deixar a face da caixa. Nosso objetivo foi analisar a performance de tal algoritmo quando aplicado a quadráticas. Realizamos um conjunto significativo de experimentos numéricos, comparando o GENCAN com sua versão pré-condicionada.

O pré-condicionamento é feito no método dos gradientes conjugados, utilizado na minimização da quadrática restrita à face ativa corrente, com o objetivo de contornar problemas de condicionamento da matriz Hessiana e assim acelerar a convergência do algoritmo. Percebemos que o GENCAN com pré-condicionamento é mais robusto do que sem pré-condicionamento, convergindo em todos os problemas testados. Além disso, em ambas versões, os problemas foram resolvidos em um tempo bastante razoável, verificando-se que o tempo gasto pelo GENCAN com QNCGNA é 31,74% superior ao gasto pelo GENCAN sem pré-condicionador, porém o número de iterações dos gradientes conjugados é somente 0,44 % maior, o que nos leva a concluir que as iterações do pré-condicionador são caras. Em vista disto, acreditamos que o tempo computacional poderia ser reduzido se fosse usado um pré-condicionador que tornasse mais simples o sistema linear a ser resolvido no método dos gradientes

conjugados.

Uma primeira idéia seria analisar o desempenho do GENCAN com um pré-condicionador diagonal, como descrito na seção 4.1.2. No entanto, inserir modificações em um software tão elaborado e computacionalmente eficiente como GENCAN não é uma tarefa simples. Uma alternativa seria produzir um novo código, baseado no algoritmo implementado em GENCAN, com diferentes pré-condicionadores para o método dos gradientes conjugados, e então realizar uma análise detalhada dos resultados obtidos.

Também fizemos uma pequena análise do desempenho do GENCAN comparado aos softwares LANCELOT e QUACAN, ambos baseados em estratégia de região de confiança, e podemos afirmar que GENCAN mostrou-se competitivo.

Foi muito produtiva e interessante a realização desta dissertação. Estudamos muitos métodos e nos concentramos em colocar os resultados mais relevantes e remeter o leitor para buscar informações adicionais em livros ou artigos específicos.

Agradecemos a oportunidade de realizar esta pesquisa que nos trouxe muito conhecimento na área.

Apêndice A

Notações

- $f \in C^p$ se refere à função f que pertence ao espaço de todas as funções p -continuamente diferenciáveis.
- $A > 0$ se refere a uma matriz definida positiva $A \in \mathbb{R}^{n \times n}$, ou seja, se $\langle Ax, x \rangle > 0, \forall x \neq 0, x \in \mathbb{R}^n$.
- $A \geq 0$ se refere a uma matriz semidefinida positiva $A \in \mathbb{R}^{n \times n}$, ou seja, se $\langle Ax, x \rangle \geq 0, x \in \mathbb{R}^n$.
- $A < 0$ se refere a uma matriz definida negativa, $A \in \mathbb{R}^{n \times n}$, ou seja, se $\langle Ax, x \rangle < 0, x \in \mathbb{R}^n$.
- $A \leq 0$ se refere a uma matriz semidefinida negativa, $A \in \mathbb{R}^{n \times n}$, ou seja, se $\langle Ax, x \rangle \leq 0, \forall x \neq 0, x \in \mathbb{R}^n$.
- $V = \text{span}\{u_0, u_1, \dots, u_k\}$ se refere ao espaço gerado pelos vetores linearmente independentes $\{u_0, u_1, \dots, u_k\}$.
- $R(A)$ se refere à imagem de A .
- $\alpha = O(\beta)$ se existe alguma constante positiva c tal que $\alpha \leq c\beta$.
- $o(g(x)) = f(x)$ ver apêndice B (B.2).

- $\|\cdot\|$ se refere à norma Euclidiana em C^n . Seja $x \in C^n$, $x = (x_1, x_2, \dots, x_n)^T$;
 $\|x\| = (\sum_{i=1}^n |x_i|^2)^{\frac{1}{2}}$
- P_Ω se refere à projeção ortogonal de um vetor v em Ω .
- $cond(A) = \|A^{-1}\| \|A\|$. Esse número é chamado de número de condição da matriz A . Observe que se A é auto adjunta então $\|A^{-1}\|_2 \|A\|_2 = \frac{\max \lambda(A)}{\min \lambda(A)}$, onde λ são os autovalores da matriz A .
- $Int(M)$ se refere ao interior de uma região considerada.

