DISSERTAÇÃO DE MESTRADO

UM MÉTODO DE REGIÃO DE CONFIANÇA PARA MINIMIZAÇÃO IRRESTRITA SEM DERIVADAS

Liliana Jiménez Urrea
Aluna

Prof. Dra. Véra Lucia da Rocha Lopes
Orientadora

DMA - IMECC - UNICAMP
Novembro de 2008

UM MÉTODO DE REGIÃO DE CONFIANÇA PARA MINIMIZAÇÃO IRRESTRITA SEM DERIVADAS

Este exemplar corresponde à redação final da dissertação devidamente corrigida e defendida por Liliana Jiménez Urrea e aprovada pela comissão julgadora.

Campinas, 14 de Novembro de 2008.

Profa. Dra. Véra Lucia da Rocha Lopes
Orientadora

Banca Examinadora

Profa. Dra. Véra Lucia da Rocha Lopes
Profa. Dra. Maria Ap. Diniz Ehrhardt
Prof. Dr. Paulo A. Valente Ferreira

Dissertação apresentada ao Instituto de Matemática, Estatística e Computação Científica, IMECC - UNICAMP, como requisito parcial para obtenção do título de MESTRE em MATEMÁTICA APLICADA.

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP
Bibliotecária: Crisllene Queiroz Custódio – CRB8a 162/2005

Jiménez Urrea, Liliana
J564m Um método de região de confiança para minimização irrestrita sem derivadas / Liliana Jiménez Urrea -- Campinas, [S.P. : s.n.], 2008.

Orientador : Vera Lucia da Rocha Lopes
Dissertação (Mestrado) - Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Otimização irrestrita. 2. Interpolação. 3. Lagrange, Funções de. 4. Métodos sem derivadas. 5. Método de região de confiança. I. Lopes, Vera Lucia da Rocha. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.

(cqc/imecc)

Título em inglês: On the region method for unconstrained minimization without derivatives

Palavras-chave em inglês (Keywords): 1. Unrestricted optimization. 2. Interpolation. 3. Functions, Lagrangian. 4. Derivative-free methods. 5. Trust-region method.

Área de concentração: Otimização

Titulação: Mestre

Banca examinadora: Profa. Vera Lucia da Rocha Lopes (IME-UNICAMP)
Profa. Maria Aparecida Diniz Ehrhardt
Prof. Paulo A. Valente Ferreira

Data da defesa: 14/11/2008

Programa de Pós-Graduação: Mestrado em Matemática Aplicada

Dissertação de Mestrado defendida em 14 de novembro de 2008 e aprovada

Pela Banca Examinadora composta pelos Profs. Drs.

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

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

Prof. (a). Dr (a). PAULO AUGUSTO VALENTE FERREIRA

Resumo

Neste trabalho apresentamos métodos de minimização irrestrita, de uma função objetivo F de várias variáveis, que não fazem uso nem do gradiente da função objetivo - métodos *derivative-free*, nem de aproximações do mesmo. Nosso objetivo básico foi estudar e comparar o desempenho de métodos desse tipo propostos por M. J. D. Powell, que consistem em aproximar a função F por funções quadráticas - *modelos quadráticos* - e minimizar tal aproximação em regiões de confiança. Além do algoritmo de Powell de 2002 - $UOBYQA$ - são testados: uma variante dele, na qual utilizamos a escolha de alguns parâmetros, por nós estabelecida, e também a nova versão de $NEWUOA$, proposta por Powell em 2006. Todos os testes foram realizados com problemas da coleção de Hock-Schittkowski. São comparados os resultados numéricos obtidos pelos métodos de Powell: entre eles mesmos e também entre eles e um método de busca padrão de autoria de Virginia Torczon, o qual define, em cada iteração, um conjunto padrão de direções de busca a partir do ponto atual, procurando melhores valores para F.

Abstract

In this work we study numerical methods to solve problems of nonlinear programming without constraints, which do not make use, neither of the gradient of the objective function, nor of approaches to it. A method that consists on the approximation of the function F by a quadractic model, due to Powell (2002), *UOBYQA*, and a variant of this method were implemented. A new version of the *NEWUOA*, introduced by Powell in 2006, was also implemented. Besides the Powell algorithm, commentaries of the implementations are done. Numerical tests of such implementations with problems of the Hock-Schittkowski collection, are made at the end of the work. There are also comparisons of the Powell methods among themselves, and also a comparison among the Powell methods with a pattern search method, which looks for the improvement of the value of the objective function throughout a set of directions, depending on the current point. Such a method is due to Virginia Torczon.

Ao Tulio Emiro, o grande e
eterno amor de minha vida,
que me faz acreditar: "Sou uma
figura perfeita e corajosa. Porém, ...
tenho amor e muita sorte...sou feliz,
tenho tudo o que peço, porque
Deus sempre está do meu lado."

Agradecimentos

Gostaria muito de agradecer a todas as pessoas que me ajudaram, tanto no Brasil, quanto na Colômbia. Todos de alguma forma contribuíram nesta caminhada, e fizeram com que eu pudesse chegar até aqui. Infelizmente, impossível será destacar todos os amigos, familiares, professores, colegas, neste pequeno agradecimento, mas gostaria que soubessem o quanto foram importantes para mim.

Aos meus pais, gostaria que soubessem a importância de sempre estarem presentes em minha vida. À minha mãe, Ana Cecilia, sempre me apoiando e me dando forças para seguir neste caminho e ao meu pai, Enrique, que com toda certeza está, de lá do Céu, me abençoando e olhando por mim. Ao meu filho, Carlos, que amo muito, agradeço pela paciência de conviver durante todo esse tempo sem minha presença, encarando tudo com muita maturidade.

Ao Tulio, meu companheiro, amigo, namorado, colega, meu grande amor, por todo apoio, paciência, e principalmente por estar sempre ao meu lado nos momentos mais árduos, me mostrando que a vida, além de ser maravilhosa, foi feita para ser intensamente vivida.

Agradeço às minhas irmãs, Nelly, Lucy e Nancy, pela força depositada. Aos meus sobrinhos, que de um modo geral me ajudaram a superar todas as expectativas. À minha família, em especial minhas tias: María, Evelia e Uva, pela preocupação e pela ajuda nas horas difíceis. À minha Madrinha, pe-

las orações e apoio, mesmo com a tristeza da despedida do meu querido país.

Não posso deixar de agradecer à minha mãe do Brasil, Véra, por ter aceito me orientar, desenvolvendo um excelente trabalho comigo. Além dos muitos conselhos, paciência e confiança, que com certeza, ficarão marcados em meu coração para sempre.

À toda a equipe de professores da Pós-Graduação em Matemática Aplicada do *IMECC*, em especial aos professores Aurélio e Cheti por terem sido verdadeiros amigos e pelas sugestões valiosíssimas. À Sandra pela paciência nas explicações, à Professora Valéria na colaboração de grande valia na conclusão deste trabalho. Ao professor Paulo da *FEEC* por ter aceito participar da banca examinadora, e pelos seus comentários, que enriqueceram este trabalho. E principalmente, agradeço à *UNICAMP* pela oportunidade de estudo a mim proporcionada, me permitindo adquirir grande conhecimento.

Agradeço a todos os meus amigos do Brasil, em especial Linda (Rosi e Nena) e Miguel pelos momentos inesquecíveis, à Cristiano e sua esposa Gláucia por toda ajuda. Não posso deixar de agradecer à Mael e ao seu namorado João, ao Rodrigo pelas ótimas dicas, ao querido amigo Washington, pelo cafés compartilhados. Por fim, agradeço à querida amiga Viviana da Argentina. Ao Jairo e sua esposa Angela, que me apoiaram nos momentos difíceis e ao Hugo e sua esposa Margarita, ao Pablo e Pedro, todos da Colômbia, meus sinceros muito obrigada, pois todos, de onde quer que estivessem, tenho certeza que torceram muito por mim.

Aos funcionários do *IMECC*, em particular ao Ednaldo, à Cidinha, à Tânia e à Dona Fátima, pelo apoio que me deram e, em especial, à Josefa.

Finalmente, agradeço a Deus e à Virgem Santíssima, por terem me dado a vida e as muitas oportunidades que fazem de mim uma pessoa feliz, e que acredita em um amanhã melhor.

CONTEÚDO

ÍNDICE DE TABELAS

ÍNDICE DE FIGURAS

Símbolos usados

$\Re$	Números reais
$\|.\|$	Norma Euclidiana
Π	Conjunto dos polinômios quadráticos
m	Número de pontos de interpolação
$x_1 = x_{beg}$	Ponto inicial
x_i	$i = 1, 2, \cdots, m$, pontos de interpolação
Ω , N	Ω e $N \subset \Re^n$, regiões de confiança
$x_k = x_\Delta$	Elemento calculado na iteração da região de confiança Ω
$x_k = x_Q$	Elemento calculado na iteração da região de confiança N
$x_k + s_k$	Solução aproximada do problema
ρ_{beg}	Raio inicial da região de confiança
ρ_{end}	Raio final da região de confiança
Q	Modelo quadrático, aproximação da função objetivo F
g_Q	Gradiente do modelo quadrático Q
H_Q	Hessiana do modelo quadrático Q
ℓ_j	j-ésima função de Lagrange
δ_{ij}	Delta de Kronecker
C_k	Matriz geradora do método de busca padrão
P_k	Padrão de direções coordenadas
Bc_k^i	Direção do passo
s_k^i	O passo $s_k^i = \Delta_k Bc_k^i$, com Δ_k, o tamanho do passo
$O(.)$	$f = O(g) \equiv \|f(x)\| \leq C \|g(x)\|, \ \forall x$

CAPÍTULO 1

Introdução

Se queres que o futuro seja diferente
do passado deves planejar o presente.
L.J.

Em otimização, um problema se diz irrestrito se seus parâmetros podem assumir quaisquer valores. Para o caso de nosso interesse neste trabalho, minimizamos uma função objetivo de várias variáveis, sobre todo o espaço $\Re^n$. A formulação matemática é:

$$\underset{x}{\text{minimizar}} \quad F(x) \quad , \tag{1.1}$$

onde $x \in \Re^n$ é uma n-upla de números reais $(x_1, x_2, \cdots, x_n)$, com $n \geq 1$ e $F : \Re^n \to \Re$, função continuamente diferenciável.

Nosso interesse e motivação para estudar algoritmos que não fazem uso de derivada - *derivative-free* - para resolver este problema é a grande demanda de profissionais de várias áreas por tais ferramentas. Muitas vezes, calcular o valor da função $F(x)$ dado o vetor x, é muito caro; outras vezes, os valores das derivadas de F em x não são fáceis de encontrar, ou porque $F(x)$ resulta de algum fenômeno físico ou químico, ou, mais comumente, porque este é o resultado de uma complexa simulação de um computador. Este fato pode ser visto, por exemplo, nas aplicações apresentadas em *On trust region methods for unconstrained minimization without derivative* [16]. Em resumo,

a quantidade e a importância dos problemas que podem e devem ser resolvidos por esses métodos, serviram de motivação para estudarmos métodos *derivative-free* para minimização irrestrita.

Dentre todos os métodos *derivative-free* conhecidos encontramos os métodos de *busca direta*, os quais, além de não fazer uso explícito do gradiente da função F, não fazem uso de aproximações do mesmo, e tampouco de aproximações da função F, por exemplo por polinômios interpolantes. Estes métodos determinam, entre um conjunto de valores da função F em possíveis soluções, qual é o melhor valor, comparado com um valor de F já considerado como melhor, e conseqüentemente determinam a melhor solução do problema. A estratégia consiste em fazer comparações de cada um destes valores com o melhor valor obtido até o momento. Esta estratégia não necessita avaliar a função F em todas as soluções possíveis, visto que no momento no qual encontra-se um valor melhor, a solução nova é definida e um novo conjunto de valores é obtido [9].

Dentre os métodos de *busca direta*, podemos mencionar: o método simplex de Nelder e Mead [11], o método de Hookes e Jeeves [9], e os algoritmos de busca direta introduzidos por Dennis e Torczon [6, 23]. Estes métodos são baseados em uma geometria padrão pré-definida e usam essa ferramenta para decidir quando a função objetivo pode ser avaliada. Uma vantagem importante destes métodos é que eles não precisam da suavidade da função objetivo.

Neste trabalho voltamos nossa atenção a métodos de região de confiança, quando uma aproximação para $F(x)$, $x \in \Re^n$, é construída e o próximo vetor de variáveis é gerado, usualmente procurando o mínimo da aproximação em uma região conveniente de $\Re^n$. A aproximação usada é chamada *modelo quadrático*, por se tratar de uma aproximação por polinômios quadráticos. Nesses algoritmos é necessário, a cada iteração, a resolução de subproblemas da forma:

$$\begin{array}{ll} \text{minimizar} & Q(x_k + s) \\ s \in & \Omega \end{array} \tag{1.2}$$

onde k é um inteiro em $[1, \widehat{m}]$ com $\widehat{m}$ no intervalo $[n+1, m]$, m definido por $\frac{1}{2}(n+1)(n+2)$ e o conjunto $\Omega = \{x : \|s\| \leq \Delta\}$ a região de confiança, onde Δ é algum parâmetro positivo chamado raio da região de confiança Ω.

Usamos o sofware $UOBYQA$ de Powell, o qual é um algoritmos para

minimização irrestrita sem derivadas, que constróem modelos quadráticos por interpolação em valores da função objetivo, para uma escolha conveniente dos pontos de interpolação. Utilizamos funções de Lagrange porque elas têm algumas propriedades importantes e úteis. Em especial, elas mostram se uma troca de um ponto de interpolação preserva a não singularidade das equações de interpolação e fornecem uma cota de erro para o modelo quadrático. Além disso, elas podem ser atualizadas quando um ponto de interpolação é trocado. Em *UOBYQA: unconstrained optimization by quadratic approximation* [17], se propõe um novo algoritmo para minimização irrestrita, dada a função F, formando modelos quadráticos por interpolação, onde cada modelo é definido por $\widehat{m}$ valores da função. Em *On the Lagrange function of quadratic models that are defined by interpolation* [16], é provado que a cota de erro pode controlar ou ajustar o raio da região de confiança, de forma a se obter excelentes propriedades de convergência num algoritmo para problemas de minimização irrestrita.
Esses métodos, do tipo *derivative-free*, são aqueles que usam ferramentas de *interpolação-aproximação* introduzidos por Winfield [26, 28], e por Powell [13, 14]. Nesses métodos, o modelo polinomial (linear ou quadrático) de aproximação da função objetivo F, é construído e minimizado, no contexto duma região de confiança [3]. Um fato comum aos algoritmos desse tipo é que eles constróem uma base conveniente do espaço $\Re^n$, e geram o modelo por uma construção de uma combinação linear dos polinômios da base. Em particular, interpolam valores conhecidos da função F. Powell escolhe para seu trabalho, uma base formada por polinômios de Lagrange [5].

Powell desenvolveu um novo sofware em Fortran 77 chamado $NEWUOA$, para a resolução deste problema, considerando $m = 2n + 1$ pontos de interpolação. Esse sofware descreve alguns resultados numéricos que testam a eficiência e a exatidão na minimização irrestrita sem derivadas de funções de 320 variáveis. Ele sugere uma variação da técnica para manter o sistema de equações de interpolação não singular. Detalhes deste software e suas técnicas podem ser encontrados no artigo [19].

Neste trabalho fizemos a seguinte abordagem:

1. Estudamos os métodos propostos por Powell [16, 17, 18] e Virginia Torczon [24].

2. Estudamos a estrutura do modelo quadrático Q gerado pelo algoritmo, que é construído por interpolação em um número definido de pontos.

Esses pontos são determinados de uma forma particular, considerando valores da função F.

3. Introduzimos em $UOBYQA$ [17], uma modificação que consiste em tomar $\rho = 0.2$ parâmetro fixo, $M = |0.5F(x)|$ cota de erro, e o número de pontos de interpolação dado por $m = \frac{1}{2}(n+1)(n+2)$.

4. Realizamos os testes numéricos com os algoritmos $UOBYQA$ de Powell [17], o algoritmo $UOBYQA_{mod}$ (algoritmo modificado de Powell) e o algoritmo $NEWUOA$, com os problemas apresentados em [21] e [8].

