

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE MATEMÁTICA ESTATÍSTICA E COMPUTAÇÃO CIENTÍFICA
DEPARTAMENTO DE MATEMÁTICA APLICADA

**OTIMIZAÇÃO IRRESTRITA SEM DERIVADAS BASEADA EM
INTERPOLAÇÃO POLINOMIAL**

Thiago Rincão

Mestrado em Matemática Aplicada - Campinas - SP

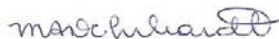
Orientadora: Prof. Dra. Maria Aparecida Diniz Ehrhardt

DMA - IMECC - UNICAMP

OTIMIZAÇÃO IRRESTRITA SEM DERIVADAS BASEADA EM INTERPOLAÇÃO POLINOMIAL

Este exemplar corresponde à redação final da dissertação devidamente corrigida e defendida por Thiago Rincão e aprovada pela comissão julgadora.

Campinas, 23 de Julho de 2008.



Profa. Dra. Maria Aparecida Diniz Ehrhardt
Orientadora

Banca Examinadora:

Profa. Dra. Maria Aparecida Diniz Ehrhardt (IMECC/UNICAMP)
Prof. Dr. Aurélio Ribeiro Leite de Oliveira (IMECC/UNICAMP)
Prof. Dr. José Marcos Lopes (Departamento de Matemática/UNESP Ilha Solteira)

Dissertação apresentada ao Instituto de Matemática, Estatística e Computação Científica, UNICAMP, como requisito parcial para obtenção do Título de MESTRE em MATEMÁTICA APLICADA E COMPUTACIONAL.

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

Rincão, Thiago

R471o Otimização irrestrita sem derivadas baseada em interpolação
polinomial / Thiago Rincão -- Campinas, [S.P. :s.n.], 2008.

Orientadora : Maria Aparecida Diniz Ehrhardt

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

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

Título em inglês: Derivative-free unconstrained optimization based on polynomial
interpolation.

Palavras-chave em inglês (Keywords): 1. Nonlinear programming. 2. Unconstrained
optimization. 3. Derivative-free methods.

Área de concentração: Otimização

Titulação: Mestre em Matemática Aplicada

Banca examinadora:

Profa. Dra. Maria Aparecida Diniz Ehrhardt (IMECC/UNICAMP)

Prof. Dr. Aurélio Ribeiro Leite de Oliveira (IMECC/UNICAMP)

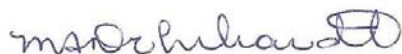
Prof. Dr. José Marcos Lopes (Departamento de Matemática/UNESP Ilha Solteira)

Data da defesa: 23/07/2008

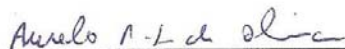
Programa de pós-graduação: Mestrado em Matemática Aplicada

Dissertação de Mestrado defendida em 23 de julho de 2008 e aprovada

Pela Banca Examinadora composta pelos Profs. Drs.



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



Prof. (a). Dr (a). AURELIO RIBEIRO LEITE DE OLIVEIRA



Prof. (a). Dr (a). JOSÉ MARCOS LOPES

À minha família.

Agradecimentos

Agradeço:

Aos meus pais, Jurandir e Maria de Fátima, e ao meu irmão Thadeu, por todo carinho e amor a mim dispensado, e pelo incomensurável apoio durante todo o processo de desenvolvimento desse trabalho.

À minha tia Cida, por me hospedar e agüentar meu instável humor nos momentos difíceis, sempre me apoiando em todos momentos.

À minha orientadora Cheti, pela paciência, agilidade, eficiência e inúmeros atendimentos.

Aos professores Aurélio, Vera e José Marcos, pela participação nas bancas de qualificação e/ou defesa. A todos os professores que passaram por minha vida, os quais direta ou indiretamente contribuíram para realização desse trabalho.

A todos meus amigos, pela motivação e por momentos de descontração, me fazendo esquecer das obrigações, e permitindo a elas retornar com muito mais energia.

Edson, por em mim aguçar o interesse em desenvolver esse trabalho nesta universidade, e ao meu primo Mauro pelas dicas.

Powell, por disponibilizar seus papers e softwares utilizados nessa dissertação.

A todos que de alguma forma tenham me ajudado e que por ventura eu tenha esquecido de agradecer.

Finalmente a Deus, por meu alicerce.

Que eu possa corresponder às expectativas que acompanham o título de mestre.

Resumo

Neste trabalho, tratamos de problemas de minimização irrestrita. Estudamos as condições de otimalidade para este tipo de problema, bem como os métodos clássicos para sua resolução, tais como: o método do Gradiente, de Newton e os Quase-Newton. Abordamos também procedimentos de busca linear e de região de confiança, conhecidos como estratégias de globalização. No entanto, nosso principal interesse está voltado a métodos de minimização que não fazem uso das derivadas da função objetivo. Neste sentido, enfocamos um método de minimização irrestrita, sem derivadas, baseado em interpolação quadrática, proposto por M. J. D. Powell, que está implementado no *software* NEWOUA. Com o objetivo de avaliar o desempenho computacional do algoritmo realizamos vários experimentos numéricos.

Abstract

In this work, we are interested in unconstrained minimization problems. We study the optimality conditions for this kind of problem, and the classical methods for its resolution, such as: the Gradient method, Newton and Quasi-Newton methods. We also consider line search and trust region techniques, which are known as globalization strategies. However, our main interest is the study of derivative-free minimization methods. In this sense, we consider a derivative-free unconstrained minimization approach, based on quadratic interpolation, proposed by M. J. D. Powell, which is implemented in the software NEWUOA. In order to analyze the performance of the algorithm, we present several numerical experiments.

Notação

- $f \in C^p$ - diz-se que f é de classe C^p , se tiver derivadas parciais contínuas até a ordem p .
- $\nabla f_k = \nabla f(x_k)$.
- $\nabla^2 f_k = \nabla^2 f(x_k)$.
- $\|A\|_F = \left\{ \sum_{i=1}^n \sum_{j=1}^n a_{ij}^2 \right\}^{\frac{1}{2}}$, $A \in \mathbb{R}^{n \times n}$.
- $\eta_k = o(v_k)$ se $\lim_{k \rightarrow \infty} \frac{\eta_k}{v_k} = 0$.

Sumário

1	Introdução	1
2	Minimização sem Restrições: Conceitos Fundamentais e Métodos de Resolução	3
2.1	Introdução	3
2.2	Condições de Otimalidade	4
2.3	Método do Gradiente	6
2.4	Método de Newton	8
2.5	Métodos Quase-Newton	11
2.5.1	A atualização secante simétrica de Powell e Broyden	13
3	Estratégias de Globalização	16
3.1	Introdução	16
3.2	Métodos de Busca Linear	16
3.3	Métodos de Região de Confiança	19
3.3.1	Esboço do algoritmo	21
3.3.2	O Ponto de Cauchy	23
3.3.3	O Método de <i>dog-leg</i>	24
3.3.4	Aproximação de Steihaug	28
3.3.5	Redução obtida pelo ponto de Cauchy	30
3.3.6	Convergência para pontos estacionários	32
4	Um Método sem Derivadas Baseado em Interpolação Quadrática	34
4.1	O Modelo Quadrático e o Cálculo dos Pontos Iniciais	34
4.2	O Subproblema de Região de Confiança	37

5	Experimentos Numéricos	39
5.1	Problemas de Moré, Garbow e Hillstrom	39
5.2	Problema SPHRPTS	41
6	Conclusões	47

Lista de Figuras

3.1	Condição de Armijo.	17
3.2	Condição do ângulo	18
3.3	Trajetória exata e aproximação Dogleg.	26
5.1	Comparação do número de avaliações de função para diferentes valores de m das tabelas (5.2) e (5.3)	42
5.2	Comparação do tempo computacional gasto para os diferentes valores de m das tabelas (5.2) e (5.3)	43
5.3	Comparação do número de avaliações de função entre NEWUOA e GENCAN	46
5.4	Comparação do tempo computacional gasto entre NEWUOA e GENCAN	46

Lista de Tabelas

5.1	Resultados da aplicação do método em problemas de [12]	. . .	40
5.2	Resultados para o problema SPHRPTS com $m = 2n + 1$	. . .	42
5.3	Resultados para o problema SPHRPTS com $m = \frac{1}{2}(n+1)(n+2)$		43
5.4	Resultados para o problema SPHRPTS - GENCAN		45

Capítulo 1

Introdução

Neste trabalho, consideraremos o problema irrestrito

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

onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Ou seja, buscamos um ponto $x^* \in \mathbb{R}^n$ que soluciona o problema (1.1). Tal vetor é denominado minimizador de $f(x)$.

Dentre os diversos métodos para solucionar este tipo de problema, os mais usuais são baseados no cálculo de derivadas de $f(x)$. Citemos, por exemplo, o método do Gradiente, o método de Newton e os métodos Quase-Newton. Apesar da indiscutível eficiência e eficácia de tais métodos, nas últimas décadas uma parte da comunidade científica passou a trabalhar em métodos que não usam Hessiana ou mesmo gradiente da função f , estimulados pela dificuldade ou impossibilidade de tais cálculos; atualmente esses métodos são conhecidos como métodos sem derivadas ou “derivative-free”. Os métodos sem derivadas são muito atrativos, pois, além de apresentarem, em geral, bom desempenho, são de fácil utilização, já que é preciso apenas que o usuário forneça a função a ser minimizada.

Hoje em dia, existem vários métodos sem derivadas que tentam encontrar uma solução para os problemas do tipo (1.1), como os de busca direta e os que aproximam as derivadas por diferenças finitas. Longe de querer competir com os métodos clássicos de minimização, os algoritmos de busca direta são de crescente interesse por representarem uma alternativa em ocasiões onde métodos mais elaborados falham. O método de Nelder-Mead [13] está entre

os mais utilizados. Essa popularidade reside nos bons resultados práticos, especialmente se levada em conta a simplicidade do algoritmo. A partir de um simplex no $\mathbb{R}^n$, a idéia do método é, a cada iteração, substituir o vértice com o valor menos desejado da função objetivo ou, quando isso não for possível, reduzir as dimensões do simplex. Além dos métodos de busca direta temos os que simplesmente substituem o cálculo das derivadas, por aproximações como diferenças finitas.

Entre todos os métodos que não usam derivadas, pesquisamos, para este trabalho, uma proposta de M. J. D. Powell, cujas idéias foram introduzidas em [16] e aprimoradas, gerando o software NEWUOA [17, 18, 19]. Trata-se de um método para problemas de minimização irrestrita, baseado em aproximações quadráticas da função objetivo. Tais aproximações são altamente úteis quando se deseja obter uma taxa rápida de convergência. Por outro lado, cada modelo quadrático tem $\frac{1}{2}(n+1)(n+2)$ parâmetros independentes a serem determinados, onde n é a dimensão do problema. Esta quantidade de cálculo é proibitivamente cara em muitas aplicações com n grande. O método tenta, então, construir modelos quadráticos satisfatórios com menos dados. No começo de cada iteração, o modelo quadrático tem que satisfazer somente m condições de interpolação, onde m pode variar entre $2n+1$ e $\frac{1}{2}(n+1)(n+2)$.

O sucesso desse método se deve à técnica usada para a atualização do modelo quadrático, baseada na minimização da norma de Frobenius da diferença das Hessianas do modelo quadrático velho e novo. A cada iteração devemos obter um novo ponto, o qual ocupará a posição de um dos pontos de interpolação. Este ponto é obtido a partir de uma estratégia de região de confiança.

Este trabalho está organizado da seguinte forma. No capítulo 2, após uma pequena revisão de elementos importantes de otimização não linear, apresentaremos o método do Gradiente, o método de Newton e métodos do tipo Quase-Newton, usuais para minimização irrestrita com derivadas. No Capítulo 3, trataremos de estratégias de globalização, como busca linear e algumas técnicas de região de confiança. No quarto Capítulo descreveremos o método proposto por Powell e no Capítulo 5, mostraremos alguns experimentos numéricos, realizando uma análise dos resultados obtidos. Apresentaremos nossas conclusões no Capítulo final.

Capítulo 2

Minimização sem Restrições: Conceitos Fundamentais e Métodos de Resolução

2.1 Introdução

Neste capítulo apresentaremos conceitos fundamentais que formam a base deste estudo em otimização, e os métodos mais utilizados e conhecidos para minimização sem restrições, com uso de derivadas.

Consideremos o problema de minimização irrestrita da forma

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

onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Ou seja, buscamos um ponto $x^* \in \mathbb{R}^n$ que soluciona o problema (2.1). Tal vetor é denominado minimizador de $f(x)$. Dois tipos de minimizadores podem ser considerados: local e global.

Definição 2.1.1. Um ponto $x^* \in \mathbb{R}^n$ é um minimizador local de f se e somente se existe $\epsilon > 0$ tal que $f(x) \geq f(x^*)$ para todo $x \in \mathbb{R}^n$ tal que $\|x - x^*\| < \epsilon$.

Definição 2.1.2. Um ponto $x^* \in \mathbb{R}^n$ é um minimizador global de f se e somente se $f(x) \geq f(x^*)$, $\forall x \in \mathbb{R}^n$.

Consequência direta das definições de minimizadores local e global é o fato de que todo minimizador global é também minimizador local.

Passamos agora a outras definições e resultados importantes.

Definição 2.1.3. Definimos a derivada direcional de f em x na direção de p , por $D_p f(x) \equiv \lim_{\epsilon \rightarrow 0} \frac{f(x + \epsilon p) - f(x)}{\epsilon}$. Se f é diferenciável, então $D_p f(x) = \nabla f(x)^T p$.

Definição 2.1.4. Dado $x \in \mathbb{R}^n$, $p \in \mathbb{R}^n$ é chamada *direção de descida a partir de x* se existe $\epsilon > 0$ tal que, para todo $t \in (0, \epsilon]$, $f(x + tp) < f(x)$.