Apêndice B

Definições

Definição 1. *Derivada Direcional*

Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ continuamente diferenciável em um conjunto convexo $D \subset \mathbb{R}^n$. Então, para $x \in D$ e qualquer perturbação $d \in \mathbb{R}^n$, a derivada direcional de f em x na direção de d , definida por

$$\frac{\partial f(x)}{\partial d} = \lim_{\epsilon \rightarrow 0} \frac{f(x + \epsilon d) - f(x)}{\epsilon}, \quad (\text{B.1})$$

existe e é igual a $\nabla f(x)^T d$.

Definição 2. *Direção de descida*

$d \in \mathbb{R}^n$ é uma direção de descida de f a partir de x se, e somente se, existe $\bar{\alpha} > 0$ tal que $f(x + \alpha d) < f(x)$ para todo $\alpha \in (0, \bar{\alpha}]$.

Proposição 5. *Sejam $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $f \in C^1$, $x \in \mathbb{R}^n$ tal que $\nabla f(x) \neq 0$, $d \in \mathbb{R}^n$ tal que $\nabla f(x)^T d < 0$. Então d é direção de descida.*

Demonstração. Ver [18]. □

Definição 3. *Assuma que $g(x) \neq 0$ para todo $x \neq a$ em algum intervalo contendo a . A notação*

$$f(x) = o(g(x)) \quad \text{enquanto } x \rightarrow a \quad (\text{B.2})$$

significa que

$$\lim_{x \rightarrow a} \frac{f(x)}{g(x)} = 0$$

Definição 4. *Convexidade*

O termo convexo pode ser aplicado tanto a conjuntos quanto a funções.

$S \subset \mathbb{R}^n$ é um conjunto convexo se uma linha reta conectando quaisquer dois pontos em S se encontra totalmente no interior de S . Formalmente, para quaisquer dois pontos $x \in S$ e $y \in S$, nós temos $\alpha x + (1 - \alpha)y \in S \forall \alpha \in [0, 1]$.

f é uma função convexa se seu domínio é um conjunto convexo e se para quaisquer dois pontos x e y no domínio, o gráfico de f se encontra abaixo da reta que conecta $(x, f(x))$ a $(y, f(y))$ no espaço \mathbb{R}^{n+1} . Ou seja, nós temos

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y), \quad \forall \alpha \in [0, 1],$$

e estritamente convexa, quando

$$f(\alpha x + (1 - \alpha)y) < \alpha f(x) + (1 - \alpha)f(y), \quad \forall \alpha \in [0, 1],$$

Teorema 8. *Teorema de Taylor*

Suponha que $f: \mathbb{R}^n \rightarrow \mathbb{R}$ é continuamente diferenciável e que $d \in \mathbb{R}^n$. Então temos que:

$$f(x + d) = f(x) + \nabla f(x + td)d,$$

para algum $t \in (0, 1)$. Além disso, se f é duas vezes continuamente diferenciável, temos que

$$\nabla f(x + d) = \nabla f(x) + \int_0^1 \nabla^2 f(x + td)d \, dt, \quad (\text{B.3})$$

e que

$$f(x + d) = f(x) + \nabla f(x)^T d + \frac{1}{2} d^T \nabla^2 f(x + td)d,$$

para algum $t \in (0, 1)$.