5. Comparamos os resultados numéricos do algoritmo $UOBYQA_{mod}$, com os resultados obtidos pelo método de busca padrão de Virginia Torczon [24], o qual implementamos em Fortran 90, definindo em cada iteração um conjunto padrão de direções de busca no ponto atual em procura melhores valores para F.

Tomando como base o artigo de Powell de 2002 [17] para desenvolver a abordagem acima, os artigos citados foram usados da seguinte forma:

1. Powell 2003 [18], para gerar os pontos de interpolação, inicializar e atualizar as funções de Lagrange, e atualizar o modelo quadrático.

2. Powell 2002 [17], para construir o modelo quadrático, para solucionar os dois subproblemas:

   - Minimizar o modelo quadrático dentro de uma região de confiança Ω com raio Δ.
   - Maximizar a j-ésima função de Lagrange dentro de uma região de confiança N com raio ρ.

   e para atualizar Δ e ρ.

3. Powell 2001 [16], para preservar a não singularidade do sistema $Q(x_i) = F(x_i),\ i = 1, 2, \ldots, m$, utilizando as propriedades das funções de Lagrange, e para introduzir uma forma de construir uma cota de erro para o modelo quadrático.

Organizamos este trabalho em Capítulos. No Capítulo 2, apresentamos alguns conceitos básicos importantes para o desenvolvimento deste trabalho. No Capítulo 3, apresentamos detalhes do método baseado no algoritmo *UOBYQA Unconstrained Optimization BY Quadratic Approximation* [17], onde um novo vetor de variáveis é calculado minimizando o modelo atual

dentro duma região de confiança. Técnicas são descritas para ajustar o raio da região de confiança, e para a escolha das posições dos pontos de interpolação que mantêm a não singularidade. Descrevemos ainda a atualização dos modelos quadráticos. A partir disso, geramos o ponto x_Δ pelo método de Moré e Sorensen [10], que fornecerá uma melhor aproximação para a solução do problema (1.1).

No Capítulo 4, apresentamos o método de Virginia Torczon [24], um método iterativo que soluciona o problema (1.1), que não faz uso do gradiente da função F e tampouco de aproximações da mesma. O método define, em cada iteração, um conjunto padrão de direções de busca no ponto atual e procura melhores valores para F, ao longo de algumas delas, sustentado em um critério de decréscimo simples.

No Capítulo 5, fazemos uma análise dos resultados obtidos nos problemas numéricos testados.

O Capítulo 6 contém algumas conclusões, alguns agradecimentos e algumas propostas de trabalhos futuros.

CAPÍTULO 2

Conceitos fundamentais

2.1 Introdução

Neste capítulo apresentamos alguns fatos básicos sobre os problemas de otimização, que são necessários ao longo deste trabalho. Serão apresentados os resultados de existência de soluções e as condições de otimalidade para problemas irrestritos, os quais são conceitos que consideramos importantes no estudo de qualquer problema de otimização. Além disso, mostraremos a idéia geral dos métodos de aproximação, apresentando a técnica de interpolação polinomial e o algoritmo geral do método de região de confiança.

Sejam dados um conjunto $D \subset \Re^n$ e uma função $F : D \to \Re^n$. O problema principal a ser considerado neste trabalho é achar um minimizador de F no conjunto D. Este problema de programação não linear será escrito como

$$\begin{array}{c} \text{minimizar} \quad F(x) \\ x \in D \end{array} . \tag{2.1}$$

O conjunto D será chamado de *conjunto viável* do problema, os pontos de D serão chamados de *pontos viáveis*, e F será chamada de *função objetivo*.

Definição 2.1.1 (Métodos derivative-free)**.** Um método *derivative-free* é um método iterativo usado para resolver um problema de otimização, se ele não faz uso do gradiente da função F.

2.2 Classes de soluções

Definição 2.2.1 (Minimizador local)**.** Seja $F : D \to \Re$. Um ponto $x^* \in D$, é um minimizador local de F sobre D se existe $\epsilon > 0$ tal que

$$F(x^*) \leq F(x),$$

$\forall x \in D$, tal que $\|x - x^*\| < \epsilon$.

Definição 2.2.2 (Minimizador local estrito)**.** Seja $F : D \to \Re$. Um ponto $x^* \in D$, é um minimizador local estrito de F sobre D se existe $\epsilon > 0$ tal que

$$F(x^*) < F(x),$$

$\forall x \in D$ com $x \neq x^*$, tal que $\|x - x^*\| < \epsilon$.

Definição 2.2.3 (Minimizador global)**.** Seja $F : D \to \Re$. Um ponto $x^* \in D$, é um minimizador global de F sobre D se

$$F(x^*) \leq f(x), \quad \forall x \in D.$$

Definição 2.2.4 (Minimizador global estrito)**.** Seja $F : D \to \Re$. Um ponto $x^* \in D$, é um minimizador global estrito de F sobre D se

$$F(x^*) < F(x), \quad \forall x \in D \text{ com } x \neq x^*.$$

Proposição 2.2.1 (Condição de otimalidade)**.** *Se x^* é um mínimo local de F sobre D, então*

$$\nabla F(x^*)^T (x - x^*) \geq 0 \quad \forall x \in D.$$

Definição 2.2.5 (Ponto estacionário)**.** Um ponto x^* que satisfaz a condição de otimalidade (2.2.1) chama-se ponto estacionário do problema (2.1).

Definição 2.2.6 (Valor ótimo)**.** Dizemos que $\bar{v} \in [-\infty, +\infty)$ definido por

$$\bar{v} = \underbrace{inf}_{x \in D} F(x)$$

é o valor ótimo do problema (2.1).

Uma função pode admitir vários minimizadores globais, mas o valor ótimo (global) da função objetivo nesses minimizadores é, sempre o mesmo.

2.3 Condições de otimalidade para problemas sem restrições

Quando $D = \Re^n$, dizemos que o problema (2.1) é *irrestrito*, e quando $D \neq \Re^n$ falamos de *otimização com restrições*. Em nosso caso consideramos o problema irrestrito e estudaremos as condições que devem ser satisfeitas quando um ponto x^* é um minimizador (local) do problema (2.1). Condições deste tipo se chamam *condições necessárias de otimalidade*. Também estudaremos as condições que garantem que um ponto dado é minimizador local do problema. As condições deste último tipo se chamam *condições suficientes de otimalidade*.

Teorema 2.3.1 (Condição necessária de primeira ordem)**.** *Suponhamos que a função $F : \Re^n \to \Re$ seja diferenciável no ponto $x^* \in \Re^n$. Se x^* é um minimizador local do problema (2.1), então $\nabla F(x^*) = 0$.*

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*. No caso de funções que possuem derivadas de segunda ordem, a proposição abaixo deve ser satisfeita:

Teorema 2.3.2 (Condição necessária de segunda ordem)**.** *Suponhamos que $F : \Re^n \to \Re$ seja duas vezes diferenciável no ponto $x^* \in \Re^n$. Se x^* é um minimizador local do problema (2.1), então $\nabla F(x^*) = 0$ e a matriz Hessiana $\nabla^2 F(x^*)$ é semidefinida positiva.*

Se $\nabla^2 F(x^*)$ é semidefinida positiva, significa que para qualquer $y \in \Re^n$, $y^T \nabla^2 F(x^*) y \geq 0$. Novamente as condições acima não garantem que x^* seja solução do problema (1.1). Condições suficientes são dadas na proposição abaixo:

Teorema 2.3.3 (Condição suficiente de segunda ordem)**.** *Suponha que F'' seja contínua em uma vizinhança de x^*, $\nabla F(x^*) = 0$ e $\nabla F(x^*)$ seja definida positiva. Então x^* é um minimizador local estrito de F ($F(x^*) < F(x), \forall x \in \Re^n$).*

Esta última condição não é necessária para que um ponto seja um minimizador de uma função.

Apesar dos Teoremas (2.2.1), (2.2.2) e (2.2.3) terem fundamental importância na classificação de possíveis soluções de (1.1), a maioria das demonstrações de convergência de algoritmos se contenta apenas em provar que os pontos limites das seqüências geradas são pontos estacionários da função objetivo. Diremos então que um algoritmo é *globalmente convergente* se gera uma seqüência $\{x_k\}_{k=0}^{\infty}$ tal que todo ponto limite desta seqüência é um ponto estacionário de F. Para a demonstração desses três teoremas, ver Bertsekas e/ou Nocedal [1, 12].

2.4 Aproximação: Interpolação polinomial

O estudo teórico é baseado em ferramentas de *interpolação-aproximação* que permitam encontrar um polinômio que passe por $n+1$ pontos conhecidos, denotados por $(x_0, y_0), (x_1, y_1), \cdots, (x_n, y_n)$ onde $x_0 \neq x_1 \neq \cdots \neq x_n$.

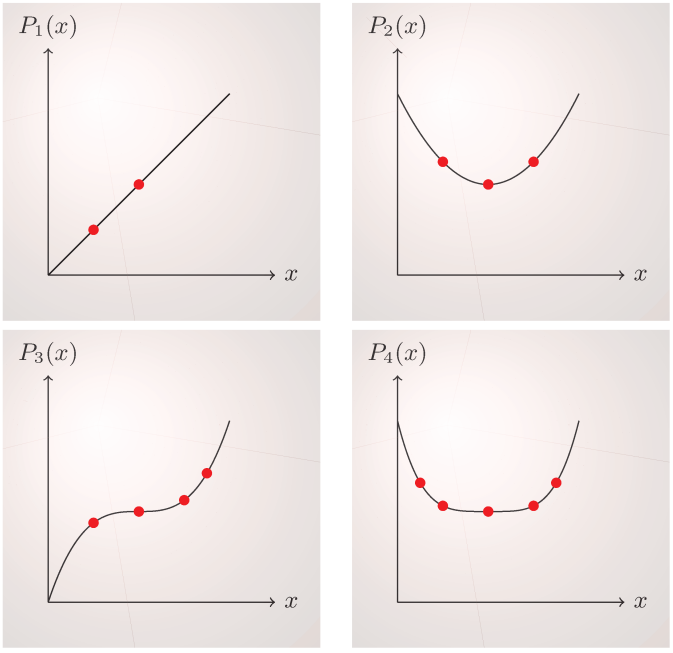

Figura 2.1. Polinômio interpolador de grau menor ou igual a n

Nessa classe de problemas, coloca-se a aproximação de funções que são conhecidas em pontos discretos, ou seja, conhecidos $y_0 = F(x_0)$, $y_1 = F(x_1)$,

$\cdots$, $y_n = F(x_n)$ e gostaríamos de encontrar um polinômio $p(x)$ tal que $p(x_0) = F(x_0)$, $p(x_1) = F(x_1)$, $\cdots$, $p(x_n) = F(x_n)$. Este problema é chamado de *Interpolação de Lagrange* (Joseph-Louis Lagrange, 1736-1813), se satisfaz estas condições, e o polinômio $p(x)$ que satisfaz estas condições é chamado de *polinômio interpolador* de $F(x)$ nos pontos $x_0, x_1, \cdots, x_n$. Na figura (2.1) nos dá uma idéia gráfica do polinômio interpolador dos $n+1$ pontos, em $\Re$.

Teorema 2.4.1. *Seja $f(x)$ uma função conhecida nos $(n+1)$ pontos distintos $x_0, x_1, \cdots, x_n$. Existe um único polinômio $p(x)$, de grau menor ou igual a n, tal que*

$$p(x_i) = f(x_i) \qquad \text{para } i = 0, 1, \cdots, n.$$

Demostração:

$$\begin{aligned} p(x) &= f(x_0)\ell_0^n(x) + f(x_1)\ell_1^n(x) + \cdots + f(x_n)\ell_n^n(x) && (2.2)\\ &= \sum_{k=0}^{n} f(x_k)\ell_k^n(x) && (2.3)\end{aligned}$$

onde ℓ_k^n são polinômios de Lagrange definidos por:

$$\begin{aligned} \ell_k^n(x) &= \frac{(x-x_0)(x-x_1)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n)}{(x_k-x_0)(x_k-x_1)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n)} \\ &= \prod_{i=0, i\neq k}^{n} \frac{x-x_i}{x_k-x_i}. \end{aligned}$$

A verificação que $p(x)$ é o polinômio interpolador segue diretamente da substituição dos pontos de interpolação, observando que:

$$\ell_k^n(x_k) = 1 \text{ e } \ell_k^n(x_i) = 0 \text{ para } i \neq k.$$

A unicidade é conseqüência de $p(x) \equiv 0$ ser o único polinômio de grau n que se anula em $n+1$ pontos distintos. Se supondo que existam dois polinômios interpoladores $p_1(x)$ e $p_2(x)$, usamos (2.2) com $p(x) = p_1(x) - p_2(x)$ para mostrar que $p(x) \equiv 0$ e portanto $p_1(x) \equiv p_2(x)$ ■

2.5 Região de confiança

Os métodos de *região de confiança* geram passos com a ajuda de um modelo quadrático da função objetivo. Os métodos de *busca linear*, muitas vezes,

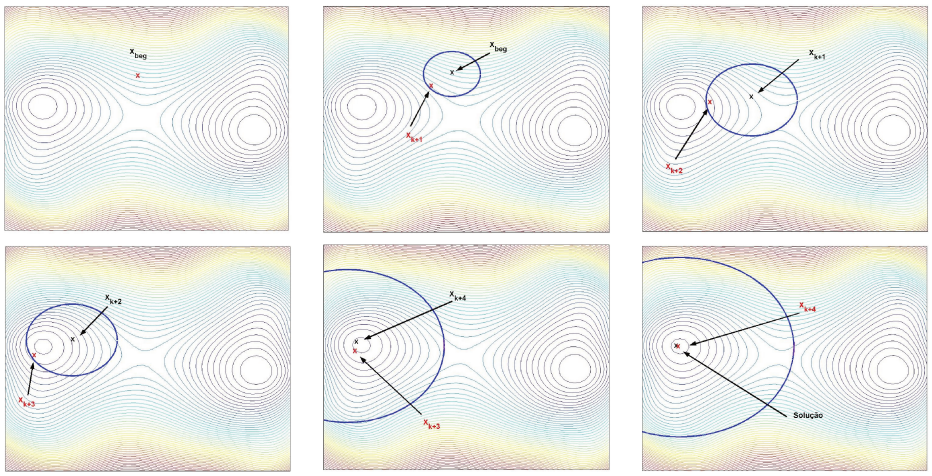

Figura 2.2. Minimizando Q_k dentro de uma região de confiança.

também utilizam este tipo de procedimento. Porém eles usam o modelo de forma diferente. Métodos de *busca linear* podem trabalhar com aproximações quadráticas para encontrar um passo conveniente com comprimento α ao longo desta direção de busca [15]. Métodos de *região de confiança* definem uma região em torno do ponto x_k, em torno da qual também foi construído o modelo de aproximação, como mostramos na figura (2.2), e então escolhem a direção ao final da qual o minimizador aproximado dentro da região vai ser obtido. Ou seja, eles escolhem o comprimento e a direção ao mesmo tempo. Se o passo não é aceito, eles reduzem o tamanho da região e encontram um novo minimizador.

Uma condição para definir o algoritmo é a escolha de um raio da região a cada iteração. Assim, dado um passo s_k definimos o quociente

$$\rho_k = \frac{F(x_k) - F(x_k + s_k)}{Q_k(0) - Q_k(s_k)}. \tag{2.4}$$

O numerador é chamado de redução atual, e o denominador é a redução prevista. Note que, como s_k é obtido minimizando o modelo Q_k sobre a região que inclui o passo $s_k = 0$, a redução prevista sempre é não negativa. Assim, se ρ_k está perto de 1, temos um bom ajuste entre o modelo quadrático Q e a função F com este passo. Aumentamos então o tamanho da região de confiança na iteração seguinte.

Se ρ_k é positivo, porém não está próximo de 1, não trocamos o tamanho da região de confiança; porém, se está perto de zero ou é negativo, reduzimos a região de confiança. O seguinte algoritmo descreve o processo.

Algoritmo 1
Dados de entrada:

Sejam $0 < \mu < \eta < 1$, $0 < \gamma_1 < \gamma_2 < 1 < \gamma_3$ constantes e $x_0 \in \Re^n$ e $\Delta_0 > 0$.