Proposição 2.1.1. Suponha $f \in C^1$ e $\nabla f(x) \neq 0$. Seja p tal que $\nabla f(x)^T p < 0$. Então existe $\epsilon > 0$ tal que para todo $t \in (0, \epsilon]$, $f(x + tp) < f(x)$, ou seja, p é direção de descida.

Dem.: Consideramos a função $\phi(t) \equiv f(x + tp)$. Então $\phi(0) = f(x)$, e aplicando a regra da cadeia temos $\phi'(0) = \nabla f(x)^T p$.

Como $\phi'(0) = \lim_{t \rightarrow 0} \frac{\phi(t) - \phi(0)}{t}$, então para $0 < t < \epsilon$, com ϵ suficientemente pequeno, o sinal de $\phi'(0)$ e o sinal de $\phi(t) - \phi(0)$ deve ser o mesmo.

Como $\nabla f(x)^T p < 0$ temos que $\phi'(0) < 0$ e $\phi(t) - \phi(0) < 0$ para $0 < t < \epsilon$, portanto $f(x + tp) < f(x)$. ■

Definição 2.1.5. Uma função f é Lipschitz contínua com constante γ em um conjunto X , ou seja $f \in Lip_\gamma(X)$, se para todo $x, y \in X$,

$$|f(x) - f(y)| \leq \gamma \|x - y\|.$$

2.2 Condições de Otimalidade

Apesar da definição de minimizador não usar derivadas, é comum analisarmos possíveis minimizadores através de gradientes e hessianas. Para tal consideremos as seguintes proposições.

Proposição 2.2.1. *Condições necessárias de primeira ordem:*

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

Dem.: Ver Friedlander [6].

A condição acima deve ser satisfeita por qualquer candidato a minimizador de funções diferenciáveis. Pontos y tais que $\nabla f(y) = 0$ são chamados pontos estacionários. Nem todos os pontos estacionários são minimizadores de f . Por exemplo, para $f(x) = -x^2$ o ponto $x = 0$ satisfaz $f'(0) = 0$, mesmo não sendo um minimizador local. No caso de funções que possuem derivadas de segunda ordem, a proposição abaixo deve ser satisfeita.

Proposição 2.2.2. *Condições necessárias de segunda ordem:*

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

Dem.: Ver Friedlander [6].

Por $\nabla^2 f(x^*) \geq 0$ queremos dizer que $\nabla^2 f(x^*)$ é semi-definida positiva, ou seja, para qualquer $y \in \mathbb{R}^n$, $y^t \nabla^2 f(x^*) y \geq 0$. Novamente, as condições acima não garantem que x^* seja solução do problema (2.1). Por exemplo, para $f(x) = x^3$ o ponto $x = 0$ satisfaz $f'(0) = 0$ e $f''(0) \geq 0$, mesmo não sendo a origem minimizador local. Condições suficientes são dadas na proposição abaixo.

Proposição 2.2.3. *Condições suficientes de segunda ordem:*

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

Dem.: Ver Friedlander [6].

Com $\nabla^2 f(x^*) > 0$ queremos dizer que $\nabla^2 f(x^*)$ é definida positiva, ou seja, para qualquer $y \in \mathbb{R}^n - \{0\}$, $y^t \nabla^2 f(x^*) y > 0$. Esta última condição não é necessária para que um ponto seja um minimizador de uma função. De fato, para $f(x) = x^4$, o minimizador $x^* = 0$ satisfaz $f'(x^*) = 0$, porém $f''(x^*) = 0$.

Nas seções a seguir, deste capítulo, veremos os principais métodos de otimização sem restrições com uso de derivadas, tais como, Método do Gradiente, Método de Newton, Método Quase-Newton, com destaque para o Método Secante de Powell e Broyden.

2.3 Método do Gradiente

Observe que a direção $p = \frac{-\nabla f(x)}{\|\nabla f(x)\|}$ é a que fornece a menor derivada direcional, dentre todas as direções unitárias a partir do ponto x (lembramos que direção unitária é toda direção tal que $\|p\| = 1$). De fato, seja p uma direção de descida tal que $\nabla f(x)^T p < 0$. Por Cauchy-Schwarz temos $|\nabla f(x)^T p| \leq \|\nabla f(x)\| \|p\|$; sendo p uma direção unitária temos $\|p\| = 1$, logo $-\nabla f(x)^T p \leq \|\nabla f(x)\|$. Portanto, $p = \frac{-\nabla f(x)}{\|\nabla f(x)\|}$ é a direção, entre todas as direções unitárias que satisfazem $\nabla f(x)^T p < 0$, que fornece o maior valor em módulo da derivada direcional. Assim, a função objetivo diminuirá se avançarmos nessa direção, e a máxima diminuição será obtida minimizando, ao longo dela. Desta forma, no método do Gradiente, adotamos a cada iteração a direção de busca determinada por:

$$p_k = -\nabla f(x_k).$$

Os passos de uma iteração k do método do Gradiente, ou de Máxima Descida, estão descritos a seguir.

Algoritmo 2.3.1. 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 $p_k = -\nabla f(x_k)$.

Passo 2: Determinar α_k , minimizador de $f(x_k + \alpha p_k)$ sujeita a $\alpha \geq 0$.

Passo 3: Fazer $x_{k+1} = x_k + \alpha_k p_k$.

Observaremos agora dois casos.

Caso 1: Função objetivo quadrática

Se

$$f(x) = \frac{1}{2}x^t Bx + b^t x + c, \quad (2.2)$$

com B simétrica definida positiva, então existe um único $x^* \in \mathbb{R}^n$ que é minimizador global de f . De fato, se x^* é um ponto estacionário de (2.2) e B é simétrica definida positiva, então $b = -Bx^*$, logo,

$$\begin{aligned} f(x) &= \frac{1}{2}x^t Bx + b^t x + c = \frac{1}{2}x^t Bx - (x^*)^t Bx + c \\ &= \frac{1}{2}(x - x^*)^t B(x - x^*) - \frac{1}{2}(x^*)^t B(x^*) + c > -\frac{1}{2}(x^*)^t B(x^*) + c \\ &= \frac{1}{2}(x^*)^t Bx^* - (x^*)^t Bx^* + c = \frac{1}{2}(x^*)^t Bx^* + b^t x^* + c = f(x^*). \end{aligned}$$

Portanto, como B é simétrica definida positiva e $f(x) > f(x^*)$ para todo x , temos que x^* é o único minimizador global de (2.2).

Neste caso podemos calcular o tamanho do passo α_k que minimiza $f(x_k - \alpha \nabla f_k)$, diferenciando

$$f(x_k - \alpha \nabla f_k) = \frac{1}{2}(x_k - \alpha \nabla f_k)^t B(x_k - \alpha \nabla f_k) + b^t(x_k - \alpha \nabla f_k) + c.$$

Dessa forma o minimizador de $f(x_k - \alpha \nabla f_k)$ sujeito a $\alpha \geq 0$ é determinado por:

$$\alpha_k = \frac{\nabla^t f(x_k) \nabla f(x_k)}{\nabla^t f(x_k) B \nabla f(x_k)}.$$

O teorema 2.3.2 garante a convergência da sequência gerada pelo Algoritmo 2.3.1, quando f é uma função quadrática, para qualquer aproximação inicial e que a ordem de convergência da sequência associada $\{f(x_k)\}$ é linear.

Teorema 2.3.2. *Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função quadrática com matriz Hessiana B simétrica definida positiva. Seja x^* o minimizador global de f .*

Dado $x_0 \in \mathbb{R}^n$, arbitrário, o Algoritmo 2.3.1 gera uma sequência $\{x_k\}$ tal que:

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

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

e

$$f(x_{k+1}) - f(x^*) \leq \left(\frac{(A-a)}{(A+a)} \right)^2 (f(x_k) - f(x^*)),$$

onde “A” e “a” são o maior e o menor autovalor de B, respectivamente.

Dem.: Ver Luenberger [9].

Caso 2: Função objetivo não quadrática

Enunciaremos um teorema que se refere as condições de convergência para o método do Gradiente no caso geral.

Teorema 2.3.3. *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 2.3.1 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 linearmente a $f(x^*)$ com taxa não superior a $(\frac{(A-a)}{(A+a)})^2$, onde “A” e “a” são o maior e o menor autovalor de $\nabla^2 f(x^*)$, respectivamente.*

Dem.: Ver Luenberger [9].

Neste caso, dizemos que o Algoritmo 2.3.1 está bem definido se for possível completar o Passo 2 com um número finito de tentativas para α .

2.4 Método de Newton

Para compreendermos melhor a direção adotada pelo método de Newton, consideremos inicialmente a seguinte proposição.

Proposição 2.4.1. *Se f é uma função quadrática, como em (2.2) onde a matriz hessiana B é definida positiva, $x_0 \in \mathbb{R}^n$ é dado e a direção $p \in \mathbb{R}^n$ é*

definida por:

$$p = -B^{-1}(Bx_0 + b), \quad (2.3)$$

então

$$x^* \equiv x_0 + p \quad (2.4)$$

é o minimizador global de f em $\mathbb{R}^n$.

Dem.: Ver Friedlander [6].

Logo a direção p é a solução do sistema linear

$$Bp = -(Bx_0 + b) = -\nabla f(x_0).$$

Portanto, minimizar uma função quadrática com hessiana definida positiva é um problema equivalente a resolver um sistema linear com matriz simétrica e definida positiva.

Se a função não é quadrática e temos uma aproximação x_k da solução de (2.1), podemos utilizar o resultado anterior de funções quadráticas aproximando $f(x_k + p)$ por uma quadrática através dos 3 primeiros termos da sua expansão em série de Taylor em torno de x_k :

$$q(p) = f(x_k) + \nabla^t f(x_k)p + \frac{1}{2}p^t \nabla^2 f(x_k)p.$$

Chamamos de $c = q(0) = f(x_k)$, $b = \nabla q(0) = \nabla f(x_k)$, $B = \nabla^2 q(0) = \nabla^2 f(x_k)$. Se escrevemos $q(p) = \frac{1}{2}p^t Bp + b^t p + c$ e se $\nabla^2 f(x_k)$ é definida positiva, podemos calcular o minimizador global desta quadrática a partir de $p_0 = 0$.

Assim, obtemos

$$p^* = -B^{-1}(Bp_0 + b) = -B^{-1}b = -(\nabla^2 f_k)^{-1} \nabla f_k.$$

Isso nos sugere escolher a direção $p_k = -(\nabla^2 f_k)^{-1} \nabla f_k$, que é a base do método de Newton. Nesse método, devemos encontrar a cada iteração uma direção de busca determinada por:

$$p_k = -(\nabla^2 f_k)^{-1} \nabla f_k. \quad (2.5)$$

Os passos de uma iteração k do método de Newton, estão descritos a seguir.

Algoritmo 2.4.2. *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: *Determinar p_k tal que*

$$\nabla^2 f(x_k) p_k = -\nabla f(x_k).$$

Passo 2: *Fazer $x_{k+1} = x_k + p_k$.*

Como a matriz hessiana $\nabla^2 f_k$ pode nem sempre ser definida positiva, p_k pode nem sempre ser uma direção de descida. Se isso ocorrer temos formas de contornar esse problema (ver [4]).

O teorema a seguir determina condições para a convergência do método de Newton.

Teorema 2.4.3. *Suponha que f é duas vezes diferenciável e que a hessiana $\nabla^2 f(x)$ é Lipschitz contínua em uma vizinhança da solução x^* . Suponha ainda que condições suficientes de segunda ordem para o problema (2.1) são satisfeitas em x^* . Considere a sequência $\{x_k\}$ gerada pelo Algoritmo 2.4.2. Então*

1. *Se o ponto inicial x_0 está suficientemente perto de x^* , a sequência de iterações converge para x^* ;*
2. *A taxa de convergência de $\{x_k\}$ é quadrática, isto é*

$$\|x^{k+1} - x^*\| \leq a \|x^k - x^*\|^2,$$

onde $a > 0$ e

3. *A sequência das normas dos gradientes $\{\|\nabla f_k\|\}$ converge quadraticamente para zero.*

Dem.: Ver Nocedal & Wright [14].

O método de Newton apresenta algumas desvantagens: obriga o cálculo, em cada iteração, não só do gradiente da função como também da matriz hessiana, e a resolução de um sistema linear por iteração. O cálculo da hessiana pode consumir muito tempo de computação ou ainda, muitas vezes, ser difícil, ou mesmo impossível. Para estes casos podemos recorrer aos métodos Quase-Newton.

2.5 Métodos Quase-Newton

Suponha que a direção de busca tem a forma:

$$p_k = -B_k^{-1} \nabla f_k, \quad (2.6)$$

onde a matriz simétrica B_k é atualizada a cada iteração, com o objetivo de aproximar $\nabla^2 f_k$. A fórmula (2.6) caracteriza a direção de busca dos métodos Quase-Newton. Dentre esses, destacamos os métodos secantes, que descreveremos a seguir.

É desejável determinar matrizes B_k de modo que o trabalho computacional do método resultante seja menor que o do método de Newton e tais que a sequência $\{x_k\}$ gerada por ele, quando converge, tenha uma boa taxa de convergência.

Se quisermos obter um comportamento melhor do que o do método do gradiente, precisaremos utilizar alguma informação de segunda ordem.

Outra vez a análise específica das funções quadráticas é pertinente.

Se x^* é o minimizador global de uma quadrática com matriz hessiana definida positiva, o método de Newton encontra x^* numa única iteração a partir de qualquer $x_0 \in \mathbb{R}^n$. O método do gradiente converge a x^* , mas, pode exigir um número muito grande de iterações.

Um método intermediário para funções quadráticas deveria se aproximar de x^* mais rapidamente que o método do Gradiente, sem estar baseado no conhecimento completo da matriz hessiana, como o método de Newton. Surgem então os métodos quase-Newton.