Apêndice C

Tabelas

| Problema | N | T | I | ITG | ITGC | F | SupN |
|----------|-------|---------|---|------|---------|-------------|-----------|
| BQP1VAR | 1 | 0 | 0 | 1 | 1 | 0,0000E+00 | 0,00E+00 |
| BQPGABIM | 46 | 0 | 1 | 6 | 20 | -3,7903E-05 | 6,20E-07 |
| BQPGASIM | 50 | 0 | 1 | 5 | 20 | -5,5198E-05 | 8,300E-07 |
| BQPGAUSS | 2003 | 6,48 | 1 | 159 | 8486 | -3,6258E-01 | 3,10E-05 |
| BIGGSB1 | 5000 | 1745,04 | 1 | 5002 | 3123488 | 1,5000E-02 | 2,20E-16 |
| CVXBQP1 | 10000 | 0,02 | 0 | 1 | 1 | 2,2502E+06 | 0,00E+00 |
| CHENHARK | 5000 | 26,25 | 1 | 211 | 51739 | -2,0000E+00 | 8,00E-05 |
| HS3 | 2 | 0 | 1 | 1 | 1 | 3,2300E-05 | 3,60E-05 |
| JNLBRNG1 | 9604 | 10,35 | 1 | 90 | 2981 | -1,8057E-01 | 3,40E-05 |
| JNLBRNG2 | 9604 | 7,75 | 1 | 29 | 2575 | -4,1487E+00 | 8,40E-05 |
| JNLBRNGA | 9604 | 4,41 | 1 | 44 | 1323 | -2,7110E-01 | 6,80E-05 |
| JNLBRNGB | 9604 | 13,31 | 1 | 64 | 4583 | -6,3007E+00 | 7,80E-07 |
| NCVXBQP1 | 10000 | 0,03 | 0 | 1 | 1 | -1,9855E+10 | 0,00E+00 |
| NCVXBQP2 | 10000 | 0,27 | 1 | 26 | 81 | -1,3340E+10 | 4,90E-09 |
| NCVXBQP3 | 10000 | 213,32 | 6 | 7449 | 98702 | -6,5575E+09 | 5,30E-03 |
| NOBNDTOR | 5184 | 3,07 | 1 | 47 | 1283 | -4,4993E-01 | 5,90E-05 |
| OBSTCLAE | 9604 | 6,37 | 1 | 115 | 1563 | 1,8865E+00 | 1,000E-09 |
| OBSTCLAL | 9604 | 3,58 | 1 | 43 | 1112 | 1,8865E+00 | 9,40E-06 |
| OBSTCLBL | 9604 | 6,45 | 1 | 235 | 584 | 7,2722E+00 | 8,50E-06 |
| OBSTCLBM | 9604 | 2,13 | 1 | 52 | 311 | 7,2722E+00 | 3,20E-09 |
| OBSTCLBU | 9604 | 3,62 | 1 | 102 | 546 | 7,2722E+00 | 4,40E-05 |
| ODNAMUR | 11130 | 135,87 | 1 | 1312 | 67006 | 9,2366E+03 | 2,20E-05 |
| OSLBQP | 8 | 0 | 0 | 1 | 1 | 6,2500E+00 | 0,00E+00 |
| PENTDI | 5000 | 0 | 1 | 1 | 0 | -7,5000E-01 | 2,70E-10 |
| QUDLIN | 5000 | 0 | 0 | 1 | 0 | -1,2500E+09 | 0,00E+00 |
| SIM2BQP | 1 | 0 | 0 | 1 | 0 | 0,0000E+00 | 0,00E+00 |
| SIMBQP | 2 | 0 | 1 | 4 | 3 | 6,0185E-35 | 3,50E-17 |
| TORSION1 | 5184 | 1,55 | 1 | 45 | 716 | -4,3028E-01 | 4,10E-05 |
| TORSION2 | 5184 | 0,89 | 1 | 32 | 357 | -4,3028E-01 | 2,00E-05 |
| TORSION3 | 5184 | 0,52 | 1 | 22 | 243 | -1,2170E+00 | 5,50E-09 |
| TORSION4 | 5184 | 0,36 | 1 | 21 | 146 | -1,2170E+00 | 8,30E-07 |
| TORSION5 | 5184 | 0,19 | 1 | 12 | 97 | -2,8634E+00 | 1,40E-09 |
| TORSION6 | 5184 | 0,15 | 1 | 10 | 60 | -2,8634E+00 | 4,00E-09 |
| TORSIONA | 5184 | 1,74 | 1 | 46 | 721 | -4,1830E-01 | 3,90E-05 |
| TORSIONB | 5184 | 1,23 | 1 | 35 | 474 | -4,1830E-01 | 9,20E-10 |
| TORSIONC | 5184 | 0,54 | 1 | 22 | 242 | -1,2042E+00 | 5,50E-09 |
| TORSIOND | 5184 | 0,45 | 1 | 22 | 172 | -1,2042E+00 | 6,40E-08 |
| TORSIONE | 5184 | 0,23 | 1 | 12 | 96 | -2,8502E+00 | 1,40E-09 |
| TORSIONF | 5184 | 0,17 | 1 | 10 | 59 | -2,8502E+00 | 4,40E-09 |