Para $k = 0, 1, 2, \cdots$ até convergir.

Passo 1. Aproximamos g_Q e H_Q, como descrito na seção (2.2).
Passo 2. Determinamos s_k, aproximação da solução do problema (1.2).
Passo 3. Calculamos ρ_k, dado na equação (2.4).
Passo 4. Se $\rho_k \leq \mu$ então $\Delta_k = \Delta \in [\gamma_1\Delta_k, \gamma_2\Delta_k]$ e voltamos ao passo 2.
Passo 5. $x_{k+1} = x_k + s_k$.
Passo 6. Se $\rho_k \leq \eta$ então $\Delta_{k+1} \in [\gamma_2\Delta_k, \Delta_k]$.
Caso contrário $\Delta_{k+1} \in [\Delta_k, \gamma_3\Delta_k]$.

CAPÍTULO 3

O método estudado

Os pontos não têm
partes nem dimensões.
Como podem combinar-se
para formar uma linha?
J. A. Lindon

3.1 Introdução

Neste capítulo apresentamos um método iterativo de Powell [17], tema básico do nosso estudo, que estima a solução do problema (1.1), onde uma aproximação para $F(x)$, $x \in \Re^n$ é construída a partir de valores da função objetivo. O próximo vetor de variáveis é gerado, usualmente tomando o mínimo da aproximação em uma região $S \subset \Re^n$. Para isso, definimos o espaço vetorial Π de todos os polinômios quadráticos de $\Re^n$ em $\Re$. A dimensão de Π é o número m dado por $\frac{1}{2}(n+1)(n+2)$, onde n é o número de variáveis. Cada modelo quadrático Q é um elemento de Π e é definido por $\widehat{m}$ valores da função F, onde $\widehat{m}$ está no intervalo $[n+1, m]$. Nos métodos de região de confiança que consideramos, Q é definido por condições de interpolação da forma

$$Q(x_i) = F(x_i), \quad i = 1, 2, \cdots, \widehat{m} \tag{3.1}$$

onde os pontos de interpolação $x_i \in \Re^n$, $i = 1, 2, \cdots, \widehat{m}$, estão em posições que garantem a não singularidade do sistema (3.1).
O usuário tem que fornecer um vetor inicial de variáveis x_{beg}, e valores inicial

e final, ρ_{beg} e ρ_{end}, de um raio da região de confiança ρ. Quando ρ é reduzido, x_{beg} é trocado pelo vetor de variáveis que gera o menor valor da função F. Assim x_{beg} é avaliado e Q é construído, da forma

$$Q(x) = c_Q + g_Q^T(x - x_{beg}) + \frac{1}{2}(x - x_{beg})^T H_Q(x - x_{beg}), \qquad x \in \Re^n. \quad (3.2)$$

Logo os parâmetros do modelo quadrático são o número real c_Q, o vetor $g_Q \in \Re^n$ e a matriz $n \times n$ simétrica, H_Q. Como o espaço vetorial de polinômios quadráticos de $\Re^n$ em $\Re$ tem dimensão $\widehat{m}$, os parâmetros de Q são definidos pela equação (3.1).

Em Powell [17], precisamos de funções de Lagrange para conseguir o polinômio interpolador Q que satisfaz as condições (3.1). Para $j = 1, 2, \cdots, \widehat{m}$, a j-ésima função de Lagrange é o polinômio quadrático ℓ_j de $\Re^n$ em $\Re$ que tem a propriedade

$$\ell_j(x_i) = \delta_{ij}, \quad i = 1, 2, \cdots, \widehat{m} \quad (3.3)$$

onde δ_{ij} é o delta de Kronecker. Tomamos

$$\ell_j(x) = c_j + g_j^T(x - x_{beg}) + \frac{1}{2}(x - x_{beg})^T H_j(x - x_{beg}), \quad x \in \Re^n \quad (3.4)$$

com g_j e H_j gradiente e hessiana da função de Lagrange ℓ_j. As condições (3.1) e (3.3) implicam na identidade

$$Q(x) = \sum_{j=1}^{\widehat{m}} F(x_j)\ell_j(x), \qquad x \in \Re^n. \quad (3.5)$$

Da equação (3.4) temos que os parâmetros de Q têm os valores

$$c_Q = \sum_{j=1}^{\widehat{m}} F(x_j)c_j, \quad g_Q = \sum_{j=1}^{\widehat{m}} F(x_j)g_j \quad e \quad H_Q = \sum_{j=1}^{\widehat{m}} F(x_j)H_j. \quad (3.6)$$

Powell propõe a resolução de dois subproblemas de região de confiança. O primeiro, o problema (1.2), utilizando o método de Moré e Sorensen [10], obtendo um ponto $x_\Delta = x_k + s$, $s \in \Omega$, em uma iteração de região de confiança. O segundo, o problema,

$$\begin{array}{c} \text{maximizar} \\ s \in \ N \end{array} \quad |\ell_j(x_k + s)| \quad , \quad (3.7)$$

onde $N = \{x : \|s\| \leq \rho\}$, com $\rho \leq \Delta$, $\rho_{beg} \geq \rho \geq \rho_{end}$, e ℓ_j é uma função de Lagrange contida em Π. No problema (3.7), obtemos um ponto $x_Q = x_k + s$, $s \in N$, em uma iteração padrão, o qual garante uma conveniente nova posição do ponto de interpolação.

3.2 Conceitos básicos do método de Powell

Sejam x_{beg} ponto inicial do algoritmo, ρ_{beg} e ρ_{end}, valores inicial e final de um raio da região de confiança ρ. Definimos os seguintes elementos para o desenvolvimento de nosso trabalho.

1. **Pontos de interpolação**

   Os pontos de interpolação x_i, $i = 1, 2, \cdots, \widehat{m}$, são escolhidos da seguinte forma: tomamos $x_1 = x_{beg}$ o valor inicial; e as componentes de $x_{2j} = x_{beg} + \rho_{beg} e_j$, $j = 1, 2, \cdots, n$, com e_j o j-th vetor coordenado em $\Re^n$. A escolha de x_{2j+1} depende dos valores de $F(x_{2j})$. Em Powell [17] é definido σ_j como -1 no caso em que $F(x_{2j}) \geq F(x_{beg})$ ou $\sigma_j = 1$ quando $F(x_{2j}) < F(x_{beg})$. Levando à seguinte fórmula:

$$x_{2j+1} = \begin{cases} x_{beg} - \rho_{beg} e_j, & \text{se } \sigma_j = -1, \\ x_{beg} + 2\rho_{beg} e_j, & \text{se } \sigma_j = +1, \end{cases} \quad j = 1, 2, \cdots, n \tag{3.8}$$

   Além disso temos

$$i(p, q) = 2n + 1 + p + \frac{1}{2}(q - 1)(q - 2), \quad 1 \leq p < q \leq n, \tag{3.9}$$

   e os outros pontos de interpolação

$$x_{i(p,q)} = x_{beg} + \rho_{beg}(\sigma_p e_p + \sigma_q e_q), \quad 1 \leq p < q \leq n. \tag{3.10}$$

   Usando (3.9) garantimos que os índices dos vetores em (3.10) fiquem no intervalo $[2n + 2, \widehat{m}]$ de números inteiros.

   Assim, para $\widehat{m} = m$ pontos de interpolação x_i, $i = 1, 2, \cdots, m$, com $m = \frac{1}{2}(n + 1)(n + 2)$.
   Para a função de Rosenbrock: $f(x) = (10(y - x^2))^2 + (1 - x)^2$, com $n = 2$, temos $m = 6$ pontos de interpolação. A figura (3.1) mostra esses pontos.

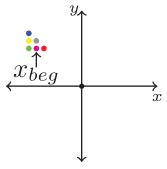

Figura 3.1. Pontos de interpolação da função de Rosenbrock

Determinamos o vetor $x_1 = x_{beg}$, melhor ponto de interpolação, ou seja o ponto que satisfaz $F(x_1) \leq F(x_i)$, $i = 2, \cdots, m$, o decréscimo suficiente da função objetivo. Como mostra-se na figura (3.1).

2. **Aproximando o gradiente e hessiana do modelo quadrático Q**

   Primeiramente, lembramos que nossos métodos não utilizam derivadas da F nem de sua aproximação Q, obtida por interpolação em valores da função F. Logo, o gradiente e a hessiana de F são aproximados no modelo Q que é minimizado no contexto de uma região de confiança.

   Tomando o ponto inicial $x_1 = x_{beg}$, que gera o parâmetro $c_Q = F(x_1)$ do modelo quadrático, para $j = 1, 2, \cdots, n$, dado de que $x_{2j} - x_{beg}$ e $x_{2j+1} - x_{beg}$ são diferentes de zero e múltiplos de e_j, permite que $(g_Q)_j$ e $(H_Q)_{jj}$ sejam deduzidos de $F(x_{2j}) - F(x_{beg})$ e $F(x_{2j+1}) - F(x_{beg})$, onde $(g_Q)_j$ e $(H_Q)_{pq}$ denotam a j-ésima componente de g_Q e o (p, q)-ésimo elemento de H_Q, respectivamente. Finalmente, os elementos fora da diagonal da matriz simétrica H_Q são obtidos das equações (3.1), (3.2) e (3.10) o que implica na identidade

$$\begin{aligned} c_Q + \rho_{beg}[\sigma_p(g_Q)_p + \sigma_q(g_Q)_q] \quad &+ \quad \frac{1}{2}\rho_{beg}^2[(H_Q)_{pp} + 2\sigma_p\sigma_q(H_Q)_{pq} + (H_Q)_{qq}] \\ &= \quad F(x_{i(p,q)}),\ 1 \leq p < q \leq n. \end{aligned} \tag{3.11}$$

   De fato, os requeridos elementos necessários da matriz $(H_Q)_{pq}$ são os únicos valores desconhecidos nesta relação para cada p e q. Assim, as posições dos pontos de interpolação reduzem o trabalho de resolver o sistema $\widehat{m}\text{x}\widehat{m}$ inicial (3.1) para somente $O(n^2)$ operações. Como se mostra no seguinte exemplo:

   - Gradiente
     Para $n = 3$, com $m = 10$ pontos de interpolação, temos:

$$g_Q = \begin{pmatrix} F(x_2) - F(x_{beg}) \\ F(x_4) - F(x_{beg}) \\ F(x_6) - F(x_{beg}) \end{pmatrix}$$

   - Hessiana
     Para $n = 3$, a matriz simétrica H_Q, esta definida por:

$$H_Q = \begin{pmatrix} F(x_3) - F(x_{beg}) & F(x_8) & F(x_9) \\ F(x_8) & F(x_5) - F(x_{beg}) & F(x_{10}) \\ F(x_9) & F(x_{10}) & F(x_7) - F(x_{beg}) \end{pmatrix}$$

3. **As funções de Lagrange**

Considerando o índice definido na equação (3.9) para $1 \leq p < q \leq n$, a função de Lagrange $\ell_{i(p,q)}$ tem uma expressão bem fácil. Se $p \neq q$, então a $i(p,q)$-ésima posição dos produtos $(x_i - x_{beg})_p(x_i - x_{beg})_q$, $i = 1, 2, \cdots, \widehat{m}$, é diferente de zero, onde a notação $(x - x_{beg})_j$ denota a j-ésima componente de $x - x_{beg}$. Portanto temos:

$$\ell_{i(p,q)}(x) = (\sigma_p \sigma_q / \rho_{beg}^2)(x - x_{beg})_p (x - x_{beg})_q, \quad x \in \Re^n, \quad 1 \leq p < q \leq n, \tag{3.12}$$

que prova que $(H_{i(p,q)})_{pq} = \sigma_p \sigma_q / \rho_{beg}^2$ é o único coeficiente de $\ell_{i(p,q)}$ distinto de zero.

Denotamos os coeficientes distintos de zero de ℓ_k, $k = 1, 2, \cdots, 2n+1$, fazendo uso dos polinômios quadráticos

$$\left.\begin{array}{rcl} \hat{\ell}_{2j}(x) & = & \dfrac{(x - x_{beg})_j (x - x_{2j+1})_j}{(x_{2j} - x_{beg})_j (x_{2j} - x_{2j+1})_j} \\ \hat{\ell}_{2j+1}(x) = & & \dfrac{(x - x_{beg})_j (x - x_{2j})_j}{(x_{2j+1} - x_{beg})_j (x_{2j+1} - x_{2j})_j} \end{array}\right\}, x \in \Re^n, \ j = 1, 2, \cdots, n. \tag{3.13}$$

Estas relações satisfazem as condições de Lagrange $\hat{\ell}_r(x_i) = \delta_{ir}$, $r = 2, 3, \cdots, 2n+1$, para inteiros i no intervalo $[1, n]$. As exceções serão para $i = i(p,j)$, $1 \leq p < j$, e $i = i(j,p)$, $j < q \leq n$, onde j é um inteiro entre $[1, n]$ tal que r é igual a $2j$ ou $2j+1$. Das propriedades de Lagrange e da fórmula (3.12) segue que a função

$$\ell_r(x) = \hat{\ell}_r(x) - \sum_{p=1}^{j-1} \hat{\ell}_r(x_{i(p,j)}) \ell_{i(p,j)}(x) - \sum_{q=j+1}^{n} \hat{\ell}_k(x_{i(p,j)}) \ell_{i(p,j)}(x), \quad x \in \Re^n, \tag{3.14}$$

satisfaz $\ell_r(x_i) = \delta_{ir}$ $i = 1, 2, \cdots, \widehat{m}$, quando a primeira ou segunda soma é eliminada ($j = 1$ ou $j = n$, respectivamente). As expressões (3.12), (3.13) e (3.14) mostram que os elementos fora da diagonal de $H_r = \nabla^2 \ell_r$, distintos de zero, são guardados na j-ésima linha e coluna de H_r. Isso também implica que $(H_r)_{jj}$ é o único elemento da diagonal de H_r diferente de zero.

Portanto, é fácil calcular os parâmetros das funções de Lagrange ℓ_r, $r = 2, 3, \cdots, 2n+1$. O algoritmo obtém os valores da identidade elementar

$$\ell_1(x) = 1 - \sum_{i=2}^{\widehat{m}} \ell_i(x), \qquad x \in \Re^n. \tag{3.15}$$

Além disso, a inicialização do procedimento para a primeira iteração é no menor inteiro no intervalo $[1, \widehat{m}]$ onde $F(x_k)$ assume o mínimo de $F(x_i)$, $i = 1, 2, \cdots, \widehat{m}$. Escolhemos valores iniciais $\rho = \rho_{beg}$ e $\Delta = \rho_{beg}$. A decisão entre as alternativas para a primeira iteração de nosso algoritmo é que o problema (1.2) possa ser resolvido.

Observação 3.2.1. As definições acima, junto com os $\widehat{m}$ valores da função objetivo definem o modelo quadrático Q dado na equação (3.5). Em cada iteração é necessário a atualização do modelo quadrático e também das funções de Lagrange ℓ_j, $j = 1, 2, \cdots, \widehat{m}$.

Assim, as $\widehat{m}$ funções de Lagrange determinam o seguinte sistema:

$$\begin{pmatrix} \ell_1(x_1) & \ell_1(x_2) & \ell_1(x_3) & \cdots & \ell_1(x_{\widehat{m}}) \\ \ell_2(x_1) & \ell_2(x_2) & \ell_2(x_3) & \cdots & \ell_2(x_{\widehat{m}}) \\ \vdots & \vdots & \vdots & & \vdots \\ \ell_{\widehat{m}}(x_1) & \ell_{\widehat{m}}(x_2) & \ell_{\widehat{m}}(x_3) & \cdots & \ell_{\widehat{m}}(x_{\widehat{m}}) \end{pmatrix} \begin{pmatrix} F(x_1) \\ F(x_2) \\ \vdots \\ F(x_{\widehat{m}}) \end{pmatrix} = \begin{pmatrix} Q(x_1) \\ Q(x_2) \\ \vdots \\ Q(x_{\widehat{m}}) \end{pmatrix}$$

os pontos de interpolação ficam em posições que garantem a não singularidade do sistema, pois, cada função de Lagrange obedece a condição $\ell_j(x_i) = \delta_{ij}, \quad i = 1, 2, \cdots, \widehat{m}$.