Se

$$f(x) = \frac{1}{2}x^t Bx + b^t x + c,$$

temos que

$$\nabla f(x) = Bx + b$$

e

$$\nabla f(x + p) - \nabla f(x) = B(x + p) - Bx = Bp \text{ para todo } p \in \mathbb{R}^n.$$

Temos então as seguintes equações:

$$\nabla f(x + p) - \nabla f(x) = Bp$$

ou

$$B^{-1}(\nabla f(x + p) - \nabla f(x)) = p.$$

Observemos que estas equações fornecem informação sobre B ou B^{-1} utilizando ∇f em dois pontos. São conhecidas como equações secantes.

Estas idéias sugerem o seguinte algoritmo:

Algoritmo 2.5.1. *Sejam $x_0 \in \mathbb{R}^n$ arbitrário, $B_0 \in \mathbb{R}^{n \times n}$ simétrica e definida positiva. Seja $k = 0$.*

Se $\nabla f(x_k) \neq 0$, os passos para obter x_{k+1} são:

Passo 1: *Calcular p_k tal que $B_k p_k = -\nabla f(x_k)$.*

Passo 2: *Determinar $x_{k+1} = x_k + p_k$.*

Passo 3: *Obter B_{k+1} simétrica e definida positiva tal que*

$$B_{k+1} p_k = \nabla f(x_{k+1}) - \nabla f(x_k). \quad (2.7)$$

Passo 4: $k = k + 1$

Neste algoritmo, a Hessiana é aproximada satisfazendo a equação secante. Esta aproximação pode ser obtida de muitas formas. Apresentaremos aqui a atualização secante simétrica de Powell, por esta ser a base das aproximações usadas no algoritmo descrito no capítulo 4. Esta atualização não requer avaliações adicionais da função ou do gradiente, se mantém sempre simétrica mas não definida positiva. A atualização secante definida positiva que é, em prática, considerada a mais próspera atualização secante para a Hessiana, é

a BFGS - esta forma de atualização foi descoberta independentemente por Broyden [2], Fletcher [5], Goldfarb [7] e Shanno [20] em 1970.

Há também métodos secantes que aproximam a inversa da Hessiana. Neste caso, a resolução do sistema linear do passo 1 do algoritmo 2.5.1 é substituída por um produto da matriz B_k por $-\nabla f(x_k)$, ou seja:

$$p_k = -B_k \nabla f(x_k).$$

2.5.1 A atualização secante simétrica de Powell e Broyden

Consideremos o problema (2.1). Vamos supor que, para resolver o problema, obtemos x_{k+1} a partir de x_k com tamanho de passo igual a um, e que vamos aproximar $\nabla^2 f(x_{k+1})$ por B_{k+1} usando técnicas secantes. Para isso consideraremos a equação secante

$$B_{k+1} s_k = y_k, \quad (2.8)$$

onde

$$s_k = x_{k+1} - x_k, \quad y_k = \nabla f(x_{k+1}) - \nabla f(x_k). \quad (2.9)$$

Podemos agora usar a atualização secante para aproximar $\nabla^2 f(x_{k+1})$ por

$$(B_{k+1})_1 = B_k + \frac{(y_k - B_k s_k) s_k^T}{s_k^T s_k}, \quad (2.10)$$

onde $B_k \in \mathbb{R}^{n \times n}$ é nossa aproximação para $\nabla^2 f(x_k)$. O principal problema com (2.10) é que mesmo que B_k seja simétrica, $(B_{k+1})_1$ não o será, a menos que $y_k - B_k s_k$ seja um múltiplo de s_k . Temos várias razões para que a aproximação da hessiana deva ser simétrica. Então buscamos B_{k+1} que verifique $B_{k+1} = B_{k+1}^T$ como também (2.8).

Uma idéia razoável para construir $(B_{k+1})_2$ simétrica, a partir de $(B_{k+1})_1$, é fazer a projeção na norma de Frobenius de $(B_{k+1})_1$ no subespaço S das matrizes simétricas. É fácil provar que, nesse caso,

$$(B_{k+1})_2 = \frac{1}{2}((B_{k+1})_1 + (B_{k+1})_1^T) = B_k + \frac{(y_k - B_k s_k) s_k^T + s_k (y_k - B_k s_k)^T}{2 s_k^T s_k}.$$

Agora $(B_{k+1})_2$ é simétrica, se B_k é simétrica, porém não satisfaz, em geral, a equação secante. Contudo, alguém pode considerar continuar este processo, gerando $(B_{k+1})_3, (B_{k+1})_4, \dots$ por

$$(B_{k+1})_{2\nu+1} = (B_{k+1})_{2\nu} + \frac{(y_k - (B_{k+1})_{2\nu}s_k)s_k^T}{s_k^T s_k},$$

$$(B_{k+1})_{2\nu+2} = \frac{1}{2}((B_{k+1})_{2\nu+1} + (B_{k+1})_{2\nu+1}^T).$$

Aqui cada $(B_{k+1})_{2\nu+1}$ é a matriz mais próxima de $(B_{k+1})_{2\nu}$ em $Q(y_k, s_k) = \{B \in \mathbb{R}^{n \times n} | Bs_k = y_k\}$, e cada $(B_{k+1})_{2\nu+2}$ é a matriz simétrica mais próxima de $(B_{k+1})_{2\nu+1}$ onde $\nu \in N$. M. J. D. Powell mostrou que esta sequência de matrizes converge para

$$(B_{k+1}) = B_k + \frac{(y_k - B_k s_k)s_k^T + s_k(y_k - B_k s_k)^T}{s_k^T s_k} - \frac{\langle y_k - B_k s_k, s_k \rangle s_k s_k^T}{(s_k^T s_k)^2}. \quad (2.11)$$

A atualização secante (2.11) é também conhecida como atualização simétrica de Powell e Broyden (PSB) [15]. É fácil confirmar que B_{k+1} determinada por (2.11) obedece a equação secante (2.8), e que $B_{k+1} - B_k$ é uma matriz simétrica de posto dois. Na realidade Dennis e Moré (1977) [3] provaram o seguinte teorema.

Teorema 2.5.2. *Seja $B_k \in \mathbb{R}^{n \times n}$ simétrica, $s_k, y_k \in \mathbb{R}^n$, $s_k \neq 0$. Então a única solução para*

$$\begin{aligned} & \text{Minimizar}_{B \in \mathbb{R}^{n \times n}} \|B - B_k\|_F \\ & \text{sujeito a } Bs_k = y_k, \quad (B - B_k) \text{ simétrica,} \end{aligned} \quad (2.12)$$

é determinada por (2.11).

Dem: ver Dennis e Moré (1977) [3].

O teorema a seguir estabelece que o método converge localmente para algum zero isolado de $\nabla f(x)$, desde que $\nabla^2 f(x^*)$ seja simétrica e não singular, porém não necessariamente definida positiva.

Teorema 2.5.3. *Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ duas vezes continuamente diferenciável em um conjunto convexo aberto $D \subset \mathbb{R}^n$, e que a hessiana $\nabla^2 f(x)$ é Lipschitz contínua em D . Suponha ainda que condições suficientes de segunda ordem para o problema (2.1) são satisfeitas em $x^* \in D$ e que B_0 é simétrica. Então existem constantes positivas ϵ, δ tais que se $\|x_0 - x^*\|_2 < \epsilon$ e $\|B_0 - \nabla^2 f(x^*)\|_2 \leq \delta$, o método PSB que gera $\{x_k\}$ por*

$$x_{k+1} = x_k - B_k^{-1} \nabla f(x_k),$$

$$s_k = x_{k+1} - x_k, \quad y_k = \nabla f(x_{k+1}) - \nabla f(x_k),$$

$$(B_{k+1}) = B_k + \frac{(y_k - B_k s_k) s_k^T + s_k (y_k - B_k s_k)^T}{s_k^T s_k} - \frac{\langle y_k - B_k s_k, s_k \rangle s_k s_k^T}{(s_k^T s_k)^2},$$

é bem definido, $\{x_k\}$ permanece em D e converge q -superlinearmente para x^ , ou seja, para alguma sequência $\{c_k\}$ que converge a 0,*

$$\|x_{k+1} - x^*\| \leq c_k \|x_k - x^*\|.$$

Dem: ver Dennis & Schnabel [4].

Capítulo 3

Estratégias de Globalização

3.1 Introdução

As Proposições 2.2.1, 2.2.2 e 2.2.3 têm importância fundamental na classificação de possíveis soluções de (2.1). No entanto a maioria das demonstrações de convergência de algoritmos se contenta apenas em provar que os pontos limites das sequências geradas são pontos estacionários da função objetivo. Diremos que um algoritmo é globalmente convergente se gera uma sequência $\{x_k\}_{k=0}^{\infty}$, a partir de x_0 arbitrário, tal que todo ponto limite desta sequência é um ponto estacionário de f .

Para que os algoritmos desfrutem destas propriedades de convergência global é preciso que apresentem decréscimo suficiente da função objetivo de iteração a iteração. Surgem então estratégias como as de “busca linear” e de região de confiança, conhecidas como estratégias de globalização, as quais descreveremos neste capítulo.

3.2 Métodos de Busca Linear

Nos métodos de busca linear, dada uma aproximação x_k , e escolhida uma direção p_k , associamos à iteração corrente a função de uma variável $\phi(\alpha) = f(x_k + \alpha p_k)$, $\alpha \in \mathbb{R}$. A intenção é encontrar α_k que defina o novo vetor $x_{k+1} = x_k + \alpha_k p_k$. Para tanto, é desejável que sejam feitas poucas avaliações de f , e é usual pedir que

$$f(x_{k+1}) \leq f(x_k).$$

Diferentes escolhas de α_k definem os diferentes métodos. A liberdade que esse tipo de algoritmo permite, na escolha dos passos, exige normalmente o uso de alguns artifícios (decréscimo suficiente de f , por exemplo) para assegurar convergência global.

O decréscimo suficiente impede passos grandes com pouco decréscimo. Um critério de busca linear bastante usado pede que α_k verifique

$$f(x_k + \alpha_k p_k) < f(x_k) + c_1 \nabla^t f(x_k) \alpha_k p_k, \quad \text{para todo } k \in N,$$

onde $c_1 \in (0, 1)$ é uma constante. Esta condição exige que o decréscimo seja, em certo sentido, proporcional ao tamanho do passo.

Observemos que, sendo p_k uma direção de descida, resulta

$$c_1 \nabla^t f(x_k) \alpha_k p_k < 0$$

e, portanto, essa condição significa que queremos algo mais que simplesmente um decréscimo no valor da função. Chamamos essa condição de condição de Armijo [6].

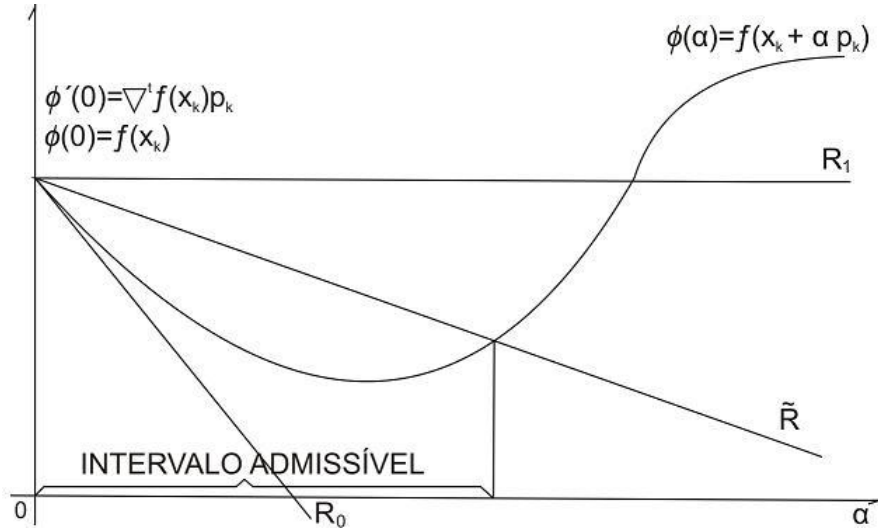


Figura 3.1: Condição de Armijo.

Na figura 3.1, R_0 é a reta que passa pelo ponto $(0, \phi(0))$ e tem coeficiente angular $\phi'(0)$. A equação de R_0 é

$$z = \phi(0) + \phi'(0)\alpha.$$

R_1 é a reta que passa pelo mesmo ponto e tem coeficiente angular 0. $\tilde{R}$ é uma reta que passa pelo mesmo ponto com coeficiente angular entre $\phi'(0)$ e 0. Portanto, o coeficiente angular de $\tilde{R}$ pode ser escrito da forma $c_1\phi'(0)$, com $c_1 \in (0, 1)$. Logo, a equação de $\tilde{R}$ é:

$$z = \phi(0) + c_1\phi'(0)\alpha.$$

Substituindo nesta equação $\phi(0)$ por $f(x_k)$ e $\phi'(0)$ por $\nabla^t f(x_k)p_k$, obtemos

$$z = f(x_k) + c_1\alpha\nabla^t f(x_k)p_k.$$

Então, os valores de α que verificam a condição de Armijo são os que estão na região admissível na Figura 3.1.

Para impedir que as direções sejam “quase” ortogonais a $\nabla f(x_k)$, pediremos que, dada uma constante $\beta \in (0, 1)$,

$$\nabla^t f(x_k)p_k \leq -\beta \|\nabla f(x_k)\| \|p_k\|, \text{ para todo } k \in N.$$

Se θ é o ângulo entre $\nabla f(x_k)$ e p_k ,

$$\cos \theta = \frac{\nabla^t f(x_k)p_k}{\|\nabla f(x_k)\| \|p_k\|},$$

e, consequentemente,

$$\cos \theta \leq -\beta.$$

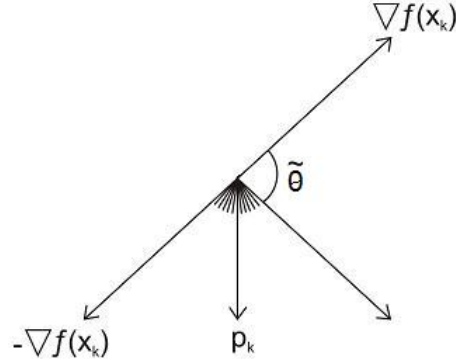


Figura 3.2: Condição do ângulo

Na Figura 3.2, se $\tilde{\theta}$ é um ângulo tal que $\cos \tilde{\theta} = -\beta$, p_k deve formar um ângulo maior que $\tilde{\theta}$ com $\nabla f(x_k)$. Observemos um algoritmo para minimizar funções sem restrições, que seja o mais geral possível e que incorpore essas condições.