Tabela C.1: GENCAN sem pré-condicionadores

| PROBLEMA | N | T | I | ITG | ITGC | F | SupN |
|----------|-------|---------|---|------|---------|-------------|----------|
| BQP1VAR | 1 | 0,00 | 0 | 1 | 1 | 0,0000E+00 | 0,00E+00 |
| BQPGABIM | 46 | 0,00 | 1 | 6 | 20 | -3,7903E-05 | 6,20E-07 |
| BQPGASIM | 50 | 0,00 | 1 | 5 | 20 | -5,5198E-05 | 8,30E-07 |
| BQPGAUSS | 2003 | 10,84 | 1 | 172 | 12855 | -3,6258E-01 | 9,60E-07 |
| BIGGSB1 | 5000 | 2384,43 | 1 | 5002 | 3123488 | 1,5000E-02 | 2,20E-16 |
| CVXBQP1 | 10000 | 0,03 | 0 | 1 | 1 | 2,2502E+06 | 0,00E+00 |
| CHENHARK | 5000 | 36,28 | 1 | 231 | 54970 | -2,0000E+00 | 7,50E-05 |
| HS3 | 2 | 0,00 | 1 | 1 | 1 | 3,2350E-05 | 3,60E-05 |
| JNLBRNG1 | 9604 | 11,42 | 1 | 78 | 2874 | -1,8057E-01 | 3,40E-05 |
| JNLBRNG2 | 9604 | 8,47 | 1 | 29 | 2575 | -4,1487E+00 | 8,40E-05 |
| JNLBRNGA | 9604 | 5,16 | 1 | 46 | 1396 | -2,7110E-01 | 4,30E-09 |
| JNLBRNGB | 9604 | 15,00 | 1 | 65 | 4602 | -6,3007E+00 | 1,00E-07 |
| NCVXBQP1 | 10000 | 0,03 | 0 | 1 | 1 | -1,9855E+10 | 0,00E+00 |
| NCVXBQP2 | 10000 | 0,27 | 1 | 27 | 68 | -1,3340E+10 | 6,10E-05 |
| NCVXBQP3 | 10000 | 0,26 | 1 | 24 | 59 | -6,5575E+09 | 5,50E-06 |
| NOBNDTOR | 5184 | 3,12 | 1 | 48 | 1280 | -4,4993E-01 | 5,60E-05 |
| OBSTCLAE | 9604 | 7,26 | 1 | 119 | 1654 | 1,8865E+00 | 1,20E-09 |
| OBSTCLAL | 9604 | 3,80 | 1 | 43 | 1112 | 1,8865E+00 | 9,40E-06 |
| OBSTCLBL | 9604 | 6,89 | 1 | 247 | 574 | 7,2722E+00 | 2,30E-09 |
| OBSTCLBM | 9604 | 2,78 | 1 | 71 | 362 | 7,2722E+00 | 3,10E-09 |
| OBSTCLBU | 9604 | 3,72 | 1 | 100 | 514 | 7,2722E+00 | 4,40E-05 |
| ODNAMUR | 11130 | 176,79 | 1 | 1305 | 73821 | 9,2366E+03 | 1,00E-05 |
| OSLBQP | 8 | 0,00 | 1 | 1 | 1 | 6,2500E+00 | 0,00E+00 |
| PENTDI | 5000 | 0,00 | 1 | 1 | 0 | -7,5000E-01 | 2,70E-10 |
| QUDLIN | 5000 | 0,00 | 0 | 1 | 0 | -1,2500E+09 | 0,00E+00 |
| SIM2BQP | 1 | 0,00 | 0 | 1 | 0 | 0,0000E+00 | 0,00E+00 |
| SIMBQP | 2 | 0,00 | 1 | 4 | 3 | 6,0185E-35 | 3,60E-17 |
| TORSION1 | 5184 | 1,71 | 1 | 45 | 716 | -4,3028E-01 | 4,10E-05 |
| TORSION2 | 5184 | 1,05 | 1 | 32 | 359 | -4,3028E-01 | 2,00E-05 |
| TORSION3 | 5184 | 0,55 | 1 | 22 | 243 | -1,2170E+00 | 5,50E-09 |
| TORSION4 | 5184 | 0,39 | 1 | 21 | 143 | -1,2170E+00 | 8,00E-07 |
| TORSION5 | 5184 | 0,21 | 1 | 12 | 97 | -2,8634E+00 | 1,40E-09 |
| TORSION6 | 5184 | 0,16 | 1 | 10 | 60 | -2,8634E+00 | 4,00E-09 |
| TORSIONA | 5184 | 1,85 | 1 | 46 | 721 | -4,1830E-01 | 3,90E-05 |
| TORSIONB | 5184 | 1,15 | 1 | 34 | 393 | -4,1830E-01 | 8,50E-05 |
| TORSIONC | 5184 | 0,57 | 1 | 22 | 242 | -1,2042E+00 | 5,50E-09 |
| TORSIOND | 5184 | 0,51 | 1 | 22 | 173 | -1,2042E+00 | 6,80E-08 |
| TORSIONE | 5184 | 0,24 | 1 | 12 | 96 | -2,8502E+00 | 1,40E-09 |
| TORSIONF | 5184 | 0,17 | 1 | 10 | 59 | -2,8502E+00 | 4,40E-09 |