3.3 Atualização das funções de Lagrange

Retornemos à atualização dos coeficientes das funções de Lagrange quando o ponto de interpolação x_t, para algum t inteiro entre 1 e $\widehat{m}$, é trocado por uma nova posição $\widetilde{x}_t$, e as posições dos outros pontos de interpolação se conservam. Chamando as funções de Lagrange antiga e nova ℓ_i, $i = 1, 2, \cdots, \widehat{m}$, e $\widetilde{\ell}_i$, $i = 1, 2, \cdots, \widehat{m}$, respectivamente, os pontos $\widetilde{x}_t$ têm que satisfazer a condição

$$\ell_t(\widetilde{x}_t) \neq 0, \tag{3.16}$$

porque em outro caso o polinômio quadrático ℓ_t pode desaparecer do novo conjunto de pontos de interpolação. Assim, o novo sistema de equações (3.1) será singular. Além disso, para cada inteiro i em $[1, \widehat{m}]$, a diferença $\widetilde{\ell}_i$-ℓ_i será

um múltiplo de $\widetilde{\ell}_t$. Logo, para cada i, o fator multiplicador é definido pela equação $\widetilde{\ell}_i(\widetilde{x}_t) = \delta_{it}$. Assim deduzimos a fórmula

$$\widetilde{\ell}_t(x) = \ell_t(x)/\ell_t(\widetilde{x}_t), \quad x \in \Re^n, \tag{3.17}$$

e

$$\widetilde{\ell}_i(x) = \ell_i(x) - \ell_i(\widetilde{x}_t)\widetilde{\ell}_t(x), \quad x \in \Re^n, \quad i \neq t. \tag{3.18}$$

Powell explica que as fórmulas dadas em (3.17) e (3.18), têm excelentes propriedades de estabilidade. Em particular, se ℓ_t é qualquer polinômio quadrático, e se $\widetilde{x}_t$ é qualquer ponto de $\Re^n$ que satisfaz a condição (3.16), então a equação (3.17) dá a identidade $\widetilde{\ell}_t(\widetilde{x}_t) = 1$. Logo a expressão (3.18) dá $\widetilde{\ell}_i(\widetilde{x}_t) = 0$, $i \neq t$. Além disso, se x_j é o ponto de interpolação que foi trocado, então os valores

$$\ell_i(x_j) = \delta_{ij}, \qquad i = 1, 2, \cdots, \widehat{m} \tag{3.19}$$

já foram satisfeitos. Portanto a fórmula (3.19) é verdadeira para um certo j que é diferente de t. Neste caso a fórmula (3.17) prova que $\widetilde{\ell}_t(x_j) = 0$ é derivado de $\ell_t(x_j) = 0$, onde a função (3.18) satisfaz $\widetilde{\ell}_i(x_j) = \ell_i(x_j) = \delta_{ij}$, $i \neq t$.

3.4 Atualização do modelo quadrático

O modelo quadrático tem que ser revisado também, quando x_t é trocado para uma nova posição $\widetilde{x}_t$, onde $\widetilde{x}_t$ pode ser x_Δ ou x_Q, e os outros pontos de interpolação não se trocam. Então, tomando Q e $\widetilde{Q}$ como o antigo e o novo modelo, respectivamente, a equação $\widetilde{Q}(x_i) = Q(x_i)$, $i \neq t$, tem que ser satisfeita. Isto ocorre desde que $\widetilde{Q} - Q$ é um múltiplo da função de Lagrange (3.17), que dá o valor $\widetilde{Q}(\widetilde{x}_t) = F(\widetilde{x}_t)$. Logo $\widetilde{Q}$ é o polinômio quadrático

$$\widetilde{Q}(x) = Q(x) + \frac{F(\widetilde{x}_t) - Q(\widetilde{x}_t)}{\ell_t(\widetilde{x}_t)}\ell_t(x), \qquad x \in \Re^n. \tag{3.20}$$

O método de região de confiança guarda os coeficientes de Q e todas as funções de Lagrange. O número total de coeficientes será aproximadamente $\widehat{m}^2$. Então os coeficientes de $\widetilde{Q}$ são obtidos da fórmula (3.20) em apenas $O(\widehat{m})$ operações.

A equação (3.20) implica $Q(\widetilde{x}_t) = F(\widetilde{x}_t)$ para qualquer função $Q(x)$, $x \in \Re^n$, e também implica $Q(\widetilde{x}_i) = Q(x_i)$, $i \neq t$. Portanto, qualquer fracasso na condição $Q(x_t) = F(x_t)$ é corrigido quando x_t é trocado, e qualquer fracasso futuro se deve a erros de arredondamento do computador.

3.5 Uma cota de erro para o modelo quadrático

Assumimos nesta seção que a função objetivo $F(x)$, $x \in \Re^n$, tem terceiras derivadas que são limitadas e contínuas. Por enquanto, se y é qualquer ponto em $\Re^n$, e s é qualquer vetor em $\Re^n$ que tem comprimento Euclidiano unitário, ou seja, $\|s\| = 1$, então a função de uma variável

$$\Psi(\alpha) = F(y + \alpha s), \quad \alpha \in \Re, \tag{3.21}$$

tem terceiras derivadas limitadas e contínuas. Logo, existe o menor número não negativo M, independente de y e s, tal que cada função desta forma tem a propriedade

$$\left|\Psi'''(\alpha)\right| \leq M, \quad \alpha \in \Re. \tag{3.22}$$

Este valor de M é conveniente para a cota de erro exposto no seguinte teorema.

Teorema 3.5.1. *Sejam as condições dadas no parágrafo acima, e suponhamos que os pontos de interpolação $x_i \in \Re^n$, $i = 1, 2, \cdots, \widehat{m}$, fiquem em posições tais que a equação (3.1) define um único polinômio quadrático Q de $\Re^n$ a $\Re$. Então para cada $x \in \Re^n$, o erro do modelo quadrático satisfaz a condição*

$$|Q(x) - F(x)| \leq \frac{1}{6} M \sum_{j=1}^{\widehat{m}} |\ell_j(x)| \, \|x - x_j\|^3 , \tag{3.23}$$

onde as funções ℓ_j, $j = 1, 2, \cdots, \widehat{m}$, são funções de Lagrange das equações de interpolação.

Ver demostração do teorema em Powell [16].

Para o nosso caso, o valor de M não está disponível, mas na prática o algoritmo utiliza a desigualdade (3.23), onde experimentos numéricos provam que esta é satisfeita para qualquer número real M, mesmo se F não tiver qualquer terceira derivada. Considerando o modelo quadrático e as funções de Lagrange, e tomando $F(x)$ que será calculada para algum $x \in \Re^n$, observamos que ambos os lados da desigualdade (3.23) são iguais a zero se x é um ponto de interpolação. Então excluímos este caso. Portanto, nenhuma das distâncias $\|x - x_j\|$, $j = 1, 2, \cdots, \widehat{m}$, é zero. Da equação elementar

$$\sum_{j=1}^{\widehat{m}} \ell_j(x) = 1, \quad x \in \Re^n, \tag{3.24}$$

temos que o valor da soma da equação (3.23) é positivo. Logo, essa expressão e o valor de $F(x)$ geram uma cota inferior para M. Optamos por tomar M como a maior cota inferior, que é aplicada para estimar a precisão do modelo quadrático atual em um novo vetor de variáveis x. Portanto, tomamos um parâmetro não negativo $\frac{1}{6}M$, que é incluído no início de cada iteração.

Seja λ_Q o menor autovalor da matriz H_Q e seja a quantidade

$$\epsilon = \frac{1}{2}\Delta^2 max\left[0, \lambda_Q\right]. \tag{3.25}$$

Então

$$\frac{1}{6}M\sum_{j=1}^{\widehat{m}} \left|\ell_j(x)\right| \left\|x - x_j\right\|^3 < \epsilon. \tag{3.26}$$

Cada parcela $\frac{1}{6}M\sum_{j=1}^{\widehat{m}} \left|\ell_j(x)\right| \left\|x - x_j\right\|^3$, $j = 1, 2, \cdots, \widehat{m}$, é comparada separadamente com a expressão (3.25), que dá uma boa cota de erro na prática. Além disso, o algoritmo dá uma atenção particular para o vetor x que satisfaz $\|x - x_k\| \leq \rho$. Powell, usa uma estimativa do número $max\left\{\left|\ell_j(x)\right| : \|x - x_k\| \leq \rho\right\}$ que chamamos θ_j e que faz uma aproximação $\left\|x - x_j\right\|^3 \approx \left\|x_k - x_j\right\|^3$. Além disso, se prova a condição

$$\|x_j - x_k\| < \beta\rho, \quad j = 1, 2, \cdots, \widehat{m}, \tag{3.27}$$

que é importante no caso em que $\epsilon = 0$. Estes ingredientes fornecem o seguinte critério: assumir que os pontos de interpolação geram um modelo quadrático aceitável se ao menos uma destas condições for satisfeita:

$$\frac{1}{6}M\theta_j \left\|x_j - x_k\right\|^3 \leq \epsilon \quad ou \quad \|x_j - x_k\| < \beta\rho, \tag{3.28}$$

para qualquer j em $[1, \widehat{m}]$. Caso contrário, se a melhora para o modelo é exigida, então um ponto de interpolação x_j que não satisfaz as desigualdades (3.28) é trocado.

3.6 Os subproblemas de região de confiança

Para a resolução do primeiro subproblema

$$\begin{array}{ll} \text{minimizar} & Q(x_k + s) \\ s \in & \Omega \end{array}$$

utilizamos o método de *Moré e Sorensen* [10], proposto para a resolução de subproblemas da forma (1.2), no contexto do algoritmo de Newton com região de confiança para minimização irrestrita.

Em cada iteração, o método fornece uma aproximação $\lambda \geq 0$, para o parâmetro λ^* buscando obter uma matriz $H_Q + \lambda I$ positiva definida e então obtém s solução do sistema $(H_Q + \lambda I)s = -g_Q$, pedindo que $\|s\| = \Delta$ durante o processo de atualizar λ.

Os seguintes resultados proporcionam condições necessárias e suficientes para um ponto $s \in \Re^n$ para ser a solução do problema (1.2).

Lema 3.6.1. *Se s é solução para (1.2) então s é uma solução para a equação da forma*

$$(H_Q + \lambda I)s = -g_Q \tag{3.29}$$

com $H_Q + \lambda I$ semidefinida positiva, $\lambda \geq 0$, e $\lambda(\Delta - \|s\|) = 0$.

Isto pode ser observado na figura (3.2).

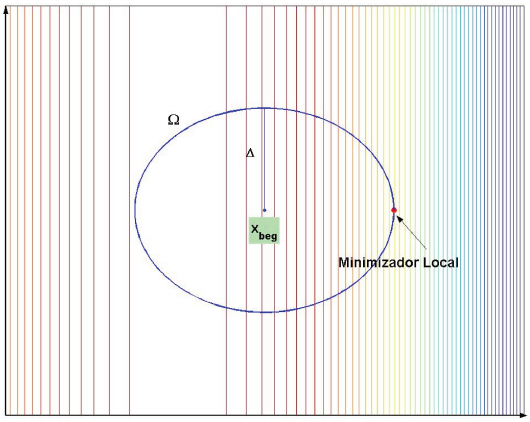

Figura 3.2. Matriz $H_Q + \lambda I$ semidefinida positiva, onde o minimizador está na fronteira da região Ω.

Lema 3.6.2. *Seja $\lambda \in \Re$, $s \in \Re^n$ satisfazendo a equação (3.29) com $H_Q + \lambda I$ semidefinida positiva.*

- ⋆ *s resolve $Q(s) = \min\{Q(w) : \|w\| = \|s\|\}$.*

⋆ *Se* $\lambda \geq 0$ *e* $\|s\| = \Delta$ *então* s *resolve (1.2).*

Se $H_Q + \lambda I$ *é positiva definida então* s *é a única solução para (1.2), como mostra a Figura* (3.3).

Ver demostração dos lemas em Moré e Sorensen, 1983, [10].

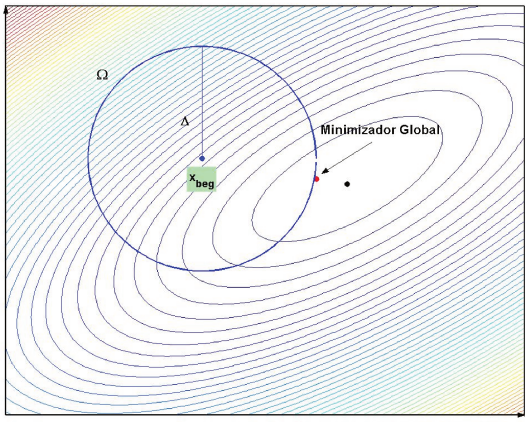

Figura 3.3. Matriz $H_Q + \lambda I$ definida positiva; o minimizador está na fronteira da região Ω.

A vantagem deste tipo de algoritmo é que, a cada iteração, o problema de encontrar $s^* \in \Re^n$ é transformado em um problema escalar de estimar $\lambda^* \in \Re$. Isto é feito através da fatoração de Cholesky da matriz $H_Q + \lambda I$ de dimensão $n \times n$, em cada iteração, e da aplicação do método de Newton para obter os zeros de uma função de uma variável da forma:

$$\varphi(\lambda) \equiv \frac{1}{\Delta} - \frac{1}{\|s_\lambda\|}. \tag{3.30}$$

A função é caracterizada pela presença de pólos, que são valores para os quais a função não está definida Em nosso caso estes pólos serão dados pelos autovalores da matriz H_Q conhecida. Se o pólo associado ao menor autovalor não está definido, teremos uma situação que chamaremos *hard case* e, neste caso, o método obtém uma aproximação para uma solução na fronteira da região de confiança.

Para a dedução do método levamos em conta:

- Estimativa do parâmetro λ:
  Consideremos a solução de $\|s\| = \Delta$ e apliquemos o método de Newton para encontrar os zeros da função (3.30), para λ no intervalo $(-\delta_1, \infty)$, onde δ_1 é o menor autovalor da matriz H_Q. O algoritmo atualiza λ pelo método de Newton.

  Algoritmo 2
  Dados de entrada:
  $\lambda \geq 0$ com $H_Q + \lambda I$ definida positiva e $\Delta > 0$.
  Passo 1. Fatorar $H_Q + \lambda I = R^T R$.
  Passo 2. Resolver $R^T q = -g_Q$.
  Passo 3. Resolver $R^T s = q$.
  Passo 4. $\lambda = \lambda + \left[\frac{\|s\|}{\|q\|}\right]^2 \left[\frac{\|s\|-\Delta}{\Delta}\right]$.

  Neste algoritmo $R^T R$ é a fatoração de Cholesky da matriz $H_Q + \lambda I$, onde $R \in \Re^{n\times n}$ é uma matriz triangular.

- Tratamento do *hard case*:

  Lema 3.6.3. *Seja* $0 < \sigma < 1$ *e suponha que*

  $$H_Q + \lambda I = R^T R, \quad (H_Q + \lambda I)s = -g, \lambda \geq 0.$$

  Se $z \in \Re^n$ *satisfaz*

  $$\|s + z\| = \Delta, \|Rz\|^2 \leq \sigma(\|Rp\|^2 + \lambda\Delta^2), \tag{3.31}$$

  então

  $$-\varphi(s + z) \geq 1/2(1 - \sigma)(\|Rp\|^2 + \lambda\Delta^2) \geq (1 - \sigma)\left|\varphi^*\right|,$$

  onde φ^* *é o valor ótimo de (1.2).*

  A demostração do lema pode ser encontrada em [10].

  A importância do lema (3.6.3) é no *hard case*. Nesta situação temos $\lambda \geq 0$ com $H_Q + \lambda I$ definida positiva e a solução s do sistema $(H_Q + \lambda I)s = -g$, satisfaz $\|s\| < \Delta$.

Podemos tentar satisfazer (3.31) com $z = \tau\hat{z}$, tomando um τ que satisfaça $\|s + \tau\hat{z}\| = \Delta$ e escolhendo $\hat{z}$ com $\|\hat{z}\| = 1$ tal que $\|R\hat{z}\|$ seja tão pequena quanto possível, onde

$$\tau = \frac{\Delta^2 - \|s\|^2}{s^T\hat{z} + sgn(s^T\hat{z})[(s^T\hat{z})^2 + (\Delta^2 - \|s\|^2)]^{1/2}}.$$

O vetor $\hat{z}$ com $\|\hat{z}\| = 1$ pode ser obtido com a técnica de LINPACK♣ (ver [2]), para estimar o menor valor singular da matriz triangular R.