Algoritmo 3.2.1.

Sejam $\sigma > 0$, c_1 e $\beta \in (0, 1)$ constantes dadas.

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: Escolher $p_k \in \mathbb{R}^n$, tal que

- (i) $\|p_k\| \geq \sigma \|\nabla f(x_k)\|$;
- (ii) $\nabla^t f(x_k) p_k \leq -\beta \|\nabla f(x_k)\| \|p_k\|$.

Passo 2: (Busca Linear)

- (i) $\alpha = 1$;
- (ii) Se $f(x_k + \alpha p_k) < f(x_k) + c_1 \alpha \nabla^t f(x_k) p_k$, ir a (iv);
- (iii) Escolher $\tilde{\alpha} \in [0.1\alpha, 0.9\alpha]$. Fazer $\alpha = \tilde{\alpha}$ e ir a (ii);
- (iv) Fazer $\alpha_k = \alpha$, $x_{k+1} = x_k + \alpha_k p_k$ e $k = k + 1$.

Lema 3.2.2. O Algoritmo 3.2.1 está bem-definido. (É possível completar a busca linear com um número finito de tentativas para α).

Dem.: Ver Friedlander [6].

Teorema 3.2.3. (Convergência Global) O Algoritmo 3.2.1 pára com algum valor k tal que $\nabla f(x_k) = 0$, ou gera uma sequência infinita $\{x_k\}$ tal que qualquer ponto de acumulação dela é um ponto estacionário de f .

Dem.: Ver Friedlander [6].

Observemos que, neste teorema, não é garantida a convergência da sequência $\{x_k\}$. No entanto, ele afirma que se existe $\lim_{k \rightarrow \infty} x_k$, então este limite é um ponto estacionário. Finalmente, se a sequência é limitada, existe um ponto de acumulação e este deve ser um ponto estacionário.

3.3 Métodos de Região de Confiança

Os métodos de Região de Confiança geram passos com a ajuda de um modelo quadrático da função objetivo. Nesses métodos, definimos uma região ao redor da atual iteração na qual confiamos no modelo para ser uma representação adequada da função objetivo, e então escolhemos o passo que leve ao minimizador aproximado do modelo nesta região de confiança. Em consequência, são escolhidos direção e comprimento do passo simultaneamente. Se um passo não for aceitável, reduzimos o raio da região e achamos um novo

minimizador. Em geral, a direção do passo muda sempre que o raio da região de confiança é alterado.

O tamanho da região de confiança é crítico para eficiência de cada passo. Se a região for muito pequena, o algoritmo perde uma oportunidade para dar um passo significativo, movendo-se muito mais lentamente para o minimizador da função objetivo nesta região. Se muito grande, o minimizador do modelo pode estar longe do minimizador da função objetivo na região. Assim, podemos ter que reduzir o raio da região e tentar novamente. Na prática, escolhemos o raio da região de confiança de acordo com o desempenho do algoritmo durante as iterações anteriores. Se o modelo tem passos bons geralmente seguros, predizendo o comportamento da função objetivo com precisão ao longo destes passos, o raio da região de confiança é uniformemente aumentado, permitindo passos longos e ambiciosos. Por outro lado, um passo falho indica que nosso modelo é uma representação inadequada da função objetivo na região de confiança atual. Assim reduzimos o raio da região e tentamos novamente.

Assumiremos que os dois primeiros termos do modelo quadrático q_k em cada iteração x_k são idênticos aos dois primeiros termos da expansão da série de Taylor de f em torno de x_k . Especificamente, temos:

$$q_k(p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T B_k p, \quad (3.1)$$

onde B_k é alguma matriz simétrica. Temos também

$$f(x_k + p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T \nabla^2 f(x_k + tp) p, \quad (3.2)$$

para algum escalar $t \in (0, 1)$, e como $q_k(p) = f_k + \nabla f_k^T p + o(\|p\|^2)$, e a diferença entre $q_k(p)$ e $f(x_k + p)$ é $o(\|p\|^2)$, o erro da aproximação é pequeno quando p é pequeno.

Quando B_k é igual à Hessiana verdadeira $\nabla^2 f(x_k)$, a função modelo na verdade coincide com os três termos da série de Taylor. O erro da aproximação é $o(\|p\|^3)$ neste caso; assim, este modelo é especialmente preciso quando $\|p\|$ é pequena.

Para obter cada passo, buscaremos uma solução do subproblema

$$\begin{aligned} \min_{p \in \mathbb{R}^n} q_k(p) &= f_k + \nabla f_k^T p + \frac{1}{2} p^T B_k p. \\ \text{s.a } \|p\| &\leq \Delta_k, \end{aligned} \quad (3.3)$$

onde $\Delta_k > 0$ é o raio da região de confiança. Para o momento, definiremos $\|\cdot\|$ como a norma Euclidiana, de maneira que a solução p^* de (3.3) seja o minimizador de q_k na bola de raio Δ_k . Assim, a técnica de região de confiança exige que resolvamos uma sucessão de subproblemas (3.3) nos quais função objetivo e restrição (a qual pode ser escrita como $p^T p \leq \Delta_k^2$) são ambas quadráticas. Quando B_k é definida positiva e $\|B_k^{-1} \nabla f_k\| \leq \Delta_k$, a solução de (3.3) é fácil de ser identificada - é simplesmente a minimização irrestrita da quadrática $q_k(p)$, gerando $p_k^B = -B_k^{-1} \nabla f_k$. Neste caso, nós chamamos p_k^B o passo completo. A solução de (3.3) não é tão óbvia em outros casos, mas normalmente pode ser encontrada sem muito esforço. Em todo caso, precisamos de somente uma solução aproximada para obter convergência e bom comportamento prático.

3.3.1 Esboço do algoritmo

O primeiro assunto a surgir na definição do método de região de confiança é a estratégia para escolha do raio da região de confiança Δ_k a cada iteração. Baseamos esta escolha na relação entre a função modelo q_k da seção anterior e a função objetivo f , determinando

$$\gamma_k = \frac{f(x_k) - f(x_k + p_k)}{q_k(0) - q_k(p_k)}. \quad (3.4)$$

O numerador é chamado redução efetiva, e o denominador é a redução prevista. Note que o passo p_k é obtido minimizando o modelo q_k em uma região que inclui o passo $p = 0$; assim a redução prevista será sempre não negativa. Deste modo, se γ_k é negativo, o novo valor da função objetivo $f(x_k + p_k)$ é maior que o valor atual $f(x_k)$, e portanto o passo deve ser rejeitado.

Por outro lado, se γ_k está próximo de 1, há boa relação entre o modelo q_k e a função f ; assim é seguro ampliar a região de confiança para a próxima iteração. Se γ_k é positivo mas não próximo de 1, não alteraremos a região de confiança, mas se está perto de zero ou é negativo, encolheremos a região de confiança. O seguinte algoritmo descreve o processo.

Algoritmo 3.3.1.

Seja $\bar{\Delta} > 0, \Delta_0 \in (0, \bar{\Delta})$, e $\eta \in [0, \frac{1}{4})$.

Para $k = 0, 1, 2, \dots$

Passo 1: Obtenha p_k (aproximadamente) resolvendo (3.3);

Passo 2: Avalie γ_k por (3.4);

Passo 3: se $\gamma_k < \frac{1}{4}$
 $\Delta_{k+1} = \frac{1}{4} \|p_k\|$

Passo 4: senão

Passo 5: se $\gamma_k > \frac{3}{4}$ e $\|p_k\| = \Delta_k$
 $\Delta_{k+1} = \min(2\Delta_k, \bar{\Delta})$

Passo 6: senão
 $\Delta_{k+1} = \Delta_k$;

Passo 7: se $\gamma_k > \eta$
 $x_{k+1} = x_k + p_k$

Passo 8: senão
 $x_{k+1} = x_k$;

fim (para).

Aqui $\bar{\Delta} > 0$ é um limitante global para o tamanho do passo. Note que o raio é aumentado somente se $\|p_k\|$ na verdade alcança o limite da região de confiança. Se o passo ficar estritamente dentro da região, deduzimos que o valor atual de Δ_k não está interferindo com o progresso do algoritmo e assim deixamos seu valor inalterado para a próxima iteração. Para transformarmos o Algoritmo (3.1) em um algoritmo prático, precisamos enfocar a resolução de (3.3). Primeiro descreveremos duas estratégias para achar soluções aproximadas que alcançam pelo menos boa redução em q_k , como a redução alcançada pelo denominado ponto de Cauchy. Este ponto é simplesmente o minimizador de q_k ao longo da direção de máxima descida $-\nabla f_k$, sujeito ao limite da região de confiança. A primeira estratégia de aproximação é o *método dog-leg (perna-de-Cachorro)*, o qual é apropriado quando a Hessiana do modelo B_k é definida positiva. A segunda estratégia, devido a Steihaug [21], é muito apropriada quando B_k é a Hessiana exata $\nabla^2 f(x_k)$ e quando essa matriz é grande e esparsa.

3.3.2 O Ponto de Cauchy

Sabemos que os métodos de busca linear não requerem tamanho de passo ótimo para ter bons resultados de convergência. De fato, somente uma aproximação do tamanho de passo ótimo que satisfaça certos critérios é necessária. Uma situação similar ocorre no método de região de confiança. Embora em princípio busquemos a solução ótima do subproblema (3.3), é suficiente, para convergência, acharmos uma solução aproximada p_k que está dentro da região de confiança e que fornece uma redução suficiente no modelo. A redução suficiente pode ser quantificada em termos do ponto de Cauchy que denotamos por p_k^c e definimos pelo seguinte procedimento:

Algoritmo 3.3.2. .

Determine o vetor p_k^s que resolve uma versão linear de (3.3), isto é,

$$\begin{aligned} p_k^s &= \arg \min_{p \in \mathbb{R}^n} f_k + \nabla f_k^T p \\ \text{s.a} \quad &\|p\| \leq \Delta_k. \end{aligned} \quad (3.5)$$

Calcule o escalar $\tau_k > 0$ que minimiza $q_k(\tau p_k^s)$ na região de confiança, isto é,

$$\begin{aligned} \tau_k &= \arg \min_{\tau > 0} q_k(\tau p_k^s) \\ \text{s.a} \quad &\|\tau p_k^s\| \leq \Delta_k; \end{aligned} \quad (3.6)$$

faça $p_k^c = \tau_k p_k^s$.

Na realidade, é fácil de escrever por uma fórmula fechada a definição do ponto de Cauchy. A solução de (3.5) é simplesmente

$$p_k^s = -\frac{\Delta_k}{\|\nabla f_k\|} \nabla f_k.$$

Para obtermos τ_k explicitamente, consideramos os casos de $\nabla f_k^T B_k \nabla f_k \leq 0$ e $\nabla f_k^T B_k \nabla f_k > 0$ separadamente. Para o primeiro caso, a função $q_k(\tau p_k^s)$ decresce monotonicamente em τ sempre que $\nabla f_k \neq 0$, assim τ_k é simplesmente o maior valor que satisfaz o limite de região de confiança, isto é, $\tau_k = 1$. Para o caso $\nabla f_k^T B_k \nabla f_k > 0$, $q_k(\tau p_k^s)$ é uma quadrática convexa em τ , assim τ_k ou é o minimizador irrestrito desta quadrática, $\frac{\|\nabla f_k\|^3}{(\Delta_k \nabla f_k^T B_k \nabla f_k)}$,

ou o minimizador está no limite da região de confiança e $\tau_k = 1$. Em resumo, temos

$$p_k^c = -\tau_k \frac{\Delta_k}{\|\nabla f_k\|} \nabla f_k, \quad (3.7)$$

onde

$$\tau_k = \begin{cases} 1 & \text{se } \nabla f_k^T B_k \nabla f_k \leq 0; \\ \min\left(\frac{\|\nabla f_k\|^3}{(\Delta_k \nabla f_k^T B_k \nabla f_k)}, 1\right) & \text{caso contrário.} \end{cases} \quad (3.8)$$

O passo de Cauchy p_k^c é computacionalmente barato, pois não requer fatorações de matrizes.

É de crucial importância decidir se uma solução aproximada do subproblema de região de confiança é aceitável. Especificamente, um método de região de confiança terá convergência se o passo p_k atingir uma redução suficiente em q_k , isto é, uma redução no modelo q_k que é pelo menos algum múltiplo fixo da redução atingida pelo passo de Cauchy a cada iteração. Consideraremos agora dois métodos para encontrar soluções aproximadas de (3.3). Ao longo desta sub-seção, e da próxima, nosso foco será o trabalho interno de uma iteração simples. Assim, omitiremos os índices “k” para simplificar a notação. Desta forma, reescrevemos o subproblema de região de confiança (3.3) como a seguir:

$$\begin{aligned} \min_{p \in \mathbb{R}^n} q(p) &\stackrel{\text{def}}{=} f + g^T p + \frac{1}{2} p^T B p, \\ \text{s.a. } \|p\| &\leq \Delta. \end{aligned} \quad (3.9)$$

onde $g = \nabla f$.