Tabela C.2: GENCAN com QNCGNA

Bibliografia

- [1] R. ANDREANI, E. G. BIRGIN, J. M. MARTÍNEZ & M. L. SCHUVERDT, *On Augmented Lagrangian methods with general lower-level constraints* Technical Report MCDO-050304, DMA, UNICAMP, Brazil, 2005.
- [2] R. ANDREANI, E. G. BIRGIN, J. M. MARTÍNEZ & M. L. SCHUVERDT, *Augmented Lagrangian methods under the constant positive linear dependence constraint qualification*, Mathematical Programming 112, pp. 5-32, (2008).
- [3] O. AXELSSON, *Iterative Solution Methods*, Cambridge University Press, Cambridge, 1994.
- [4] J. BARZILAI & J. M. BORWEIN, *Two-Point Step Size Gradient Methods*, IMA Journal of Numerical Analysis, 8, pp. 141-148, 1988.
- [5] R. H. BIELSCHOWSKY, A. FRIEDLANDER, F.M. GOMES, J.M. MARTINEZ, & M. RAYDAN, *An adaptive algorithm for bound constrained quadratic minimization*, Investigación Operativa, 7, pp. 67-102, 1997.
- [6] E. G. BIRGIN & J. M. MARTINEZ, *Large-scale active-set box-constrained optimization method with spectral projected gradients*, Computacional Optimization and Applications 23, pp. 101-125, 2002.
- [7] E. G. BIRGIN, J. M. MARTÍNEZ & M. RAYDAN, *Nonmonotone spectral projected gradient methods on convex sets*, SIAM Journal on Optimization 10, pp. 1196-1211, 2000.