- Salvaguardando λ:
  É necessária uma salvaguarda para λ para garantir que a solução seja encontrada. A salvaguarda depende de φ ser convexa e estritamente decrescente em $(-\delta_1, \infty)$, onde δ_1 é o menor autovalor da matriz H_Q. O esquema usa parâmetros λ_L, λ_U e λ_S tais que $[\lambda_L, \lambda_U]$ é um intervalo de incerteza que contém o λ desejado, e λ_S é uma cota inferior sobre $-\delta_1$. Com isso temos duas situações:

  1. Se $\lambda > \lambda_S$, então
     - $\lambda = max\{\lambda, \lambda_L\}$
     - $\lambda = max\{\lambda, \lambda_U\}$
  2. Se $\lambda \leq \lambda_S$, então $\lambda = max\left\{0.001\lambda_U, (\lambda_L\lambda_U)^{1/2}\right\}$.

  O item 1, garante que $\lambda \in [\lambda_L\,, \lambda_U]$. O segundo item garante que o comprimento do intervalo seja reduzido para garantir que λ permaneça dentro do intervalo de incerteza.

- Damos valores iniciais para λ_L, λ_U e λ_S
  Fazemos:
  $$\lambda_S = max\{-h_{ii}\}$$
  onde h_{ii} é o i-th elemento da diagonal de H_Q, e
  $$\lambda_L = max\left[0, \lambda_S, \frac{\|g_Q\|}{\Delta} - \|H_Q\|_1\right], \quad \lambda_U = \frac{\|g_Q\|}{\Delta} - \|H_Q\|_1.$$

- Fazendo atualizações dos limites λ_L, λ_U e λ_S
  Dados λ_S e λ, as regras para revisar λ_S são como segue.

♣A técnica de LINPACK é importante ao solucionar sistemas lineares para ter um método econômico estimando o número de condição $\kappa(A)$ da matriz de coeficientes. (cf. http://www.jstor.org/journals/siam.html)

Se $\lambda \in (-\delta_1, \infty)$ e $\varphi(\lambda) < 0$ então $\|s_\lambda\| < \Delta$ e podemos calcular τ e $\hat{z}$, fazemos:

$$\lambda_S = max\left(\lambda_S, \lambda - \|R\hat{z}\|^2\right). \tag{3.32}$$

Se $\lambda \leq -\delta_1$, então

$$\lambda_S = max\left[\lambda_S, \lambda + \frac{\gamma}{\|u\|^2}\right]. \tag{3.33}$$

com $\gamma \geq 0$, tal que a submatriz $H_Q + \lambda I + \gamma e_l e_l^T$ seja singular, com $l \leq n$. Atualizamos λ_L, λ_U e λ_S:

1. Se $\lambda \in (-\delta_1, \infty)$ e $\varphi(\lambda) < 0$ então

   $$\lambda_U = min\left(\lambda_U, \lambda\right);$$

   senão

   $$\lambda_L = max\left(\lambda_L, \lambda\right);$$

2. Atualizamos λ_S usando (3.32) e (3.33).
3. $\lambda_L = max\left(\lambda_L, \lambda_S\right)$.

Se não conseguirmos obter um λ tal que $\|s_\lambda\| = \Delta$ e $H_Q + \lambda I$ seja definida positiva, então procuramos um vetor $\widehat{z}$ com $\|\widehat{z}\| = 1$ onde $\|R\widehat{z}\|$ é o menor possível, R é a matriz triangular superior da fatoração de Cholesky de $H_Q + \lambda I$. Com esta situação temos o *hard case*, explicado acima. Ou seja, o problema de calcular o vetor $\widehat{z}$ se reduz a resolver o problema:

$$\begin{array}{cc} \text{minimizar} & \|R\widehat{z}\| = \sigma_{min} \\ \|\widehat{z}\| = 1 & \end{array}$$

onde σ_{min} é o menor valor singular da matriz R, podemos obter $\widehat{z}$ de maneira que $\|R\widehat{z}\| \approx \sigma_{min}$. Para o cálculo do vetor $\widehat{z}$, utilizamos dois artigos [2, 7], que sugerem uma estimativa para o vetor buscando o autovetor associado ao menor autovalor de R através de uma iteração do método das potências inverso. A sugestão é tomar e, um vetor com componentes iguais a ± 1, de forma que a solução ω do sistema $R^T\omega = e$ tenha norma grande, e escolher $\widehat{z} = \frac{v}{\|v\|}$ tal que v seja solução de $Rv = \omega$.

- O critério de convergência do método de Moré e Sorensen
  Este algoritmo pode terminar em três situações: Se encontrar durante o processo iterativo s suficientemente próximo da fronteira da região de confiança; se H_Q é definida positiva e se s, dado pelo algoritmo é solução do problema (1.2).

  Assim, dados os parâmetros σ_1 e σ_2 no intervalo $(0,1)$, e um valor inicial $\lambda \geq 0$ tal que $H_Q + \lambda I$ é definida positiva, um vetor s é computado no passo 2 do método de Newton para o menor zero da função. Então o algoritmo termina se

$$|\Delta - \|s\|| \leq \sigma_1 \Delta \quad ou \quad \|s\| \leq \Delta \quad e \quad \lambda = 0, \tag{3.34}$$

  ou

$$\|R(\tau\hat{z})\|^2 \leq \sigma_1 (2 - \sigma_1)\, max \left\{ \sigma_2, \|Rs\|^2 - \lambda\Delta^2 \right\}. \tag{3.35}$$

  No caso (3.35) o algoritmo termina e $s + \tau\hat{z}$ é a aproximação desejada.

- Convergência global
  O método de região de confiança para minimização irrestrita utilizado no algoritmo para resolver os subproblemas apresenta alguns resultados importantes de convergência.

  Se utilizamos o algoritmo de Moré e Sorensen para resolver o problema (1.2) e a aproximação s_k da solução do subproblema satisfaz (3.34) e (3.35) então é mostrado em [10] que s_k satisfaz

$$\Psi_k(s_k) - \Psi_k^* \leq \sigma_1 (2 - \sigma_1)\, max \left\{ |\Psi_k^*|, \sigma_2 \right\}, \quad \|s_k\| \leq (1 + \sigma_1)\, \Delta_k. \tag{3.36}$$

  Uma consequência desta relação é que, se $\Psi_k^* \neq 0$, então a aproximação s_k dada pelo algoritmo com $\sigma_2 = 0$ satisfaz

$$-\Psi_k(s_k) \geq \beta_1 \|\Psi_k^*\|, \ \|s_k\| \leq \beta_2 \Delta_k \tag{3.37}$$

  onde $\beta_1,\ \beta_2 > 0$. Com isso temos um primeiro resultado de convergência do algoritmo de região de confiança para minimização irrestrita dado no seguinte teorema, e cuja demostração é feita em [10].

Teorema 3.6.4. *Seja $F : \Re^n \to \Re$, duas vezes continuamente diferenciável no conjunto de nível $\Lambda = \{x | F(x) \leq F(x_0)\}$ e considere a*

seqüência $\{x_k\}$ produzida pelo algoritmo de região de confiança para minimização irrestrita, onde s_k satisfaz (3.37). Se Λ é um conjunto compacto então ou o algoritmo termina em x_ℓ com $\nabla F(x_\ell) = 0$ e $\nabla^2 F(x_\ell)$ semidefinida positiva, ou $\{x_k\}$ tem um ponto limite $x^* \in \Lambda$ com $\nabla F(x^*) = 0$ e $\nabla^2 F(x^*)$ semidefinida positiva.

Um segundo resultado importante de convergência não depende da compacidade do conjunto de nível Λ e é dado no próximo teorema, conforme [10].

Teorema 3.6.5. *Seja $F : \Re^n \to \Re$, duas vezes continuamente diferenciável no conjunto de nível $\Lambda = \{x | F(x) \leq F(x_0)\}$ e considere a seqüência $\{x_k\}$ produzida pelo algoritmo de região de confiança para minimização irrestrita, onde s_k satisfaz (3.37). Se x^* é um ponto limite de $\{x_k\}$ e $\nabla^2 F(x^*)$ é não singular então $\{x_k\}$ converge para x^* e $\nabla^2 F(x^*)$ é definida positiva.*

Este teorema pode ser estendido com a hipótese adicional $\|s_k\| \leq \beta_3 \Delta_k$ onde $\beta_3 > 0$, para mostrar que a seqüência converge para x^* com taxa de convergência q-superlinear e, se $\nabla^2 F$ é Lipschitz contínua, então a taxa de convergência é quadrática.

A seguir apresentamos a esquematização do algoritmo de Moré e Sorensen com os elementos necessários descritos ao longo desta seção.

Resumo do Algoritmo do método de Moré e Sorensen
Dados de entrada:
Δ, H_Q, g_Q, tomando $s = 0$.
Enquanto o critério de parada não for satisfeito:
Passo 1. Garantir λ no intervalo de salvaguarda.
Passo 2. Se $H_Q + \lambda I$ é positiva definida então fatorar
$H_Q + \lambda I = R^T R$.
Passo 3. Se a fatoração é realizada com sucesso, então:
Se $\|s\| < \Delta$, calcular τ e $\hat{z}$.
Verificar se os critérios de convergência são satisfeitos.
Atualizar λ_L, λ_U, λ_S.
Atualizar λ utilizando um passo do método de Newton.
Passo 4. Se a fatoração não pode ser realizada, então:
Atualizar λ_L, λ_U, λ_S.
Verificar se os critérios de convergência são satisfeitos.

Agora, se o sistema $Q(x_i) = F(x_i), \quad i = 1, 2, \cdots, \widehat{m}$ é singular então procuramos um índice $t = j$, explicado na seção (3.6). E solucionamos o problema a seguir, o qual gera o ponto x_Q na iteração chamada *iteração padrão.*

$$\begin{array}{ll} \text{maximizar} & |\ell_j(x_k + s)| \\ \text{sujeito a} & \|s\| \leq \rho \end{array}$$

onde k é um inteiro entre $[1, \widehat{m}]$ com $\widehat{m}$ no intervalo $[n+1, m]$, e é tal que $F(x_i)$ é o menor valor de $F(x_i)$, $i = 1, 2, \cdots, \widehat{m}$.

UOBYQA aplica um procedimento que faz $|\ell_j(x_k + s)|$ relativamente grande sujeito a $\|s\| \leq \rho$, o inteiro j do subproblema é sempre diferente de k, ali a condição de Lagrange (3.19) inclui $\ell_i(x_j) = 0$.

Retomando a equação (3.4) e fazendo h_j o vetor $g_j + G_j(x_k - x_{beg})$, o qual é calculado, procuramos uma aproximação s^* do problema

$$\begin{array}{ll} \text{maximizar} & |\ell_j(x_k + s)| \equiv \left| h_j^T s + \frac{1}{2} s^T G_j s \right| \\ \text{sujeito a} & \|s\| \leq \rho \end{array} \tag{3.38}$$

além disso, a restrição da região de confiança permite a s ser substituído por $-s$ logo, a equação é equivalente a

$$\begin{array}{ll} \text{maximizar} & \left| h_j^T s \right| + \left| \frac{1}{2} s^T G_j s \right| \\ \text{sujeito a} & \|s\| \leq \rho \end{array} \tag{3.39}$$

se $\widehat{s}$ e $\widetilde{s}$ são os valores que maximizam a $\left| h_j^T s \right|$ e $\left| \frac{1}{2} s^T G_j s \right|$, respetivamente sujeito a $\|s\| \leq \rho$, então s pode ser uma solução adequada do problema (3.38), está é escolhida entre $\pm\widehat{s}$ e $\pm\widetilde{s}$ que da o maior valor para o problema. De fato, para cada s factível, incluindo a solução exata, encontramos uma cota

$$\begin{aligned} \left| h_j^T s \right| + \left| \frac{1}{2} s^T G_j s \right| &\leq \left(\left| h_j^T \widehat{s} \right| + \left| \frac{1}{2} \widehat{s}^T G_j \widehat{s} \right| \right) + \left(\left| h_j^T \widetilde{s} \right| + \left| \frac{1}{2} \widetilde{s}^T G_j \widetilde{s} \right| \right) \\ &\leq 2max \left[\left(\left| h_j^T \widehat{s} \right| + \left| \frac{1}{2} \widehat{s}^T G_j \widehat{s} \right| \right), \left(\left| h_j^T \widetilde{s} \right| + \left| \frac{1}{2} \widetilde{s}^T G_j \widetilde{s} \right| \right) \right]. \end{aligned}$$

Se segue que a escolha proposta de s da o valor $|\ell_j(x_k + s)|$ que é, pelo menos, a metade do ótimo valor. O valor de $\widehat{s}$ é o vetor $\pm\rho h_j / \|h_j\|$, enquanto $\widetilde{s}$ é um autovetor dum autovalor de G_j de maior modulo, o qual é caro de calcular. Nós mostraremos como gerar $\widetilde{s}$. Usaremos um método inspirado

no *método das potencias* [27], pára obter o maior autovalor.

A técnica começa achando uma coluna de G_j, ω digamos, que tem a maior norma euclidiana. Conseqüentemente, tomando $\upsilon_1, \upsilon_1, \cdots, \upsilon_n$ as colunas da matriz simétrica G_j, deduzimos a cota

$$\begin{aligned} \|G_j\omega\| \geq \|\omega\|^2 &= max\left\{\|\upsilon_k\| : k = 1, 2, \cdots, n\right\}\|\omega\| \\ &\geq \|\omega\|\sqrt{\frac{1}{n}\sum_{k=1}^{n}\|\upsilon_k\|^2} \\ &\geq \frac{\|\omega\|}{\sqrt{n}}\sigma(G_j) \end{aligned}$$

onde $\sigma(G_j)$ é o raio espetral de G_j. Isto pode ser muito caro computacionalmente. Então, o algoritmo escolhe $\widetilde{s}$ do espaço bidimensional linear de $\Re^n$ que é gerado por ω e $G_j\omega$, onde, $\widetilde{s}$ tem a forma $\alpha\omega + \beta G_j\omega$, onde o quociente α/β é calculado para maximizar a expressão

$$\frac{\left|(\alpha\omega + \beta G_j\omega)^T G_j(\alpha\omega + \beta G_j\omega)\right|}{\|(\alpha\omega + \beta G_j\omega)\|^2} \tag{3.40}$$

o qual determina a direção de $\widetilde{s}$. Então o comprimento de $\widetilde{s}$ é definido por $\|\widetilde{s}\| = \rho$, o sinal de $\widetilde{s}$, sendo sem importância.

1. **Gerando $\widetilde{s}$**

   Definimos $V := \omega$, $D := G_j\omega$, $r := \beta/\alpha$, equação (3.40), podemos reescrever:

$$\begin{aligned} \frac{(V + rD)^T G_j(V + rD)}{(V + rD)^2} &= \frac{V^T G_j V + rV^T G_j D + rD^T G_j V + r^2 D^T G_j D}{V^2 + 2rV^T D + r^2 D^2} \\ &= \frac{D^T G_j D + 2rV^T G_j D + V^T G_j V}{V^2 + 2rV^T D + r^2 D^2} = f(r) \end{aligned}$$

   nós procuraremos a r^*, zeros da equação

$$\frac{\partial f(r^*)}{\partial r} = 0$$

$$\begin{aligned}
&\Leftrightarrow \quad (2rD^TG_jD + 2V^TG_jD)(V^2 + 2rV^TD + r^2D^2) - (r^2D^TG_jD + \\
&\qquad 2rV^TG_jD + V^TG_jV)(2rD^2 + 2V^TD) = 0 \\
&\Leftrightarrow \quad \left[(D^TG_jD)(V^TD) - D^2D^2\right] r^2 + \left[(D^TG_jD)V^2 - D^2(V^TD)\right] r + \\
&\qquad \left[D^2V^2 - (V^TD)^2\right] = 0
\end{aligned}$$

(Usando o fato que $D = G_jV \Leftrightarrow D^T = V^TG_j^T = V^TG_j$). Nós obtemos assim um simples equação $ax^2 + bx + c = 0$. Encontramos os dois zeros desta equação e escolhemos o r^* que maximiza (3.40), $\widetilde{s}$ é assim $V + r^*D$.