3.3.3 O Método de *dog-leg*

Começamos examinando o efeito do raio da região de confiança Δ na solução $p^*(\Delta)$ do subproblema (3.9). Quando B é definida positiva, já notamos que

o minimizador irrestrito de $q(p)$ é o passo completo $p^B = -B^{-1}g$. Quando este ponto é factível para (3.9), ele é obviamente uma solução, donde temos

$$p^*(\Delta) = p^B, \text{ quando } \Delta \geq \|p^B\|. \quad (3.10)$$

Quando Δ é muito pequeno, a restrição $\|p\| \leq \Delta$ assegura que o termo quadrático de q tem efeito pequeno na solução de (3.9). A verdadeira solução $p(\Delta)$ é aproximadamente igual à solução que nós obteríamos minimizando a função linear $f + g^T p$ sujeita a $\|p\| \leq \Delta$, isto é,

$$p^*(\Delta) \approx -\Delta \frac{g}{\|g\|}, \text{ quando } \Delta \text{ é pequeno.} \quad (3.11)$$

Para valores intermediários de Δ , a solução $p^*(\Delta)$ tipicamente segue uma trajetória curva.

O método dog-leg obtém uma solução aproximada, substituindo a trajetória curva de $p^*(\Delta)$ por um caminho que consiste em dois segmentos lineares. O primeiro segmento linear se estende da origem ao minimizador irrestrito, ao longo da direção de máxima descida, sendo definido por

$$p^U = -\frac{g^T g}{g^T B g} g, \quad (3.12)$$

enquanto o segundo segmento linear inicia-se no minimizador irrestrito e se estende na direção p^B . Formalmente, nós denotamos esta trajetória por $\tilde{p}(\tau)$ para $\tau \in [0, 2]$, onde

$$\tilde{p}(\tau) = \begin{cases} \tau p^U, & 0 \leq \tau \leq 1 \\ p^U + (\tau - 1)(p^B - p^U), & 1 \leq \tau \leq 2 \end{cases} \quad (3.13)$$

O método dog-leg escolhe p que minimiza o modelo q ao longo deste caminho, sujeito ao limite da região de confiança. Na realidade, nem é mesmo necessário levar a busca até o fim, porque o caminho da “perna de cachorro” cruza no máximo uma vez o limite da região de confiança e o ponto de

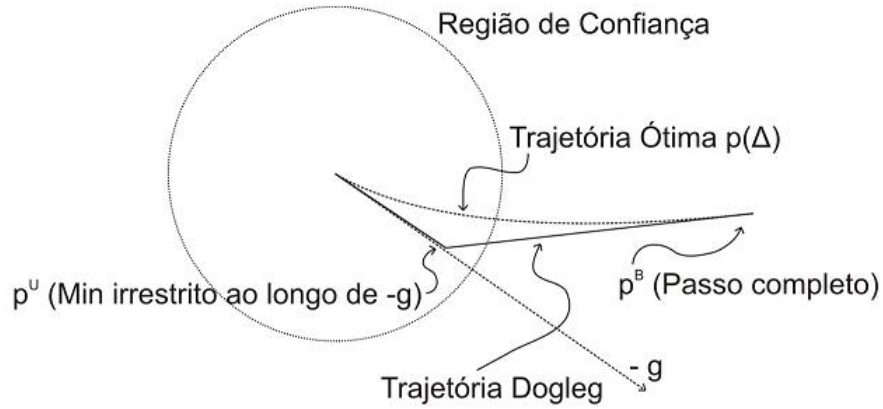


Figura 3.3: Trajetória exata e aproximação Dogleg.

intersecção pode ser calculado analiticamente. Provamos estas afirmações no seguinte lema.

Lema 3.3.3. *Seja B definida positiva. Então*

- (i) $\|\tilde{p}(\tau)\|$ é uma função crescente em τ , e
- (ii) $q(\tilde{p}(\tau))$ é uma função decrescente em τ .

Dem.: Para $\tau \in [0, 1]$ é fácil provar que (i) e (ii) acontecem, pois neste intervalo temos $\tilde{p}(\tau) = \tau p^U$. Assim restringiremos nossa atenção para o caso de $\tau \in [1, 2]$.

Para (i), definimos $h(\alpha)$ por

$$\begin{aligned}
 h(\alpha) &= \frac{1}{2} \|\tilde{p}(1 + \alpha)\|^2 \\
 &= \frac{1}{2} \|p^U + \alpha(p^B - p^U)\|^2 \\
 &= \frac{1}{2} \|p^U\|^2 + \alpha(p^U)^T(p^B - p^U) + \frac{1}{2}\alpha^2 \|p^B - p^U\|^2.
 \end{aligned}$$

Nosso resultado estará provado se pudermos mostrar que $h'(\alpha) \geq 0$ para $\alpha \in (0, 1)$. Agora,

$$\begin{aligned}
h'(\alpha) &= -(p^U)^T(p^U - p^B) + \alpha \|(p^U - p^B)\|^2 \\
&\geq -(p^U)^T(p^U - p^B) \\
&= \frac{g^T g}{g^T B g} g^T \left(-\frac{g^T g}{g^T B g} g + B^{-1} g \right) \\
&= g^T g \frac{g B^{-1} g}{g^T B g} \left[1 - \frac{(g^T g)^2}{(g^T B g)(g^T B^{-1} g)} \right] \\
&\geq 0,
\end{aligned}$$

Para (ii), definiremos $\hat{h}(\alpha) = q(\tilde{p}(1 + \alpha))$ e mostraremos que $\hat{h}'(\alpha) \leq 0$ para $\alpha \in (0, 1)$. Substituindo (3.13) em (3.9) e derivando em relação a α , obtemos,

$$\begin{aligned}
h'(\alpha) &= (p^B - p^U)^T(g + Bp^U) + \alpha(p^B - p^U)^T B(p^B - p^U) \\
&\leq (p^B - p^U)^T(g + Bp^U + B(p^B - p^U)) \\
&= (p^B - p^U)^T(g + Bp^B) = 0,
\end{aligned}$$

demonstrando o resultado. ■

Segue deste lema que o caminho $\tilde{p}(\tau)$ intercepta o limite da região de confiança $\|p\| = \Delta$ em exatamente um ponto se $\|p^B\| \geq \Delta$, e em nenhuma parte caso contrário. Como q é decrescente ao longo do caminho, o valor escolhido de p será p^B se $\|p^B\| \leq \Delta$; caso contrário é o ponto de intersecção da perna-de-cachorro e o limite da região de confiança. No último caso, nós calculamos o valor aproximado de τ resolvendo a seguinte equação quadrática em uma variável:

$$\|p^U + (\tau - 1)(p^B - p^U)\|^2 = \Delta^2.$$

A estratégia de perna-de-cachorro pode ser adaptada para o caso da matriz B ser indefinida. Neste caso o passo completo p^B não é o minimizador irrestrito de q .

3.3.4 Aproximação de Steihaug

A implementação do método de Steihaug é baseada no algoritmo de gradientes conjugados (GC), um algoritmo iterativo para resolver sistemas lineares com matrizes de coeficientes simétricas definidas positivas [8] e [10]. Nossos comentários nesta seção se concentram na diferença entre GC e a aproximação de Steihaug, que é essencialmente quando o algoritmo termina ou quando saímos da região de confiança $\|p\| \leq \Delta$, ou ainda quando encontramos a direção de curvatura negativa de B .

A aproximação de Steihaug pode ser escrita formalmente como a seguir:

Algoritmo 3.3.4.

Dado $\epsilon > 0$;

Seja $p^0 = 0$, $r_0 = g$, $d_0 = -r_0$;

Passo 1: se $\|r_0\| < \epsilon$

retornar $p = p^0$;

para $j = 0, 1, 2, \dots$

Passo 2: se $d_j^T B d_j \leq 0$

Encontre τ tal que $p = p^j + \tau d_j$ resolvendo o subproblema 3.9;

```
retornar p;
```

Passo 3: Fixar $\alpha_j = \frac{r_j^T r_j}{d_j^T B d_j}$;

Passo 4: Fixar $p^{j+1} = p^j + \alpha_j d_j$;

Passo 5: $se \ \|p^{j+1}\| \geq \Delta$

Encontre $\tau \geq 0$ tal que $p = p^j + \tau d_j$ satisfazendo $\|p\| = \Delta$;

```
retornar p;
```

Passo 6: Fixar $r_{j+1} = r_j + \alpha_j B d_j$;

Passo 7: $se \ \|r_{i+1}\| < \epsilon \|r_0\|$

retornar $p = p^{j+1}$;

Passo 8: Fixar $B_{j+1} = \frac{r_{j+1}^T r_{j+1}}{r_j^T r_j}$;

Passo 9: Fixar $d_{j+1} = r_{j+1} + B_{j+1}d_j$;

fim (pára).

O algoritmo 3.3.4 difere do algoritmo padrão GC em dois critérios extras de parada - as duas primeiras manifestações **se** dentro do loop **pára**. O primeiro **se** indica se a direção de busca atual d_j é uma direção de curvatura zero ou curvatura negativa em relação à B . O segundo leva à interrupção do

processo se p^{j+1} viola o limite da região de confiança. Em ambos os casos, um ponto final p é encontrado pela intersecção da direção de busca atual com o limite da região de confiança.

A inicialização $p^0 = 0$ é uma característica crucial do algoritmo. Depois da primeira iteração, temos

$$p^1 = \alpha_0 d_0 = \frac{r_0^T r_0}{d_0^T B d_0} d_0 = -\frac{g^T g}{g^T B g} g,$$

que é exatamente o ponto de Cauchy! Como cada iteração do método de gradientes conjugados reduz $q(\cdot)$, este algoritmo cumpre as condições necessárias para convergência global.

Outra propriedade crucial do método é que, a cada iteração, a norma de p^j é maior que a de seu antecessor. Esta propriedade é outra consequência da inicialização $p^0 = 0$. Sua implicação principal está na aceitabilidade para interromper a iteração assim que o limite da região de confiança seja alcançado, porque nenhuma iteração adicional que forneça um valor menor de q estará dentro da região de confiança. O teorema a seguir expõe formalmente estas propriedades.

Teorema 3.3.5. *A sequência de vetores gerados pelo Algoritmo 3.3.4 satisfaz*

$$0 = \|p^0\|_2 < \dots < \|p^j\|_2 < \|p^{j+1}\|_2 < \dots < \|p\|_2 \leq \Delta.$$

Dem.: Primeiro mostraremos que a sequência de vetores gerados pelo Algoritmo 3.3.4 satisfaz $(p^j)^T r_j = 0$ para $j \geq 0$ e $(p^j)^T d_j > 0$ para $j \geq 1$. O Algoritmo 3.3.4 calcula p^{j+1} recursivamente em termos de p^j , porém quando todos os termos desta recursão são escritos explicitamente, observamos que

$$p^j = p^0 + \sum_{i=0}^{j-1} \alpha_i d_i = \sum_{i=0}^{j-1} \alpha_i d_i,$$

desde que $p^0 = 0$. Multiplicando por r_j e aplicando a propriedade de subespaço expandido de GC teremos

$$(p^j)^T r_j = \sum_{i=0}^{j-1} \alpha_i d_i^T r_j = 0.$$

Uma prova por indução estabelece a relação $(p^j)^T d_j > 0$. Aplicando a propriedade de subespaço expandido novamente, obtemos

$$(p^1)^T d_1 = (\alpha_0 d_0)^T (r_1 + \beta_1 d_0) = \alpha_0 \beta_1 d_0^T d_0 > 0. \quad (3.14)$$

Fazemos agora a hipótese indutiva que $(p^j)^T d_j > 0$ e provaremos que esta implica $(p^{j+1})^T d_{j+1} > 0$. De (3.14), temos $(p^{j+1})^T r_{j+1} = 0$, e consequentemente temos

$$\begin{aligned} (p^{j+1})^T d_{j+1} &= (p^{j+1})^T (r_{j+1} + \beta_{j+1} d_j) \\ &= \beta_{j+1} (p^{j+1})^T d_j \\ &= \beta_{j+1} (p^j + \alpha_j d_j)^T d_j \\ &= \beta_{j+1} (p^j)^T d_j + \alpha_j \beta_{j+1} d_j^T d_j. \end{aligned}$$

Por causa da hipótese indutiva, a última expressão é positiva.

Agora provaremos o teorema. Se o algoritmo parar porque $d_j^T B d_j \leq 0$ ou $\|p^{j+1}\|_2 \geq \Delta$, então o ponto final p é escolhido fazendo $\|p\|_2 = \Delta$, que é o maior comprimento possível que qualquer ponto pode ter. Para cobrir todas as outras possibilidades no algoritmo devemos provar que $\|p^j\|_2 < \|p^{j+1}\|_2$ quando $p^{j+1} = p^j + \alpha_j d_j$ e $j \geq 1$. Observe que

$$\|p^{j+1}\|_2^2 = (p^j + \alpha_j d_j)^T (p^j + \alpha_j d_j) = \|p^j\|_2^2 + 2\alpha_j (p^j)^T d_j + \alpha_j^2 \|d_j\|_2^2.$$

Segue desta expressão e do nosso resultado intermediário que $\|p^j\|_2 < \|p^{j+1}\|_2$, assim nossa prova está completa. ■

3.3.5 Redução obtida pelo ponto de Cauchy

Nas discussões anteriores de algoritmos para resolução aproximada de subproblemas de região de confiança, enfatizamos repetidamente que convergência global depende da solução aproximada, a qual deve obter pelo menos redução suficiente na função modelo q como o ponto de Cauchy. Na realidade, uma fração fixa da redução de Cauchy basta. Começaremos obtendo uma estimativa da redução em q alcançada pelo ponto de Cauchy, e então usaremos esta estimativa para provar que a sucessão de gradientes $\{\nabla f_k\}$ gerada pelo algoritmo 3.3.1 ou tem um ponto de acumulação em zero ou

então converge para zero, dependendo se escolhemos o parâmetro η igual a zero ou estritamente positivo no algoritmo. Mostraremos um resultado de convergência para a versão do algoritmo 3.3.1.