- [8] I. BONGARTZ , A. R. CONN, N. I. M. GOULD & PH. L. TOINT, *CUTE: Constrained and Unconstrained Testing Environment*, ACM Transactions on Mathematical Software, 21, pp. 123 - 160, 1995.
- [9] A. R. CONN, N. I. M. GOULD & PH. L. TOINT, *A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds*, SIAM Journal on Numerical Analysis, 28, pp. 545-572, 1991.
- [10] A. R. CONN, N. I. M. GOULD & PH. L. TOINT, *On the number of inner iterations per outer iteration of a globally convergent algorithm for optimization with general nonlinear equality constraints and simple bounds*, em D.F Griffiths e G.A. Watson, editores, 'Proceedings of the 14th Biennial Numerical Analysis Conference Dundee 1991', Longmans, 1992.
- [11] A. R. CONN, N. I. M. GOULD & PH. L. TOINT, *LANCELOT - A Fortran Package for Large-Scale Nonlinear Optmization (Release A)*, Springer Verlag, Berlin, Heidelberg, New York, 1992.
- [12] C. CUNHA, *Métodos numéricos para as engenharias e ciências aplicadas*, Editora da Unicamp, 1993.
- [13] J. E. DENNIS, JR. & R. B. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, 1996.
- [14] M. A, DINIZ-EHRHARDT, M. A. GOMES-RUGGIERO, J. M. MARTINEZ & S. A. SANTOS, *Augmented Lagrangian algorithms based on the spectral projected gradient for solving nonlinear programming problems*, Journal of Optimization Theory and Applications, 123, n^o 3, pp. 497-517, 2004.
- [15] M. A, DINIZ-EHRHARDT, Z. DOSTÁL, M. A. GOMES-RUGGIERO & J. M. MARTINEZ, *Nonmonotone strategy for minimization of quadratics with simple constraints*, Aplications of Mathematics, 46, n^o 5, pp. 321-338, 2001.
- [16] M. A, DINIZ-EHRHARDT, M. A. GOMES-RUGGIERO, & S. A. SANTOS, *Numerical Analysis of leaving-face parameters in bound-constrained quadratic minimization*, Optimization Methods and Softares, 15, pp. 45-66, 2001.

- [17] M. A. DINIZ-EHRHARDT & P. C. FEIJÃO, *O desempenho de diferentes pré-condicionadores para o método de gradientes conjugados aplicado à minimização em caixas*, Relatório de Pesquisa, Agosto, 1997.
- [18] A. FRIEDLANDER, *Elementos de Programação Não-Linear*, Editora da Unicamp, 1994.
- [19] A. FRIEDLANDER, J. M. MARTINEZ & M. RAYDAN, *A new method for large-scale box constrained convex quadratic minimization problems*, Optim. Methods and Softwares, 5, pp.57-74, 1995.
- [20] M. A. GOMES-RUGGIERO & V. L. R. LOPES, *Cálculo Numérico - Aspectos Teóricos e Computacionais*, segunda edição, Makron Books, 1997.
- [21] M. R. HESTENES & E. STEIFEL, *Methods of conjugated gradient for solving linear systems*, J. of Res. Nat. Bureau Standards, 49, pp. 409-436, 1952.
- [22] C. T. KELLEY, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia, 1995.
- [23] D. G. LUENBERGER, *Linear and Nonlinear Programming*, 2 edição, Nova York, Addison - Wesley Publishing Company, 1986.
- [24] J. M. MARTINEZ, *Otimização Prática Usando o Lagrangiano Aumentado*, Opúsculo, DMA - IMECC - UNICAMP, 2006.
- [25] J. M. MARTINEZ & S. A. SANTOS, *Métodos Computacionais de Otimização*, 1 ed. Rio de Janeiro: Sociedade Brasileira de Matemática, DMA, UNICAMP, 1995.
- [26] J. NOCEDAL & S. J. WRIGHT, *Numerical Optimization*, Springer-Verlag New York, Inc., 1999.
- [27] L. G. PEDROSO, *Sobre o desempenho de métodos de busca direta para minimização irrestrita*, Dissertação de Mestrado, 2005.
- [28] L. G. PEDROSO E M. A. DINIZ-EHRARDT, *Um método iterativo para minimização de quadráticas em caixas*, Relatório de Pesquisa, RP 05/03, 2005.

- [29] M. RAYDAN, *On the Barzilai and Borwein choice of steplength for the gradient method*, IMA Journal of Numerical Analysis, 13, pp. 321-326, 1993.
- [30] M. RAYDAN, *The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem*, SIAM Journal of Optimization, 7, n^o 1, pp. 26-33, 1997.