2. **Gerando $\widehat{u}$ e $\widetilde{u}$ usando $\widehat{s}$ e $\widetilde{s}$**

   Tendo gerado $\widehat{s}$ e $\widetilde{s}$ dos modos que foram descritos, o algoritmo toma s por uma combinação linear destes vetores, mas a escolha não é restringida a $\pm\widehat{s}$ ou $\pm\widetilde{s}$ como sugerido acima (a menos que $\widehat{s}$ e $\widetilde{s}$ quase sejam ou precisamente paralelos). Ao invés, os vetores são achados $\widehat{u}$ e $\widetilde{u}$ de comprimento um, no espaço gerado por de $\widehat{u}$ e $\widetilde{u}$ isso satisfaça a condição $\widehat{u}^T\widetilde{u} = 0$ e $\widehat{u}^TG_j\widetilde{u} = 0$. O s final será uma combinação de $\widehat{u}$ e $\widetilde{u}$.

   Tomando: $T = \widetilde{s}$ e $V = \widehat{s}$ temos:

$$\widetilde{u} = cos(\theta)T + sen(\theta)V \quad e \quad \widehat{u} = sen(\theta)T + cos(\theta)V \tag{3.41}$$

   obtendo diretamente $\widehat{u}^T\widetilde{u} = 0$.

   Encontremos θ tal que $\widehat{u}^TG_j\widetilde{u} = 0$:

$$\begin{aligned}
&\widehat{u}^TG_j\widetilde{u} \quad = 0 \\
&\quad \Leftrightarrow \qquad (-sen(\theta)T + cos(\theta)V)^TG_j(cos(\theta)T + sen(\theta)V) = 0 \\
&\quad \Leftrightarrow \qquad (cos^2(\theta) - sen^2(\theta))V^TG_jT + (T^TG_jT - V^TG_jV)sen(\theta)cos(\theta) = 0
\end{aligned}$$

   Usando

$$\begin{aligned}
sen(2\theta) &= 2sen(\theta)cos(\theta) \\
cos(2\theta) &= cos^2(\theta) - sen^2(\theta) \\
tan(\theta) &= \frac{sen(\theta)}{cos(\theta)}
\end{aligned}$$

   Obtemos

$$(V^TG_jT)cos(2\theta) + \frac{T^TG_jT - V^TG_jV}{2} sen(2\theta) = 0 \Leftrightarrow \theta = \frac{1}{2} arctan\left(\frac{2V^TG_jT}{V^TG_jV - T^TG_jT}\right)$$

Usando o valor de θ na equação (4.5), obtemos os valores precisados de $\widehat{u}$ e $\widetilde{u}$.

3. **Gerando o valor final de s a partir de $\widehat{u}$ e $\widetilde{u}$**

O valor final de s tem a forma $s = \rho(cos(\phi)\widehat{u} + sen(\phi)\widetilde{u})$ e ϕ tem um dos seguintes valores: $\left\{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi, \frac{-\pi}{4}, \frac{-\pi}{2}, \frac{-3\pi}{4}\right\}$. Escolhemos o valor de ϕ o qual maximize (4.4).

Observação 3.6.1. A escolha de $\widetilde{s}$ alcança a propriedade:

$$\left|\widetilde{s}^T G_j \widetilde{s}\right| \geq \frac{1}{2\sqrt{n}}\sigma(G_j)\rho^2. \tag{3.42}$$

3.7 Atualização de Δ e ρ

O valor de Δ é atualizado em uma forma que depende do quociente

$$r = \left[F(x_k) - F(x_k + s)\right] / \left[Q(x_k) - Q(x_k + s)\right]. \tag{3.43}$$

que é a redução atual sobre a redução prevista utilizada no algoritmo de região de confiança, definido no Capítulo 2.

O novo raio da região de confiança $\Delta \geq \rho$ pode ser

$$\Delta = \begin{cases} max\left[\Delta, \frac{5}{4}\left\|s\right\|, \rho + \left\|s\right\|\right], & r \geq 0.7, \\ max\left[\frac{1}{2}\Delta, \left\|s\right\|\right], & 0.1 < r < 0.7, \\ \frac{1}{2}\left\|s\right\|, & 0.1 < r, \end{cases}$$

Se $\rho > \rho_{end}$, o algoritmo reduz ρ de ρ_{old} para ρ_{new}, aplicando a fórmula

$$\rho_{new} = \begin{cases} \rho_{end}, & \rho_{end} < \rho_{old} \leq 16\rho_{end}, \\ \sqrt{\rho_{old}\rho_{end}}, & 16\rho_{end} < \rho_{old} \leq 250\rho_{end}, \\ 0.1\rho_{old}, & \rho_{old} > 250\rho_{end}. \end{cases}$$

que é designado para dar reduções ao redor de um fator de dez que permite alcançar $\rho = \rho_{end}$.

3.8 Outros detalhes do método de Powell

No inicio de qualquer iteração do método de região de confiança que estamos usando, a função modelo $Q(x)$, $x \in \Re^n$, definida pelas condições de interpolação (3.1), está disponível. Além disso, suponhamos, sem perda de geralidade, que x_1 é o melhor ponto de interpolação, o qual significa que tem a propriedade

$$F(x_1) \leq F(x_i), \quad i = 1, 2, \cdots, \widehat{m}. \tag{3.44}$$

Se mais de um valor $F(x_i), \quad i = 1, 2, \cdots, \widehat{m}$, é mínimo, tomamos x_1 como o primeiro valor que foi calculado. Consideremos a região de confiança Ω. O ponto x_Δ será gerado pelo método de Moré e Sorensen, descrito na seção anterior.
A iteração pára, na prática, quando um ponto $\widetilde{x}_\Delta$ de Ω que satisfaz a condição

$$F(\widetilde{x}_\Delta) - F(x_1) \leq (1 - \epsilon) \left[F(x_\Delta) - F(x_1) \right], \tag{3.45}$$

é encontrado, onde ϵ é uma tolerância positiva. Assim, a escolha de $\epsilon = 0.001$, garante que a estimativa $\widetilde{x}_\Delta \approx x_\Delta$ proporciona pelo menos 99% da redução maior de F, quando $F(x_1)$ possa ser alcançado dentro da região de confiança.

As iterações do método que geram x_Δ, chamadas *iterações de região de confiança*, calculam o valor $F(x_\Delta)$ e quando um ponto de interpolação x_i, $i = 1, 2, \cdots, \widehat{m}$, será trocado por x_Δ. Além disso, Q é atualizado para satisfazer a nova equação de interpolação (3.1), como foi explicado na seção 3.2. Porém um novo vetor de variáveis diferente de x_Δ pode ser necessário para garantir que o sistema (3.1) seja não singular. Caso contrário, para alcançar a aproximação $F \approx Q$ pode ser necessário que todos os pontos de interpolação fiquem muito perto de x_1. Portanto nosso método também inclui um tipo de iteração chamada "iteração padrão", que resolve o problema (3.7) e calcula o valor da função objetivo em um novo ponto, $x_Q \in \Re^n$.

No método estudado, a primeira iteração é uma iteração de região de confiança, e uma iteração padrão é sempre realizada em uma iteração de região de confiança. Portanto a decisão de que a iteração seguinte seja uma iteração padrão é tomada durante a iteração de região de confiança. Se esta decisão é feita, então a iteração de região de confiança escolhe o ponto x_Q e o inteiro t do intervalo $[2, \widehat{m}]$ tal que x_t seja tirado do conjunto $\{x_1, x_2, \cdots, x_{\widehat{m}}\}$ e substituído por x_Q. Assim, as únicas operações da iteração padrão são o calculo de $F(x_Q)$ e a atualização de Q, necessária porque $Q(x_t) = F(x_t)$ é substituída por $Q(x_Q) = F(x_Q)$ no sistema (3.1). Todas as outras condições

de interpolação são conservadas, e trocamos x_1 por x_Q se $F(x_Q)$ é menor que $F(x_1)$.

É possível que o vetor x_Δ da iteração de região de confiança seja o ponto x_1, já que satisfaz $Q(x_1) = F(x_1)$ e a condição de decréscimo da função F com $x \in S$. Depois de gerar x_Δ, cada iteração de região de confiança satisfaz a condição

$$\|x_\Delta - x_1\| \geq \frac{1}{2}\rho \tag{3.46}$$

para a escolha de ρ que foi tratado acima. O valor da função $F(x_\Delta)$ é calculado na iteração atual se e somente se a condição (3.46) é verdadeira. O parâmetro ρ é dado para gerar passos grandes no espaço de variáveis durante a iteração; seu valor inicial será dado no início do algoritmo. Além disso, quando não é possível mais progresso com o valor atual, ρ é reduzido, exceto no caso em que o término ocorre se ρ alcança seu valor final. As reduções são por um fator de dez, e ρ nunca é aumentado.

O raio da região de confiança Δ é outro conjunto para ρ sobre cada iteração, ou é ajustado em uma forma usual, sujeito a uma cota $\Delta \geq \rho$.

$$Se \quad F(x_\Delta) \leq F(x_1) - 0.1\,[Q(x_1) - Q(x_\Delta)] \tag{3.47}$$

ou seja, o passo de x_1 a x_Δ reduz o valor da função por um décimo da quantidade que é prevista pelo modelo. Esta quantidade será positiva pela condição (3.46). O fator 0.1 do lado direito da equação (3.47) pode ser alterado por qualquer outra constante do intervalo $(0, 1)$. Se a redução é alcançada em (3.47), então a iteração seguinte é também uma iteração de região de confiança, que se inicia depois que o valor de Δ seja revisado. Isto causa uma nova função modelo que satisfaz $Q(x_\Delta) = F(x_\Delta)$, e reordena os pontos de interpolação tal que x_Δ é o novo x_1.

Se $F(x_\Delta)$ é calculado sobre uma iteração de região de confiança, porém a equação (3.47) falha, então Δ é reduzido se este excede a ρ, e usualmente Q é revisado para incluir um novo valor de F na seguinte função modelo. Além disso, x_Δ é trocado por x_1, se $F(x_\Delta) < F(x_1)$. Se as condições (3.46) ou (3.47) falham sobre uma iteração de região de confiança, então a iteração seguinte será uma iteração padrão. O algoritmo faz uma escolha do índice t da condição de interpolação que será substituído. O valor de t é um número inteiro entre $[1, \widehat{m}]$ que tem a propriedade

$$\|x_t - x_1\| = max\left\{\|x_i - x_1\| : i = 1, 2, \cdots, \widehat{m}\right\}, \tag{3.48}$$

e o quociente é $\|x_t - x_1\| / \rho$ comparado com a constante $\beta > 1$. A decisão de utilizar uma iteração padrão é tomada se o quociente excede a ρ, de outro jeito, se $\|x_t - x_1\| > \beta\rho$ e $\theta_t > \theta_*$ é verdadeira para um ponto de interpolação x_t. Portanto t é um inteiro entre $[1, \widehat{m}]$ que tem a propriedade

$$\zeta\left(\|x_t - \widetilde{x}\|\right)|\ell_t(x_\Delta)| = max\left\{\zeta\left(\|x_i - \widetilde{x}\|\right)|\ell_i(x_\Delta)| : i = 1, 2, ..., \widehat{m}\right\} \tag{3.49}$$

ζ é uma função de peso e onde $\widetilde{x}$ é escolhido entre x_1 e x_Δ que é o seguinte vetor x_1, ζ é a função $\zeta(r) = max[1, (r/\rho)^3]$, $r \geq 0$. Não esperamos atualizações para melhorar o modelo quadrático, por enquanto, se $|\ell_t(x_\Delta)| \leq 1$, $\|x_t - \widetilde{x}\| \leq \rho$ e $x_\Delta \neq \widetilde{x}$ são verdadeiros para uma escolha de t, que é o caso quando Q não é atualizado. De outra forma, x_Δ substitui a x_t na equação (3.1).

A seguir, apresentamos um resumo do algoritmo, dividido em passos, onde descrevemos os elementos mais importantes da teoria desenvolvida neste capítulo.

Dados de entrada:
$x_1 = x_{beg}$, Δ, ρ_{beg}, ρ_{end} com $\rho_{beg} \geq \rho \geq \rho_{end}$ e $\Delta \geq \rho$.
Passo 1. Definimos os pontos de interpolação.
Passo 2. Determinamos o ponto $x_1 = x_{beg}$, tal que $F(x_1) \leq F(x_i)$, $i = 1, \cdots, \widehat{m}$.
Passo 3. Definimos as funções de Lagrange e o primeiro modelo quadrático.
Verificamos se o sistema $F(x_i) = Q(x_i)$ é não singular.
Passo 4. Aproximamos o gradiente e a hessiana de Q.
Passo 5. Calculamos o ponto $x_k = x_\Delta$ ou $x_k = x_Q$ resolvendo um dos subproblemas detalhados ao longo deste capítulo, de minimização dentro de uma região de confiança conveniente.
Passo 6. Reordenamos os pontos, verificando a condição dada no passo 2.
Passo 7. O novo valor da função objetivo $F(x_k)$ é calculado, se a condição (3.46) é satisfeita.
Passo 8. Δ é atualizado, escolhemos j o índice que está no conjunto de pontos de interpolação, que esteja mais longe de x_1,
e x_1 é trocado por x_k.
Passo 9. Atualizamos o modelo quadrático Q e as funções de Lagrange por meio das equações dadas na seções (3.3) e (3.4).
Passo 10. Se $\rho > \rho_{end}$ fazer um decréscimo em ρ dada na seção (3.7).
Passo 11. As iterações são completadas, porém outros valores da função podem ser requeridos.

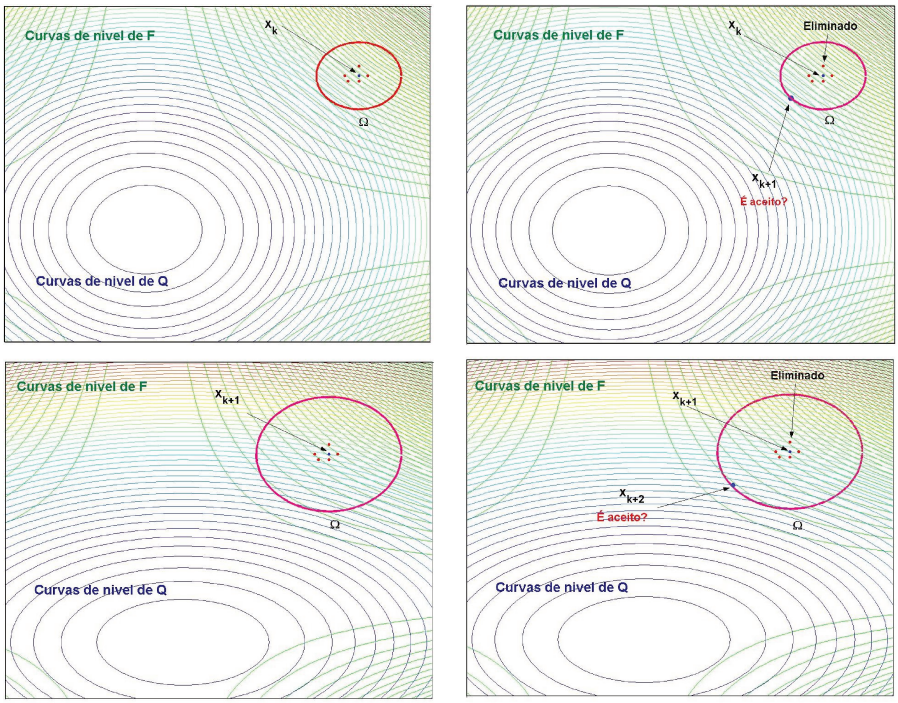

Figura 3.4. O método de Powell

À seguintes gráficas mostram alguns detalhes do método estudado:
Na *figura (1)* mostra-se o ponto inicial x_{beg}, os m pontos de interpolação, a função objetivo F e Q, o modelo quadrático, o qual é minimizado dentro duma região de confiança Ω. Na *figura (2)*, um novo ponto x_{k+1} é encontrado. Se ele satisfaz a condição de decréscimo suficiente da F, um ponto de interpolação é eliminado. Na *figura (3)*, um novo modelo quadrático Q, é requerido para ser minimizado dentro duma região de confiança de raio Δ com centro em x_{k+1}. Na *figura (4)*, um novo ponto x_{k+2} é encontrado. Se ele satisfaz a condição de decréscimo suficiente da F, um ponto de interpolação é eliminado.