Começaremos provando que o dog-leg e o algoritmo 3.3.4 produzem soluções aproximadas p_k do subproblema (3.3) que satisfazem a estimativa

$$q_k(0) - q_k(p_k) \geq c_1 \|\nabla f_k\| \min(\Delta_k, \frac{\|\nabla f_k\|}{\|B_k\|}), \quad (3.15)$$

para alguma constante $c_1 \in (0, 1]$. A presença de uma alternativa dada pelo mínimo em (3.15) é típica de métodos de região de confiança e surge por causa do limite da região de confiança. Note que quando Δ_k for o valor mínimo em (3.15), a condição lembra ligeiramente a primeira condição de Wolfe (ver [10]): a redução desejada no modelo é proporcional ao gradiente e ao tamanho do passo.

Veremos agora que o ponto de Cauchy p_k^C satisfaz (3.15), com $c_1 = \frac{1}{2}$.

Lema 3.3.6. *O ponto de Cauchy p_k^C satisfaz (3.15), com $c_1 = \frac{1}{2}$, isto é,*

$$q_k(0) - q_k(p_k^C) \geq \frac{1}{2} \|\nabla f_k\| \min\left(\Delta_k, \frac{\|\nabla f_k\|}{\|B_k\|}\right). \quad (3.16)$$

Dem.: Ver Nocedal & Wright [14].

Para satisfazer (3.15), nossa solução aproximada p_k só tem que alcançar uma redução que é pelo menos alguma fração fixa c_2 da redução alcançada pelo ponto de Cauchy. Enunciaremos esta observação como um teorema.

Teorema 3.3.7. *Seja p_k um vetor qualquer tal que $\|p_k\| \leq \Delta_k$ e $q_k(0) - q_k(p_k) \geq c_2(q_k(0) - q_k(p_k^C))$. Então p_k satisfaz (3.15) com $c_1 = \frac{c_2}{2}$. Em particular, se p_k é a solução exata p_k^* de (3.3), então satisfaz (3.15) com $c_1 = \frac{1}{2}$.*

Dem.: Ver Nocedal & Wright [14].

Note que dog-leg e o algoritmo (3.3.4) satisfazem (3.15) com $c_1 = \frac{1}{2}$, porque eles produzem soluções aproximadas p_k para a qual $q_k(p_k) \leq q_k(p_k^C)$.

3.3.6 Convergência para pontos estacionários

Para analisarmos convergência global de métodos de região de confiança, temos duas possibilidades: fixarmos o parâmetro η no algoritmo 3.3.1 igual a zero ou atribuímos a η um valor positivo pequeno. Quando $\eta = 0$, podemos mostrar que a sucessão $\{\nabla f_k\}$ de gradientes tem um limite no ponto zero. Para o teste de aceitação mais estrito com $\eta > 0$, que requer que a redução real de f seja pelo menos alguma pequena fração da diminuição prevista, temos o resultado: $\nabla f_k \rightarrow 0$.

Veremos resultados de convergência global para ambos os casos. Assumiremos que as Hessianas aproximadas B_k , em norma, são uniformemente limitadas, e que o conjunto de nível

$$\{x | f(x) \leq f(x_0)\} \quad (3.17)$$

é limitado. Para ampliar os resultados, permitiremos também que o comprimento da solução aproximada p^k de 3.3.4 exceda o limite da região de confiança, contanto que fique dentro de algum múltiplo fixo do limite; isso é,

$$\|p_k\| \leq \lambda \Delta_k, \text{ para alguma constante } \lambda \geq 1. \quad (3.18)$$

Os primeiros resultados são para o caso de $\eta = 0$.

Teorema 3.3.8. *Seja $\eta = 0$ no algoritmo 3.3.1. Suponhamos que $\|B_k\| \leq \beta$ para alguma constante β , que f seja continuamente diferenciável e limitada no conjunto de nível (3.17), e que toda solução aproximada p_k de 3.3.4 satisfaça as inequações (3.15) e (3.18), para constantes positivas c_1 e λ . Teremos então*

$$\lim_{k \rightarrow \infty} \inf \|\nabla f_k\| = 0. \quad (3.19)$$

Dem.: Ver Nocedal & Wright [14].

Para o caso de $\eta > 0$ temos o teorema a seguir.

Teorema 3.3.9. *Seja $\eta \in (0, \frac{1}{4})$ no algoritmo 3.3.1. Suponhamos que $\|B_k\| \leq \beta$ para alguma constante β , que f seja Lipschitz continuamente diferenciável e limitada no conjunto de nível (3.17), e que toda solução aproximada p_k de 3.3.4 satisfaça as inequações (3.15) e (3.18), para constantes positivas c_1 e λ . Teremos então*

$$\lim_{k \rightarrow \infty} \|\nabla f_k\| = 0. \quad (3.20)$$

Dem.: Ver Nokedal & Wright [14].

Capítulo 4

Um Método sem Derivadas Baseado em Interpolação Quadrática

Descreveremos neste capítulo um método proposto por M. J. D. Powell em 2002 [16] e aprimorado em 2003 [17] e 2004 [18, 19]. Este método é voltado a problemas de minimização, sem restrições, nos moldes do problema (2.1). Um modelo quadrático $Q \approx f$ pode ser requerido no começo de cada uma das iterações, o qual é usado em um processo de região de confiança para ajustar as variáveis. Quando Q é atualizado, o novo Q interpola f em m pontos, sendo o valor de $m = 2n + 1$ inicialmente recomendado. Só um ponto de interpolação é alterado em cada iteração. Assim, a quantidade de trabalho por iteração é somente da ordem de $(m + n)^2$, o que permite que n seja bastante grande.

4.1 O Modelo Quadrático e o Cálculo dos Pontos Iniciais

Aproximações quadráticas para a função objetivo são altamente úteis para obter uma taxa rápida de convergência em algoritmos iterativos para otimização sem restrição. Por outro lado, cada modelo quadrático tem $\frac{1}{2}(n+1)(n+2)$ parâmetros independentes e este número de cálculos de valor da função objetivo é proibitivamente caro em muitas aplicações com n grande. Então este método tenta construir modelos quadráticos satisfatórios com menos dados.

Como no capítulo 2, consideremos uma função quadrática $Q(x)$ dada por:

$$Q(x) = \frac{1}{2}x^T Gx + b^T x + c,$$

onde $x \in \mathbb{R}^n$, $G \in \mathbb{R}^{n \times n}$ simétrica, $b \in \mathbb{R}^n$ e $c \in \mathbb{R}$.

Neste método, no começo de uma iteração, o modelo quadrático $Q(x)$ tem que satisfazer somente m condições de interpolação

$$Q(x_i) = f(x_i), \quad i = 1, 2, \dots, m, \quad (4.1)$$

onde $f(x)$, $x \in \mathbb{R}^n$, é a função objetivo e m é o número de pontos de interpolação; as posições dos diferentes pontos x_i , $i = 1, 2, \dots, m$ são geradas a partir de x_0 . Requeremos $m \leq \frac{1}{2}(n+1)(n+2)$ e $m \geq n+2$, para que as equações (4.1) sempre forneçam algumas condições sobre a matriz de derivadas parciais de segunda ordem $G = \nabla^2 Q$. Lembremos que o modelo quadrático tem $\frac{1}{2}(n+1)(n+2)$ parâmetros independentes a serem determinados, isto é, $\frac{(n+1)n}{2}$ parâmetros de G , n de b e 1 de c , sugerimos $m = 2n+1$ pontos de interpolação, pois com esse número de pontos, conseguimos determinar c , o vetor b e n entradas da matriz G , forçando-a ser diagonal, o que facilita os cálculos, podendo assim n ser grande, os experimentos numéricos presentes no capítulo 5 reforçam nossa escolha.

Na primeira iteração, escrevemos o modelo quadrático na forma

$$Q(x_0 + d) = Q(x_0) + d^T \nabla Q(x_0) + \frac{1}{2} d^T \nabla^2 Q d, \quad d \in \mathbb{R}^n. \quad (4.2)$$

Se $m = 2n+1$, forçamos $\nabla^2 Q$ ser diagonal.

Quando o número de condições de interpolação (4.1) satisfaz $m \geq 2n+1$, os primeiros $2n+1$ pontos x_i , $i = 1, 2, \dots, m$, são escolhidos pelos vetores:

$$x_1 = x_0, \quad e \quad \left. \begin{array}{l} x_{i+1} = x_0 + \rho_{beg} e_i \\ x_{i+n+1} = x_0 - \rho_{beg} e_i \end{array} \right\} \quad i = 1, 2, \dots, n, \quad (4.3)$$

onde e_i é o i -ésimo vetor da base canônica em $\mathbb{R}^n$ e ρ_{beg} é o raio inicial da região de confiança (que será abordado na seção 4.2). Assim $Q(x_0)$, $\nabla Q(x_0)$ e os elementos da diagonal $(\nabla^2 Q)_{ii}$, $i = 1, 2, \dots, n$, são determinados univocamente pelas primeiras $2n+1$ equações (4.1). Alternativamente, quando m satisfaz $n+2 \leq m \leq 2n$, os pontos iniciais de interpolação são os primeiros m vetores (4.3). Segue que $Q(x_0)$, os primeiros $m - n - 1$ componentes de $\nabla Q(x_0)$ e

$(\nabla^2 Q)_{ii}$, $i = 1, 2, \dots, m - n - 1$, são definidos como anteriormente. Os demais elementos da diagonal de $\nabla^2 Q$ são um conjunto de zeros, assim como as outras componentes de $\nabla Q(x_0)$ tomam os valores $\{f(x_0 + \rho_{beg} e_i) - f(x_0)\} / \rho_{beg}$, $m - n \leq i \leq n$.

No caso $m > 2n + 1$, os pontos iniciais x_i , $i = 1, 2, \dots, m$, são escolhidos tais que as condições (4.1) também forneçam $2(m - 2n - 1)$ elementos fora da diagonal de $\nabla^2 Q$; o fator 2 é devido à simetria. Especificamente, para $i \in [2n + 2, m]$, os pontos x_i têm a forma

$$x_i = x_0 + \sigma_p \rho_{beg} e_p + \sigma_q \rho_{beg} e_q, \quad (4.4)$$

onde p e q são inteiros diferentes em $[1, n]$, e onde σ_p e σ_q são incluídos na definição:

$$\sigma_j = \begin{cases} -1, & f(x_0 - \rho_{beg} e_j) < f(x_0 + \rho_{beg} e_j) \\ +1, & f(x_0 - \rho_{beg} e_j) \geq f(x_0 + \rho_{beg} e_j), \end{cases} \quad j=1,2,\dots,n,$$

fazendo com que a escolha (4.4) leve a valores menores da função objetivo.

Desse modo o elemento $(\nabla^2 Q)_{pq} = (\nabla^2 Q)_{qp}$ é determinado pela equação (4.1), desde que toda função quadrática $Q(x)$, $x \in \mathbb{R}^n$, tenha a propriedade

$$\sigma_{beg}^{-2} \{Q(x_0) - Q(x_0 + \sigma_p \rho_{beg} e_p) - Q(x_0 + \sigma_q \rho_{beg} e_q) + Q(x_0 + \sigma_p \rho_{beg} e_p + \sigma_q \rho_{beg} e_q)\} = \sigma_p \sigma_q (\nabla^2 Q)_{pq}.$$

Por simplicidade, p e q são escolhidos na fórmula (4.4) da seguinte forma. Seja j a parte inteira do quociente $\frac{(i-n-2)}{n}$, donde $j \geq 1$ devido a $i \geq 2n + 2$; fixamos $p = i - n - 1 - jn$, que está no intervalo $[1, n]$, e permitimos q ter o valor $p + j$ ou $p + j - n$, a última escolha sendo feita no caso $p + j > n$. Consequentemente, se $n = 5$ e $m = 20$, por exemplo, há 9 pares $\{p, q\}$, gerados na ordem $\{1, 2\}$, $\{2, 3\}$, $\{3, 4\}$, $\{4, 5\}$, $\{5, 1\}$, $\{1, 3\}$, $\{2, 4\}$, $\{3, 5\}$ e $\{4, 1\}$. Todos os elementos fora da diagonal de $\nabla^2 Q$ que não são fornecidos por este procedimento são fixados como zero, o que completa a especificação do modelo quadrático inicial (4.2).

O método exige m pontos de interpolação. Dentre esses, determinamos o ponto x_{opt} como sendo o que fornece o menor valor de f até então calculado. Em seguida, devemos determinar $x_{new} = x_{opt} + d$, onde d é a solução aproximada do subproblema de região de confiança

$$\begin{aligned} & \text{Minimizar } Q(x_{opt} + d) \\ & \text{sujeita a } \|d\| \leq \Delta. \end{aligned} \quad (4.5)$$

De posse do novo ponto, alteramos o conjunto de pontos de interpolação, adicionando x_{new} no lugar de x_{MOVE} , ponto este que maximiza a distância $DIST = \|x_{MOVE} - x_{opt}\|$.

Utilizando esse novo conjunto de pontos de interpolação, atualizaremos o modelo quadrático Q_{old} para $Q_{new} = Q_{old} + D$, onde a função quadrática D é construída de forma que $\|\nabla^2 D\|_F^2 = \|\nabla^2 Q_{new} - \nabla^2 Q_{old}\|_F^2$ seja mínima, sujeita às restrições

$$D(x_i) = f(x_i) - Q_{old}(x_i), \quad i = 1, 2, \dots, m. \quad (4.6)$$

Estas restrições são equivalentes a $Q_{new}(x_i) = f(x_i)$, $i = 1, 2, \dots, m$.

O sucesso dessa atualização é devido à técnica sugerida pelo método simétrico de Broyden e Powell para atualizar $\nabla^2 Q$, quando as primeiras derivadas de f não estão disponíveis, método este descrito no capítulo 2.

A força desta técnica de atualização pode ser explicada considerando o caso em que a função objetivo f é quadrática. Se o erro $|f(x_{new}) - Q_{old}(x_{new})|$ é relativamente pequeno, então o modelo aproximou bem o novo valor de f , até mesmo se os erros das aproximações $\nabla^2 Q \approx \nabla^2 f$ forem significativos. Por outro lado, se $|f(x_{new}) - Q_{old}(x_{new})|$ é relativamente grande, então, satisfazendo $Q_{new}(x_{new}) = f(x_{new})$, a técnica de atualização deve melhorar a precisão do modelo significativamente. Resultados numéricos provam que estas alternativas são bem-vindas, pois fornecem excelente convergência da sequência de vetores de variáveis gerada pelo método, embora geralmente o erro $\|\nabla^2 Q - \nabla^2 f\|_F$ seja grande para todo Q .

4.2 O Subproblema de Região de Confiança

O método proposto por Powell caracteriza-se como um método de região de confiança, nos moldes dos algoritmos descritos na seção 3.3 desta dissertação.

A cada iteração devemos obter um passo d a partir de x_{opt} para o cálculo de x_{new} . Tal passo é obtido através da resolução do subproblema de Região de Confiança (4.5), que é resolvido através do método de gradientes conjugados truncado, vide ([8]). Observaremos mais atentamente os parâmetros envolvidos na resolução deste problema. Obtida a direção d , testamos se $\|d\| \geq \frac{1}{2}\rho$,

onde $\rho = \rho_{beg}$ na primeira iteração, podendo ser alterado se necessário nas demais; então calculamos a razão

$$\gamma = \frac{f(x_{opt}) - f(x_{opt} + d)}{Q(x_{opt}) - Q(x_{opt} + d)}$$

e atualizamos o raio da região de confiança Δ , sujeito a $\Delta \geq \rho$. O novo raio da região de confiança Δ_{new} será ρ ou Δ_{int} nos casos $\Delta_{int} \leq 1.5\rho$ ou $\Delta_{int} > 1.5\rho$, respectivamente, onde Δ_{int} é o valor

$$\Delta_{int} = \begin{cases} \frac{1}{2}\|d\|, & \gamma \leq 0.1, \\ \max\{\|d\|, \frac{1}{2}\Delta_{old}\}, & 0.1 < \gamma \leq 0.7, \\ \max\{2\|d\|, \frac{1}{2}\Delta_{old}\}, & \gamma > 0.7. \end{cases}$$

Após a atualização do raio de região de confiança, fixamos a variável inteira *MOVE* em 0 ou no índice da componente do ponto de interpolação que será alterado. Se *MOVE* > 0, modificamos Q de forma que interpole f em $x_{opt} + d$ em vez de x_{MOVE} . Se $f(x_{opt} + d) < f(x_{opt})$, então x_{opt} assume o valor de $x_{opt} + d$. Se $\gamma < 0.1$, alguns ajustes e testes podem ser feitos na tentativa de melhorar os resultados; caso contrário, iniciamos uma nova iteração do método de região de confiança.

Se $\|d\| < \frac{1}{2}\rho$, testamos se os três valores mais recentes de $\|d\|$ e $f - Q$ são pequenos. Se for este o caso, reduzimos ρ por um fator de 10 sujeito a $\rho \geq \rho_{end}$, e Δ pelo $\max[\frac{1}{2}\rho_{old}, \rho_{new}]$; senão, reduzimos Δ por um fator de 10 ou pelo limitante inferior ρ . Novamente podemos fazer uso de alguns ajustes e testes na tentativa de melhorar os resultados. Em ambos os casos, iniciamos outra iteração do método de região de confiança.

Todo este processo é feito até $\rho = \rho_{end}$, obtendo assim a solução do problema (2.1) com a precisão desejada.

Capítulo 5

Experimentos Numéricos

O método abordado no capítulo 4 está implementado no software NEWUOA, de autoria de Powell. Tanto a descrição do método, quanto aspectos da implementação computacional podem ser encontrados em [19]. O método foi programado e compilado em Fortran 77.

Com o objetivo de analisarmos o desempenho do método estudado, realizamos vários experimentos numéricos utilizando um PC, com processador AMD Athlon(tm) 64 3000+ 1.80 GHz 939P e 1,00 GB de memória RAM em Dual Channel 512 MB + 512 MB, utilizando como sistema operacional o Windows XP. O método foi testado com os problemas sugeridos por Moré, Garbow e Hillstom [12] e com um problema sugerido por Powell [19] chamado SPHRPTS.

5.1 Problemas de Moré, Garbow e Hillstom

Os primeiros testes aos quais o método foi submetido constituem uma coletânea de problemas organizada por Moré, Garbow e Hillstom [12]. Trata-se de um conjunto de 35 problemas de minimização irrestrita. As funções objetivo são soma de quadrados e diferenciáveis.

Em todos os problemas adotamos os mesmos parâmetros:

$$\rho_{beg} = |0, 2X(1)|;$$

$$\rho_{end} = 10^{-6};$$

$$Maxfun = 1,0 \times 10^6;$$

$$m = 2n + 1,$$

onde, ρ_{beg} é o tamanho inicial do raio da região de confiança, o qual é uma fração em módulo da primeira componente do vetor inicial, ou apenas 0.2 quando o vetor inicial x_0 tem todas as componentes nulas; ρ_{end} é o raio final da região de confiança; $Maxfun$ é o número máximo de avaliações de função; m é o número de pontos de interpolação e n , a dimensão do problema. A aproximação inicial x_0 usada para cada problema é a sugerida em [12].

prob	n	feval	tempo (s)	prob	n	feval	tempo (s)
1	2	161	0	19	11	1752	0,42
2	2	78	0	20	12	32797	8,37
3	2	36	0	21	100	73030	1008,76
4	2	395	0	22	100	146803	2000,34
5	2	74	0	23	100	65067	753,81
6	2	44	0	24	100	33937	398,82
7	3	193	0	25	100	152525	1786,15
8	3	118	0	26	100	11209	138,06
9	3	43	0	27	100	342130	2943,06
10	3	2061	0,06	28	100	790788	11651,64
11	3	778	0,06	29	100	752146	9180,26
12	3	278	0,01	30	100	2661	28,65
13	4	537	0,01	31	100	4915	52,37
14	4	521	0,01	32	100	423	2,50
15	4	269	0,01	33	100	1545	17,10
16	4	245	0,01	34	100	1603	17,26
17	5	4903	0,31	35	100	117300	1644,92
18	6	1781	0,12				

Tabela 5.1: Resultados da aplicação do método em problemas de [12]

A dimensão n adotada segue o valor fornecido em [12], para os problemas

1 a 20. A partir do problema 21, quando n pode ser variável, escolhemos $n = 100$. A tabela 5.1 mostra o número do problema, a dimensão n , o número de avaliações de função e o tempo de CPU em segundos.

Observando a tabela 5.1 notamos que o método convergiu em todos os 35 problemas, de maneira satisfatória. Para os problemas 1-9 de dimensões pequenas, o método convergiu instantaneamente, e para os problemas 10-19, a convergência ocorreu em menos de um segundo. Ressaltamos também que, para os problemas 24, 26, 30, 31, 32, 33, 34, todos de 100 variáveis, o método obteve convergência com poucas avaliações de função e em pouco tempo computacional. A maior quantidade de avaliações de função e o maior tempo computacional foram observados para os problemas 28 e 29, conhecidos como “*Discrete boundary value function*” e “*Discrete integral equation function*” respectivamente. Os problemas onde n é variável, ou seja, os de número 21 a 35, também foram testados para $m = \frac{1}{2}(n+1)(n+2)$ pontos de interpolação, apresentando tempo computacional inviável, reforçando que a escolha sugerida pelo autor ($m = 2n+1$) é a mais indicada se n não for pequeno.

5.2 Problema SPHRPTS

Este problema teste, conhecido como SPHRPTS, foi proposto por Powell [19]. Neste problema, n é par, e $\frac{n}{2}$ pontos têm que ser colocados na superfície de uma esfera unitária em $\mathbb{R}^3$, razoavelmente espaçados. Fazemos então que o k -ésimo ponto $p_k \in \mathbb{R}^3$ tenha as coordenadas

$$p_k = \begin{pmatrix} \cos x_{2k-1} \cos x_{2k} \\ \sin x_{2k-1} \cos x_{2k} \\ \sin x_{2k} \end{pmatrix}, \quad k = 1, 2, \dots, \frac{n}{2},$$

onde $x \in \mathbb{R}^n$ é ainda o vetor de variáveis. O problema consiste em minimizar a função

$$f(x) = \sum_{k=2}^{\frac{n}{2}} \sum_{l=1}^{k-1} \|p_l - p_k\|^{-2}, \quad x \in \mathbb{R}^n,$$

onde inicialmente os pontos p_k são igualmente espaçados no equador da esfera; o vetor x_0 tem as componentes $(x_0)_{2k-1} = \frac{4\pi k}{n}$ e $(x_0)_{2k} = 0$, $k = 1, 2, \dots, \frac{n}{2}$.

O método foi aplicado ao problema com $m = 2n+1$ pontos de interpolação e n variável, de acordo com a tabela 5.2. Usamos os parâmetros $\rho_{beg} = n^{-1}$ e $\rho_{end} = 10^{-6}$. Os resultados obtidos estão registrados na mesma. Em todos os casos obtivemos convergência, com número razoável de avaliações de função e tempo computacional satisfatório, levando em consideração as dimensões dos problemas.

$m = 2n + 1$			
n	feval	tempo (s)	Mínimo de f
20	2683	1,35	$2,50413597 \times 10^1$
40	6732	12,46	$1,33936978 \times 10^2$
80	30370	214,45	$6,72656911 \times 10^2$
160	125596	3927,00	$3,24089121 \times 10^3$

Tabela 5.2: Resultados para o problema SPHRPTS com $m = 2n + 1$

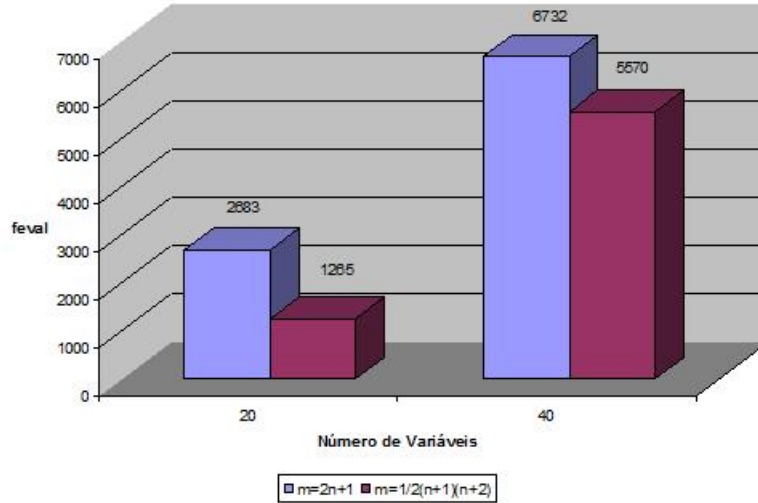


Figura 5.1: Comparação do número de avaliações de função para diferentes valores de m das tabelas (5.2) e (5.3)

$m = \frac{1}{2}(n+1)(n+2)$			
n	feval	tempo (s)	Mínimo de f
20	1265	7,62	$2,50413597 \times 10^1$
40	5570	469,25	$1,33936978 \times 10^2$
80	-	-	-
160	-	-	-

Tabela 5.3: Resultados para o problema SPHRPTS com $m = \frac{1}{2}(n+1)(n+2)$

Já a tabela 5.3 contém os dados da aplicação do método para $m = \frac{1}{2}(n+1)(n+2)$. Destacamos que nos testes com $n = 80$ e $n = 160$ o tempo computacional foi muito grande, passando de 5 horas, sem obter convergência, ou mesmo, atingir algum critério de parada. Os resultados

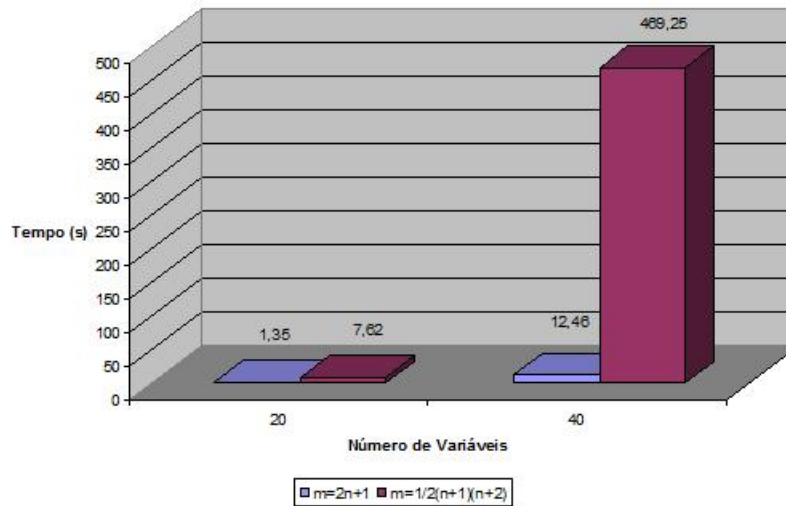


Figura 5.2: Comparação do tempo computacional gasto para os diferentes valores de m das tabelas (5.2) e (5.3)

das tabelas 5.2 e 5.3 estão resumidos nas figuras 5.1 e 5.2, onde representamos, para cada problema, o número total de avaliações de função realizado e o tempo total com os diferentes valores de m . Comparando os resultados mostrados em ambas as tabelas, quando $n = 20$ e $n = 40$, notamos que, na segunda, o número de avaliações de função é bem menor que na primeira,

porém o tempo computacional é bem maior. Ou seja, mesmo tendo mais pontos de interpolação, isto é, uma melhor aproximação de f pelo modelo quadrático, o tempo computacional é elevado e não notamos mudanças significativas no valor mínimo de f .