CAPÍTULO 4

Um método de *busca padrão*

A maioria das idéias fundamentais
da ciência são essencialmente simples e,
por regra geral, podem ser dadas em uma
linguagem compreensível para todos.
Albert Einstein (1879-1955)

4.1 Introdução

Consideramos novamente o problema (1.1), de minimizar uma função continuamente diferenciável $F : \Re^n \to \Re$. Métodos de busca direta para este problema são métodos que não fazem uso do gradiente da função F e tampouco de aproximações do mesmo. Nosso interesse é, em particular, um subconjunto de métodos de busca direta chamados métodos de *busca padrão*. Um exemplo desse tipo de método é o algoritmo de busca multidirecional de Dennis e Torczon [6, 23], que foi introduzido em 1989 e que é baseado em um simplex de $n + 1$ pontos. Um outro exemplo, que será utilizado nesse trabalho, é o método de Virginia Torczon [24], que analisa a função objetivo F em um padrão de pontos, para então decidir qual será a próxima aproximação para o minimizador. O método de busca padrão de *direções coordenadas*, com tamanho de passo fixo, que propõe a autora, mostra que, para uma iteração k e uma aproximação x_k para a solução, são necessárias duas componentes para se definir um padrão: uma base $B \in \Re^{n\times n}$ e uma matriz geradora $C_k \in Z^{n\times p}$, onde $p > 2n$. Particionamos a matriz geradora

em componentes

$$C_k = [M_k \quad -M_k \quad L_k] = [\Gamma_k \quad L_k]\,. \tag{4.1}$$

com $M_k \in M \subset Z^{n\times n}$, onde M é um conjunto de matrizes não singulares e $L_k \in Z^{n\times(p-2n)}$ que contém uma coluna de zeros. O padrão P_k é então definido pelas colunas da matriz $P_k = BC_k$. Como B e C_k tem posto n, as colunas de P_k geram $\Re^n$. Por conveniência usamos a partição da matriz geradora C_k dada em (4.1) para particionar P_k como

$$P_k = BC_k = [BM_k \quad -BM_k \quad BL_k] = [B\Gamma \quad BL_k]\,. \tag{4.2}$$

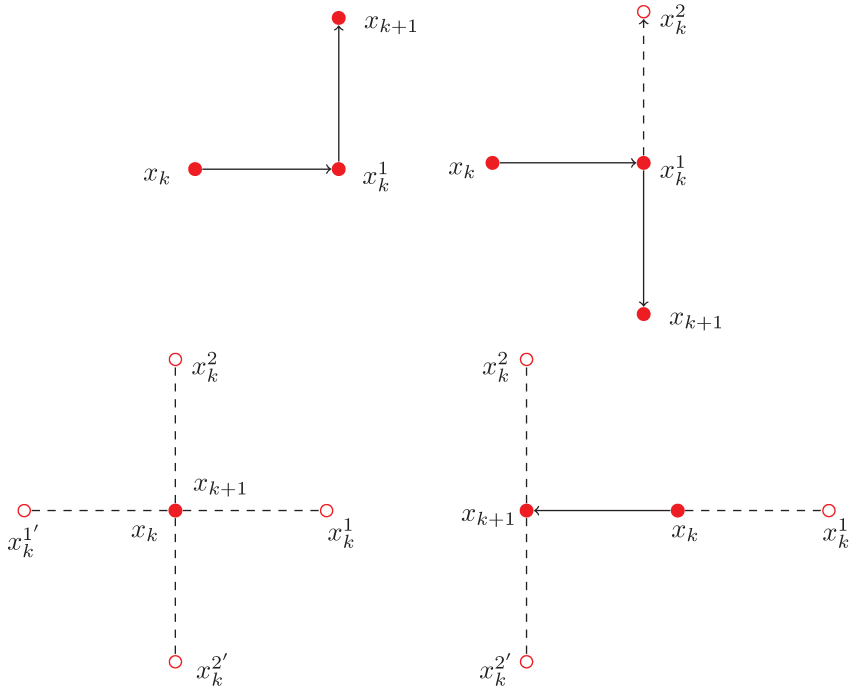

Figura 4.1. O padrão de *direções coordenadas* em $\Re^2$ com comprimento do passo Δ_k.

Dado o tamanho do passo $\Delta_k \in \Re^n$, $\Delta_k > 0$, definimos o passo s_k^i como sendo qualquer vetor da forma

$$s_k^i = \Delta_k B c_k^i, \tag{4.3}$$

onde c_k^i denota uma coluna de $C_k = \left[c_k^1 \cdots c_k^p\right]$. Observe que Bc_k^i determina a direção do passo, enquanto Δ_k é o comprimento do passo.

Em cada iteração k, definimos pontos da forma $x_k^i = x_k + s_k^i$, onde x_k é o ponto na iteração atual. Os candidatos a x_{k+1} são $x_{k+1}^i = x_k + \Delta_k Bc_k^i$. Uma única iteração de direções coordenadas quando $n = 2$, é mostrada na Fig. (4.1).

x_k^i é o ponto encontrado durante a iteraçãok, denotado por x_{k+1}. O círculo sólido indica sucesso no passo tomado, enquanto o círculo aberto indica pontos nos quais não houve decréscimo no valor da função objetivo. Na primeira figura vemos que houve decréscimo de x_k a x_k^1. No passo de x_k^1 a x_{k+1} também ocorreu um decréscimo no valor da função. A iteração termina com um novo ponto x_{k+1} que satisfaz a condição de decréscimo simples $F(x_{k+1}) < F(x_k)$. No pior caso, a penúltima gráfica da figura anterior, $2n$ pontos foram testados $x_k^1, x_k^{1'}, x_k^2$, e $x_k^{2'}$ sem produzir decréscimo no valor da função objetivo na iteração atual x_k. Neste caso, $x_{k+1} = x_k$ e o tamanho do passo deve ser reduzido na iteração seguinte.

4.2 Direções coordenadas com comprimento do passo fixo

O método de direções coordenadas com comprimento de passo fixo é um dos métodos mais simples entre os métodos de busca padrão. Este método utiliza outros nomes como, *direções alternadas e variação local*. Em seguida daremos alguns elementos importantes necessários para descrever o algoritmo geral.

4.2.1 As matrizes

Em direções coordenadas a matriz base é $B = I$, matriz identidade. A matriz geradora para direções coordenadas é fixada durante todas as iterações do método. A matriz geradora $C_k = C$ contém em suas colunas todas as possíveis combinações de $\{-1, 0, 1\}$. A matriz, C tem $p = 3^n$ colunas. Em particular, as colunas de C contêm I e $-I$, e também contêm colunas de zeros. Definimos $M = I$ e a matriz L, da equação (4.1) com $3^n - 2n$ colunas da matriz C. Como C é fixa em todas as iterações do método, não necessita ser atualizada.

Por exemplo, para $n = 2$, obtemos só a matriz C de direções coordenadas:

$$C = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} M & -M & L \end{bmatrix}$$

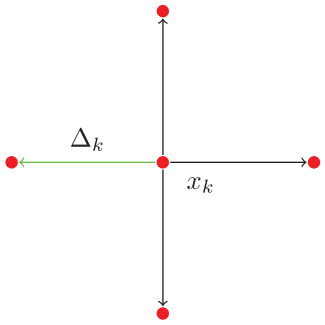

Figura 4.2. Matriz C de direções coordenadas.

A segunda, consideramos a matriz C que contém em suas colunas todas as possíveis combinações de $\{-1, 0, 1\}$, e cada coluna forma um passo padrão. Neste caso C tem 3^n colunas, como é mostrado na figura (4.3). Para $n = 2$, obtemos a matriz C:

$$C = \begin{bmatrix} 1 & 0 & -1 & 0 & 1 & 1 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 & 1 & -1 & -1 & 1 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} M & -M & L \end{bmatrix}$$

Logo, quando $n = 2$, todos os possíveis pontos definidos pelo padrão $P = BC$, para um comprimento do passo Δ_k, vistos na Fig. (4.2). Observe que o padrão inclui todos os possíveis pontos tratados na Fig. (4.1).

4.2.2 O movimento exploratório

Métodos de busca padrão conduzem a muitos movimentos exploratórios ao redor do vetor x_k. Estes movimentos são testados, verificando se o novo valor da função objetivo no ponto x_{k+1} é menor que no ponto x_k.

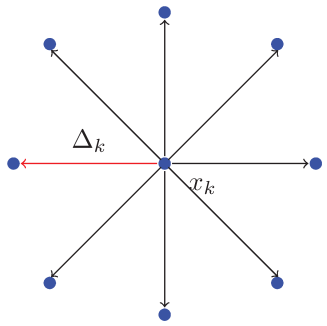

Figura 4.3. Matriz C, com 3^n colunas.

Cada método de busca padrão se distingue pela maneira na qual estes movimentos exploratórios são levados, esses, mantém as propriedades requeridas para provar a convergência dos métodos de busca padrão. Colocamos duas condições sobre os movimentos exploratórios dados na seguinte hipóteses.

Observação 4.2.1. Por notação: $y \in A$ significa que o vetor y está contido no conjunto dos vetores colunas da matriz A.

Hipóteses sobre o movimento exploratório irrestrito.

1. $s_k \in \Delta_k P_k \equiv \Delta_k BC_k \equiv \Delta_k \left[B\Gamma_k \quad BL_k\right]$.
2. Se $min\left\{F(x_k + y), y \in \Delta_k B\Gamma_k\right\} < F(x_k)$,
   então $F(x_k + s_k) < F(x_k)$.

A escolha do movimento exploratório deve garantir duas coisas:

1. A direção de qualquer passo s_k aceito em uma iteração k é definida pelo padrão P_k e seu comprimento é determinado por Δ_k.
2. Se o decréscimo simples sobre o valor da função na iteração atual pode ser encontrado entre qualquer dos $2n$ passos definidos por $\Delta_k B\Gamma_k$, então o movimento exploratório deve produzir um passo s_k que também dá um decréscimo simples sobre o valor atual da função. Em particular, $F(x_k + s_k)$ não precisa ser menor ou igual que

   $$min\left\{F(x_k + y), y \in \Delta_k B\Gamma_k\right\}.$$

   O movimento exploratório para direções coordenadas é dado no seguinte algoritmo, onde $e_i^{'}$s denota o vetor unitário coordenado.

Algoritmo 1
Dados de entrada:
x_k, Δ_k, $F(x_k)$, B, $s_k = 0$, $\rho_k = 0$ e $min = F(x_k)$.
Para $i = 1, \cdots, n$:
Fazer $s_k^i = s_k + \Delta_k B e_i$ e $x_k^i = x_k + s_k^i$.
Calcular $F(x_k^i)$.
Se $F(x_k^i) < min$ então $\rho_k = F(x_k) - F(x_k^i)$, $min = F(x_k^i)$, e $s_k = s_k^i$.
Caso contrário,
Fazer $s_k^i = s_k - \Delta_k B e_i$ e $x_k^i = x_k + s_k^i$. Calcular $F(x_k^i)$.
Se $F(x_k^i) < min$ então $\rho_k = F(x_k) - F(x_k^i)$, $min = F(x_k^i)$, e $s_k = s_k^i$.
Fim.

O movimento exploratório é executado no sentido de que a seleção do passo seguinte seja baseada no sucesso ou fracasso do passo anterior. Se há 3^n passos possíveis, nós podemos calcular um pouco menos de n passos, porém nosso cálculo não é mais que $2n$ em qualquer iteração dada, como vemos na Fig. (4.1).

Teoricamente, há duas condições que necessitam ser tratadas no algoritmo do movimento exploratório. A primeira, todos os possíveis passos devem estar contidos em $\Delta_k P$, que são mostrados nas Figuras (4.1) e (4.2). A segunda condição sobre o movimento exploratório, é muito interessante, já que as direções coordenadas mostram uma certa debilidade. Por exemplo, no primeiro gráfico da Fig. (4.1), vemos que o valor da função após o primeiro passo

$$s_k^1 = \Delta_k I \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

decresceu. No segundo passo

$$s_k^2 = \Delta_k I \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \Delta_k I \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \Delta_k I \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

como se mostra na figura (4.4).

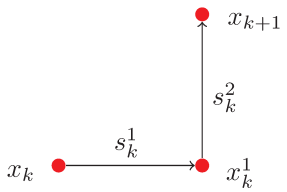

Figura 4.4. Definindo passos s_k^1 e s_k^2, com comprimento do passo Δ_k.

Decresceu ainda mais e portanto o passo foi aceito. É possível que grandes decréscimos no valor da função objetivo possam ser obtidos pelo passo:

$$s_k^{'} = \Delta_k I \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

o qual é definido por uma coluna da matriz M (o passo s_k^2 é definido pela coluna da matriz L), porém $s_k^{'}$ não é considerado quando um decréscimo simples é realizado pelo passo s_k^1. No entanto, no pior dos casos, como vemos na Fig. (3.1), o algoritmo para direções coordenadas garante que todos os $2n$ passos definidos por $\Delta_k B\Gamma = \Delta_k B\,[M - M] = \Delta_k B\,[I - I]$ são tratados depois de voltar ao passo $s_k = 0$. O movimento exploratório no algoritmo 1 analisa todos os $2n$ passos definidos por $\Delta_k B\Gamma$, a menos que satisfaz $F(x_k + s_k) < F(x_k)$ seja encontrado.

4.2.3 Método de busca padrão generalizado

O seguinte algoritmo generaliza o método de *busca padrão* para minimização irrestrita.

Algoritmo 2
Dados de entrada:
Seja $x_0 \in \Re^n$ e $\Delta_0 > 0$.
Para $k = 0, 1, \cdots,$
Calcular $F(x_k)$.
Determinar o passo s_k usando o algoritmo 1.
Calcular $\rho_k = F(x_k) - F(x_k + s_k)$.
Se $\rho_k > 0$ então $x_{k+1} = x_k + s_k$.
Caso contrário,
Fazer $x_{k+1} = x_k$.
Atualizar Δ_k.
Fim.

Para o nosso caso, utilizaremos o Algoritmo 1, que utiliza o movimento exploratório em direções coordenadas.

4.2.4 Atualizando o comprimento do passo

Tomando passos de mesmo tamanho até que não aconteça nenhum decréscimo na função objetivo, então o tamanho do passo é reduzido à metade. Isto corresponde a tomar $\theta = 1/2$ e $\Lambda = \{1\}$. Logo, $\tau = 2$, $\omega_0 = -1$, $\omega_1 = 0$. É suficiente verificar que direções coordenadas com comprimento de passo fixo corresponde a um método de *busca padrão*. (Ver teorema 3.5 [24])

Algoritmo 3 Atualizando Δ_k.
Dados de entrada:
$\tau \in \mathbb{Q}$, seja $\theta = \tau^{\omega_0}$ e $\lambda_k \in \Lambda = \{\tau^{\omega_1}, \cdots, \tau^{\omega_L}\}$,
onde $\tau > 1$ e $\{\omega_0, \omega_1, \cdots, \omega_L\} \subset \mathbb{Z}$,
$L \equiv |\Lambda| < +\infty$, $\omega_0 < 0$, $\omega_i \geq 0$, $i = 1, \cdots, L$.
Passo 1. Se $\rho_k \leq 0$ então $\Delta_{k+1} = \theta\Delta_k$.
Passo 2. Se $\rho_k > 0$, então $\Delta_{k+1} = \lambda_k\Delta_k$.

As condições sobre θ e Λ garantem que $0 < \theta < 1$ e $\lambda_i \geq 1$ para todo $\lambda_i \in \Lambda$. Se a iteração tem sucesso pode ser possível aumentar o tamanho do passo Δ_k; observamos que Δ_k não pode decrescer.