Com o objetivo de comparar o desempenho do método proposto por Powell com outro algoritmo que não fizesse uso de derivadas de f , mas trabalhasse com aproximações para estas derivadas, testamos o problema SPHRPTS e os problemas da seção 5.1 com o software GENCAN. Este software faz parte do projeto TANGO (Trustable Algorithms for Nonlinear General Optimization), um conjunto de rotinas em Fortran para otimização não linear, desenvolvido no Departamento de Matemática Aplicada da Universidade Estadual de Campinas e no Departamento de Informática da Universidade de São Paulo, sob a coordenação do Professor J. M. Martínez, da Unicamp. 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, voltado à resolução de problemas de minimização irrestrita ou em caixas, que não utiliza operações com matrizes, uma vez que baseia-se no método do Gradiente Espectral Projetado e no método de Gradientes Conjugados Truncado.

Para maiores detalhes sobre o GENCAN, consultar [1], [11] e o site do projeto TANGO (<http://www.ime.usp.br/egbirgin/tango/>).

Para os parâmetros utilizados foram adotados os seguintes valores:

$$gtype = 1$$

$$hptype = 4$$

$$precond = none$$

$$eps_{opt} = 1,0 \times 10^{-6}$$

$$max_{totit} = 1,0 \times 10^6$$

$$max_{totfc} = 5 \times max_{totit}$$

onde $gtype = 1$ define que as derivadas da função objetivo serão aproximadas por diferenças finitas; $hptype = 4$ indica que vai ser usada a atualização BFGS

para a matriz hessiana; *precond* = *none* define que não haverá nenhum pre-condicionamento no método dos gradientes conjugados; *epsopt* determina a tolerância para a norma do gradiente projetado da função objetivo; *maxtotit* indica o número máximo de iterações permitido e *maxtotfc*, o número máximo de avaliações de função.

Para a coletânea de problemas organizada por Moré, Garbow e Hillstom [12], adotamos os mesmos parâmetros acima, e os valores de n iguais aos da tabela 5.1. Como este conjunto de testes é constituído de 35 problemas de minimização irrestrita, onde as funções objetivo são somas de quadrados, de fácil diferenciabilidade, o desempenho do GENCAN foi muito bom, fazendo pouquíssimas avaliações de função, em baixo tempo computacional para todos os problemas.

Para o problema SPHRPTS obtivemos os seguintes resultados.

n	feval	tempo (s)	Mínimo de f
20	82448	2,41	$4,12499885 \times 10^1$
40	35853	5,19	$3,32499755 \times 10^2$
80	5000000	3490,38	$2,66499877 \times 10^3$
160	-	-	-

Tabela 5.4: Resultados para o problema SPHRPTS - GENCAN

Observamos que para $n = 20$ e 40 , o GENCAN convergiu, porém para $n = 80$, o mesmo parou ao atingir o máximo de avaliações de função e para $n = 160$, o tempo computacional foi muito grande, passando de 5 horas, sem obter convergência, ou mesmo, atingir algum critério de parada. Comparando os resultados obtidos na tabela 5.2 com os da tabela 5.4, notamos que a quantidade de avaliações de função feitas pelo GENCAN é bem superior, provavelmente devido ao fato deste último fazer aproximações por diferenças finitas. O tempo computacional para $n = 20$ e 80 também é superior, principalmente para $n = 80$ (aproximadamente 1528% maior). Além disso, em todos os casos, GENCAN obteve valores maiores para o mínimo da função. Estes resultados estão ilustrados nas figuras 5.3 e 5.4. Devido à discrepância entre os dados das tabelas 5.2 e 5.4, utilizamos o logaritmo dos dados de cada tabela, para evitar que as figuras ficassem distorcidas.

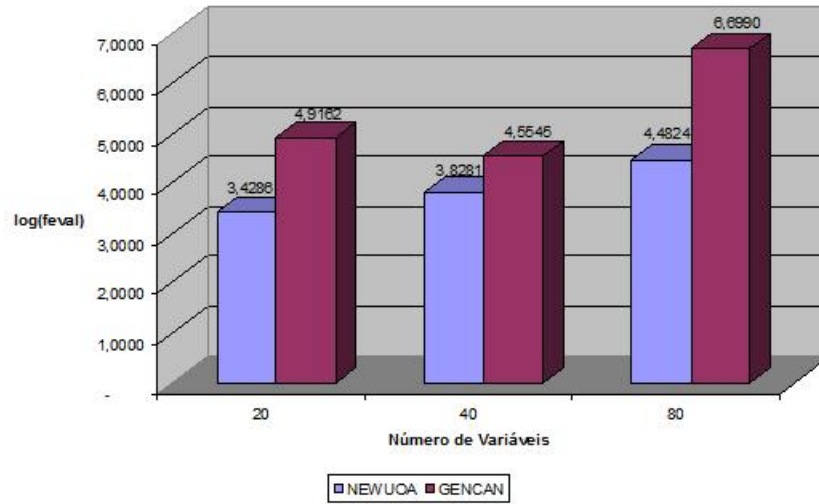


Figura 5.3: Comparação do número de avaliações de função entre NEWUOA e GENCAN

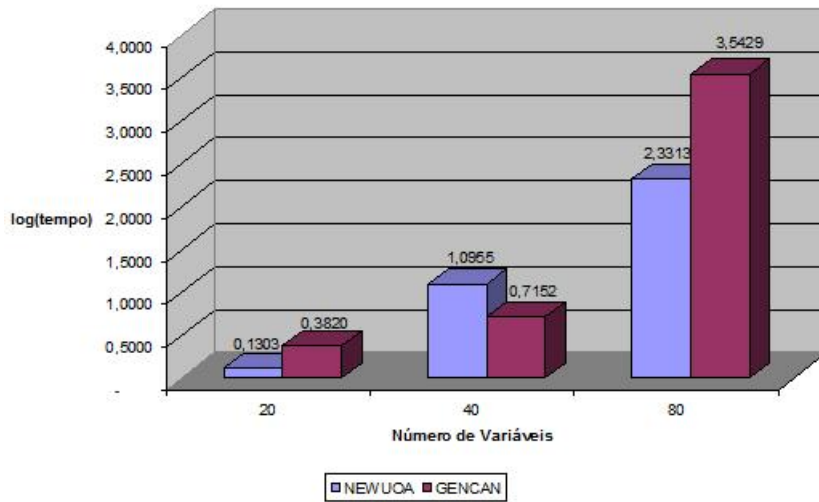


Figura 5.4: Comparação do tempo computacional gasto entre NEWUOA e GENCAN

Capítulo 6

Conclusões

Nosso interesse neste trabalho foi estudar um método para minimização de funções sem restrições, proposto por M. J. D. Powell, que não faz uso de derivadas, e é baseado em interpolação quadrática. Podemos dividir a estrutura desse método em três partes importantes: A primeira consiste na estratégia usada para escolha dos m pontos de interpolação. A segunda consiste na técnica utilizada na atualização do modelo quadrático, semelhante às usadas nos métodos quase-Newton, onde a Hessiana da função f é aproximada por outra matriz simétrica; no método proposto por Powell, a atualização do modelo quadrático é baseada na minimização da norma de Frobenius da diferença das Hessianas do modelo quadrático velho e novo. A terceira trata do subproblema de região de confiança utilizado na obtenção de um passo d usado para determinarmos um novo ponto de interpolação.

Devido à grande importância da técnica de atualização do modelo quadrático utilizada no método proposto por Powell, direcionamos parte significativa dos estudos aos métodos quase-Newton, fazendo principalmente uma análise das técnicas de atualização que estes utilizam. Muita atenção também foi dada ao subproblema de região de confiança, levando-nos a aprofundar os estudos sobre métodos de região de confiança e suas estratégias.

Analisando o desempenho computacional do método observamos que este mostrou-se bem robusto, já que convergiu para todos os problemas testados. O tempo computacional em todos os casos também foi satisfatório, considerando que não é usada nenhuma derivada da função objetivo.

Primeiramente os testes realizados faziam parte da coletânea de problemas sugeridos por Moré, Garbow e Hillstom. Neste caso, os resultados mostraram a robustez e eficiência do método. Em seguida, abordamos o

problema SPHRPTS, proposto por Powell. Analisamos o comportamento do método, variando a quantidade de pontos de interpolação, e concluímos que utilizar uma quantidade inferior de pontos ($m = 2n + 1$) faz com que o método obtenha convergência em tempo computacional reduzido e com precisão ótima. Isso acontece pois com $m = 2n + 1$ pontos forçamos a matriz Hessiana do modelo quadrático ser diagonal, possibilitando uma maior quantidade de avaliações de função em menos tempo. Por outro lado, usando $m = \frac{1}{2}(n+1)(n+2)$ pontos, a aproximação da Hessiana de f é feita de forma mais eficiente, e o método necessita de menos avaliações de função para obter convergência, porém teremos a matriz Hessiana do modelo quadrático cheia, o que reflete no tempo computacional, que passa a ser muito maior que com menos pontos de interpolação.

Com o objetivo de comparar o desempenho do método proposto por Powell, sem derivadas e baseado em interpolação polinomial, com outro algoritmo que também tem a opção de não usar derivadas, mas aproximando gradiente da função objetivo por diferenças finitas, o problema SPHRPTS também foi testado utilizando o GENCAN, software este que não utiliza operações com matrizes e baseado no método de Gradientes Conjugados Truncado e de Gradiente Espectral Projetado. Comparando ambos os softwares aplicados ao problema SPHRPTS, verificamos que o método proposto por Powell apresenta vantagens em relação ao GENCAN em todos os itens analisados: avaliações de função, tempo computacional e valor mínimo obtido para a função objetivo, na maior parte dos casos. Também submetemos o GENCAN à coletânea de testes de Moré, Garbow e Hillstom, que devido principalmente à fácil diferenciabilidade dos problemas, obteve excelente desempenho em todos os casos.

Conseguimos neste trabalho mostrar a aplicabilidade de um método sem derivadas (conhecido como *derivative – free*) para problemas onde o cálculo destas pode ser fácil, ou não muito. Fica a intenção de analisar esses métodos com uma gama maior de problemas testes, explorando principalmente o desempenho da proposta de Powell comparada com problemas não diferenciáveis.

Referências Bibliográficas

- [1] Birgin, E. G. & Martínez, J. M., “Large-Scale Active-Set Box-Constrained Optimization Method with Spectral Projected Gradients”, *Computacional Optimization and Applications* 23, pp. 101-125, 2002.
- [2] Broyden, C. G., “The coverage of a class of double-rank minimization algorithms”, Partes I e II, *J.I.M.A.* 6, pp. 76-90, 222-236, 1970.
- [3] Dennis, J. E., Jr. & Moré, J. J., “Quasi-Newton Methods, Motivation and Theory”, *SIAM Review* 19, 1977.
- [4] Dennis, J. E. & Schnabel, R. B., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, 1983.
- [5] Fletcher, R., “A new approach to variable metric algorithms”, *Comput. J.* 13, pp. 317-322, 1970.
- [6] Friedlander, A., *Elementos de Programação Não-Linear*, Editora da UNICAMP, 1994.
- [7] Goldfarb, D., “A family of variable metric methods derived by variational means”, *Math. Comp.* 24, pp. 23-26, 1970.
- [8] Kelley, C.T., *Iterative Methods for Linear and Nonlinear Equations*, Society for Industrial and Applied Mathematics (SIAM), 1995.
- [9] Luenberger, D. G., *Linear and Nonlinear Programming*. 2^a ed., Addison - Wesley Publishing Company, 1986.
- [10] Martínez, J. M. & Santos, S. A., *Métodos Computacionais de Otimização*, Sociedade Brasileira de Matemática, 1995.

- [11] Martínez, J. M., *Otimização Prática Usando o Lagrangiano Aumentado*, Opúsculo, DMA - IMECC - UNICAMP, 2006.
- [12] Moré, J.J., Garbow, B.S. & Hillstom, K.E., "Testing Unconstrained Optimization Software", ACM Transactions on Mathematical Software, Vol. 7, No.1, pp. 17-41, 1981.
- [13] Nelder, J.A. & Mead, R., "A Simplex Method for Function Minimization", The Computer Journal 7, pp. 308-313, 1965.
- [14] Nocedal, J. & Wright, S. J., *Numerical Optimization*, Springer, 1999.
- [15] Powell, M. J. D., "A new algorithm for unconstrained optimization", in *Nonlinear Programming*, J. B. Rosen, O. L. Mangasarian, and K. Ritter, eds., Academic Press, New York, pp. 31-65, 1970.
- [16] Powell, M. J. D., "UOBYQA: Unconstrained Optimization by Quadratic Approximation", Math. Programming, Vol. 92, pp. 555-582, 2002.
- [17] Powell, M. J. D., "On Trust Region Methods for Unconstrained Minimization Without Derivatives", Math. Programming, Vol. 97, pp. 605-623, 2003.
- [18] Powell, M. J. D., "On the Use of Quadratic Models in Unconstrained Minimization Without Derivatives", Optim. Meth. Software, Vol. 19, pp. 399-411, 2004.
- [19] Powell, M. J. D., "The NEWUOA Software for Unconstrained Optimization Without Derivatives", Technical Report DAMTP 2004/ NA08, Department of Applied Mathematics and Theoretical Physics, Centre for Mathematical Sciences, Wilberforce Road, Cambridge CB3 0WA, England, 2004. Disponível no endereço: http://www.damtp.cam.ac.uk/user/na/NA_papers/NA2004_08.pdf.
- [20] Shanno, D. F., "Conditioning of quasi-Newton methods for function minimization", Math. Comp. 24, pp. 647-657, 1970.
- [21] Steihaug, T., "The conjugate gradient method and trust regions in large scale optimization", SIAM Journal on Numerical Analysis, 20, pp. 626-637, 1983.