CAPÍTULO 5

Testes numéricos

A felicidade é a permanente
caminhada interior com
consciência, procurando fazer
a conquista de si mesmo.
L. J.

5.1 Introdução

Neste capítulo mostramos os resultados numéricos obtidos com o sofware de Powell: $UOBYQA$ e algumas variações do mesmo, e do método de Virginia Torczon. O primeiro deles usa, na k-ésima iteração, um conjunto de pontos de interpolação, para construir o polinômio interpolante Q da função objetivo F, a aproximação ao gradiente e a hessiana do modelo quadrático Q, minimiza o subproblema no contexto de região de confiança, por uns dos métodos descritos no capítulo 2, e procura um decréscimo suficiente do valor da função F. O segundo usa, na k-ésima iteração, um conjunto padrão P_k de direções e procura um decréscimo simples da função F. Os diferentes problemas irrestritos resolvidos, foram tirados da coleção de Hock e Schittkowski [8].

5.2 Testando uma função

Nesta seção apresentamos resultados do método $UOBYQA$, com as seguintes modificações: (i) o parâmetro inicial $\rho_{beg} = 0.2x_k(1)$ é trocado por $\rho_{beg} = 0.2$ testando valores no intervalo $(0, 1)$ e concluindo que o valor encontrado permite iniciar com um bom passo para determinar o x_k na primeira iteração, (ii) $\frac{1}{6}M$ parâmetro não negativo, é trocado $|0.5F(x_k)|$ que determina uma cota de erro do modelo quadrático em cada iteração garantindo que Q seja uma boa aproximaçãopara a função F e (iii) o número de pontos de interpolação $m = 2n+1$ é trocado por $m = \frac{1}{2}(n+1)(n+2)$, onde x_k é o novo vetor obtido na k-ésima iteração que verifica se o modelo quadrático Q é uma boa aproximação para a função objetivo F no ponto x_k.

Vejamos inicialmente, para $n = 2$, os resultados obtidos para a função de *Rosenbrock*:

$$F(x, y) = 100(x - y^2)^2 + (1 - x)^2, \ x \in \Re^n \tag{5.1}$$

implementada em Fortran, com os valores:
$\rho_{beg} = 0.2$
$\rho_{end} = 10^{-8}$
$\Delta = 1 \geq 0$ raio da região de confiança.
Valor da tolerância $tol = 10^{-8}$ (o algoritmo pára quando ρ_{end} é menor ou igual a tol)
O ponto inicial $x_{beg} = (-1.2, 1)$
$m = \frac{1}{2}(n + 1)(n + 2) = 6$, pontos de interpolação.
$\frac{1}{6}M = 0$ valor inicial.
O número máximo de avaliações de função é $MAXFUN = 5000$.

As tabelas (5.1) e (5.2) mostram os resultados para $UOBYQA_{mod}$ e $NEWUOA$. Testando a função de *Rosenbrock*, com os mesmos valores iniciais, o algoritmo $NEWUOA$ é um sofware cedido por Powell 2006 [19], com $m = 2n + 1 = 5$ pontos de interpolação.

Usamos como medidas de comparação o número de avaliações de função e o valor ótimo de F para cada valor de ρ. Com estas medidas, concluímos que o método $UOBYQA_{mod}$ teve um melhor desempenho que $NEWUOA$. No entanto, para n grande, os sistemas resolvidos por $NEWUOA$ são muito menores que os resolvidos por $UOBYQA_{mod}$, o que deixa o tempo computacional como sendo uma boa medida de desempate da eficiência dos dois métodos. Poderíamos ainda usar o número de iterações efetuadas por cada

Função de Rosenbrock $n = 2, \quad m = 6$		
Avaliações de F	*O Melhor valor de F*	ρ
8	3.70141090	$2 * 10^{-2}$
50	4.37898483 x 10^{-1}	$2 * 10^{-3}$
116	1.85413121 x 10^{-6}	$2 * 10^{-4}$
119	5.62928100 x $2 * 10^{-9}$	10^{-5}
122	3.38712487 x 10^{-12}	10^{-6}
122	3.38712487 x 10^{-12}	$1.414 * 10^{-7}$
130	1.96148329 x 10^{-20}	10^{-8}

Tabela 5.1. Solução do problema (5.1) utilizando $UOBYQA_{mod}$.

Função de Rosenbrock $n = 2, \quad m = 5$		
Avaliações de F	*O Melhor valor de F*	ρ
10	8.831410	10^{-1}
15	8.831410	10^{-2}
20	6.112364	10^{-3}
25	6.099923	10^{-4}
250	5.647275 x 10^{-10}	10^{-5}
254	4.070714 x 10^{-10}	10^{-6}
266	2.499902 x 10^{-15}	10^{-7}
270	2.222798 x 10^{-17}	10^{-8}
278	7.093898 x 10^{-24}	10^{-9}

Tabela 5.2. Solução do problema (5.1) utilizando *NEWUOA*

método, para obter a mesma precisão, como medida de comparação.

Uma observação pertinente, feita pelo próprio autor, é a limitação do número de variáveis da função F com que os métodos $UOBYQA$, ou $UOBYQA_{mod}$, podem trabalhar: $m_1 = \frac{1}{2}(n+1)(n+2)$. Comparando com $NEWUOA$, que usa $m_2 = 2n + 1$ vetores, vemos que $m_1 > m_2$ para todo $n > 1$ e assim, para $n > 20$ já se torna impossível o uso de $UOBYQA$, ou $UOBYQA_{mod}$. Por exemplo, se $n = 30$, $m_1 = 496$, ao passo que $m_2 = 61$.

Observação 5.2.1. Quando $F(x^*) = 0$, consideraremos sucesso, se o valor da função objetivo no ponto final do algoritmo de otimização for menor que 10^{-9}.

5.3 Os problemas da coleção de *Hock e Schittkowski*

Para os algoritmos testados, usamos problemas propostos na coleção de *Hock e Schittkowski* [8]. Tratase de um conjunto de problemas clássicos e diferenciáveis de minimização irrestrita. Os nomes dos problemas estão abreviados com o nome de cada autor e enumerados para melhor identifição. Nas tabelas a seguir, mostramos o número de avaliações da função (*MAXFUN*), que cada algoritmo efetua para resolver o problema, e também o valor final da função objetivo que foi alcançado. A "*" indica que o algoritmo termina porque o limite de *MAXFUN = 5000*, foi alcançado e "**" indica que o algoritmo não fornece solução, o que se pode ver na tabela (5.3). A comparação entre os algoritmos é baseada novamente no número de avaliações necessárias para alcançar a mesma precisão.

Verificamos ainda, realizando testes numéricos que, para as funções ROS[1], POW[13] e WOO[14] o algoritmo *UOBYQA* encontra uma solução melhor depois de mais alguns avaliações da função o que é feito escolhendo um ρ_{end} menor.

O algoritmo *UOBYQA* utiliza o método de Moré e Sorensen para a resolução do subproblema de região de confiança. Este algoritmo é numericamente estável e dá resultados muito bons de precisão. A função F é fornecida pelo usuário através de uma subrotina chamada CALFUN, em ambos algoritmos.

A tabela (5.3) mostra os resultados obtidos com o algoritmo $UOBYQA_{mod}$ e o *UOBYQA* original, com $m = \frac{1}{2}(n+1)(n+2)$ e $m = 2n+1$ pontos de interpolação respectivamente. De um total de 23 funções testadas, 86.95% convergem para o ponto ótimo da função com $UOBYQA_{mod}$, enquanto que, com *UOBYQA*, apenas 60.86% convergem à solução. Observemos que as dimensões dos problemas são pequenas (≤ 20).

O algoritmo $UOBYQA_{mod}$ apresenta uma melhor precisão na solução dos problemas testados. O número de pontos de interpolação é importante para o desempenho desses algoritmos, pois é com eles que construímos o primeiro modelo quadrático $Q \approx F$. Além disso eles servem para garantir que o sistema a ser resolvido seja não singular.

Na tabela (5.4), mostramos o algoritmo *NEWUOA* que Powell implementou em Fortran 77 com precisão dupla, o qual comparamos com $UOBYQA_{mod}$.

Nome	*Dimensão* $n =$	*Avaliações de F*		*Valor final da F*	
		$UOBYQA_{mod}$ $m = \frac{1}{2}(n+1)(n+2)$	$UOBYQA$ $m = 2n+1$	$UOBYQA_{mod}$ $m = \frac{1}{2}(n+1)(n+2)$	$UOBYQA$ $m = 2n+1$
ROS[1]	2	131	**	1.9614 x 10^{-20}	**
FRE[2]	2	71	**	4.8984 x 10^{1}	**
POW[3]	2	910	28	7.2828 x 10^{-26}	1.1023 x 10^{-2}
BEA[5]	2	54	105	8.0689 x 10^{-19}	1.5447 x 10^{-1}
JEN[6]	2	41	82	5.0487 x 10^{-29}	1.0742 x 10^{-13}
BAR[8]	3	77	609	8.2148 x 10^{-3}	1.1723 x 10^{-2}
GAU[9]	3	31	1420	1.5883 x 10^{-2}	1.5901 x 10^{-2}
MEY[10]	3	4043	**	8.7945 x 10^{-2}	**
GUL[11]	3	122	95	0	0
BOX[12]	3	140	**	2.2247 x 10^{-26}	**
POW[13]	4	512	**	2.4200 x 10^{-30}	**
WOO[14]	4	391	919	9.7698 x 10^{-20}	0.0412
OSB[17]	5	965	**	5.4648 x 10^{-5}	**
BIG[18]	6	606	*	5.7291 x 10^{-23}	1.8300 x 10^{-2}
OSB[19]	11	*	**	6.0425 x 10^{-2}	**
EXT[21]	10	1089	**	5.2996 x 10^{-17}	**
EXT[22]	12	1796	*	8.3799 x 10^{-39}	2.1203 x 10^{-3}
PII[24]	12	2012	3203	1.7949 x 10^{-31}	1.0190 x 10^{-15}
VAR[25]	12	831	*	9.3118 x 10^{-28}	9.2825 x 10^{-3}
DIS[28]	12	191	*	8.0752 x 10^{-31}	1.3449 x 10^{-3}
BRO[30]	12	65	*	1.6393 x 10^{-30}	1.8289 x 10^{-2}
BRY[31]	12	446	**	2.1746 x 10^{-14}	**
LIN[33]	12	924	503	2.1428	2.1428

Tabela 5.3. Testando os problemas com $UOBYQA_{mod}$ e $UOBYQA$.

Para alguns problemas da coleção de *Hock e Schittkowski*, o algoritmo *NEWUOA* é testado também (com $m = 2n + 1$ pontos de interpolação).

O algoritmo *NEWUOA* permite que o valor de n, o número de variáveis, seja grande, devido à quantidade de trabalho por iteração permitindo alcançar uma boa precisão na solução dos problemas, como se pode ver, na tabela (5.4), nos problemas ROS[1], JEN[6], BIG[18] e LIN[33]. Nos problemas OSB[17], EXT[22], VAR[25], a solução foi alcançada com o número máximo, *MAXFUN = 5000*, de avaliações da função F, comparado com o algoritmo $UOBYQA$.

5.4 Comparando com o algoritmo de *busca padrão*

Iniciamos, implementando o algoritmo de *busca padrão* em Fortran 90, com precisão dupla, utilizando a matriz C das direções coordenadas, dada na

Nome	*Dimensão* $n =$	*Avaliações da F*		*Valor final da F*	
		$UOBYQA_{mod}$ $m = \frac{1}{2}(n+1)(n+2)$	$NEWUOA$ $m = 2n+1$	$UOBYQA_{mod}$ $m = \frac{1}{2}(n+1)(n+2)$	$NEWUOA$ $m = 2n+1$
ROS[1]	2	131	184	1.9614 x 10^{-20}	4.4373 x 10^{-31}
FRE[2]	2	71	125	4.8984 x 10^{1}	4.8984 x 10^{1}
BEA[5]	2	54	115	4.4373 x 10^{-31}	4.930 x 10^{-32}
JEN[6]	2	41	82	5.0487 x 10^{-29}	0
BAR[8]	3	77	219	8.2148 x 10^{-3}	1.1723 x 10^{-3}
GAU[9]	3	31	108	1.5883 x 10^{-2}	1.5883 x 10^{-2}
GUL[11]	3	122	104	0	0
POW[13]	4	512	1080	2.4200 x 10^{-30}	5.3420 x 10^{-21}
OSB[17]	5	965	$*$	5.4648 x 10^{-5}	5.7018 x 10^{-5}
BIG[18]	6	606	2482	5.7291 x 10^{-23}	2.0812 x 10^{-30}
EXT[22]	12	1796	$*$	8.3799 x 10^{-39}	8.6078 x 10^{-39}
PII[24]	20	$*$	2107	1.7949 x 10^{-31}	7.7037 x 10^{-34}
VAR[25]	12	831	$*$	9.3118 x 10^{-28}	2.9412 x 10^{-10}
LIN[33]	20	924	733	2.1428	2.1428

Tabela 5.4. Testando os problemas com $UOBYQA_{mod}$ e $NEWUOA$.

seção (4.3.2), com 3^n colunas. Por exemplo para $n = 2$, a função de *Rosenbrock*:

$$F(x, y) = 100(x - y^2)^2 + (1 - x)^2,\ x \in \Re^n$$

com valores iniciais:
O ponto inicial $x_{beg} = (-1.2, 1)$.
$\Delta_k = 1$ o tamanho de passo inicial.
O valor de θ que reduz o tamanho do passo, foi $\theta = \frac{1}{2}$.
O valor de λ que amplia o tamanho do passo foi $\lambda = 1$.
Valor da tolerância $tol = 10^{-6}$. O algoritmo pára quando o tamanho do passo é menor que tol.

Entre com a dimensão do espaço ($n =$)	2
Entre com o ponto inicial	(−1.2 1)
Matriz de direções coordenadas	$\begin{pmatrix} 1 & -1 & 0 & 0 & 1 & 1 & -1 & -1 & 0 \\ 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 0 \end{pmatrix}$
Minimizador da função	(0.999952853 0.999905527)
Valor mínimo da F	2.22614283E-009
Avaliações de F efetuadas	24822

Testamos os problemas da coleção Hock, W. e Schittkowski [8], onde fazemos a comparação do algoritmo de busca padrão: *BUSPAD.f*, com o algoritmo $UOBYQA_{mod}$. Este tipo de algoritmo, *BUSPAD*, analisa a função objetivo em um padrão de pontos, para então decidir qual será a próxima aproximação para o minimizador. Neste caso, definimos o padrão tomando a matriz C

que tem 3^n colunas, obtendo outras possíveis direções na procura da solução.

A tabela (5.5) a seguir, mostra os dados obtidos:

Nome	*Dimensão* $n =$	*Avaliações da F*		*Valor final da F*	
		$UOBYQA_{mod}$ $m = \frac{1}{2}(n+1)(n+2)$	$BUSPAD$	$UOBYQA_{mod}$ $m = \frac{1}{2}(n+1)(n+2)$	$BUSPAD$
ROS[1]	2	131	24822	1.9614×10^{-20}	2.2261×10^{-9}
FRE[2]	2	71	102	4.8984×10^{1}	4.8963×10^{1}
POW[3]	2	910	1302	7.2828×10^{-26}	5.5902×10^{-7}
BEA[5]	2	54	1555	8.0689×10^{-19}	1.3109×10^{-12}
JEN[6]	2	41	163	5.0487×10^{-29}	3.6379×10^{-12}
BAR[8]	3	77	205	8.2148×10^{-3}	4.3587×10^{-4}
GAU[9]	3	31	75	1.5883×10^{-2}	2.5689×10^{-3}
MEY[10]	3	4043	141591	8.7945×10^{-1}	8.7943×10^{-1}
GUL[11]	3	122	3564	0	4.2247×10^{-31}
BOX[12]	3	140	4321	2.2247×10^{-26}	1.2897×10^{-17}
POW[13]	4	512	377961	2.4200×10^{-30}	3.4773×10^{-9}
WOO[14]	4	391	52136	9.7698×10^{-20}	8.7125×10^{-10}
OSB[17]	5	965	132678	5.4648×10^{-5}	4.4778×10^{-4}
BIG[18]	6	606	18654	5.7291×10^{-23}	2.4786×10^{-15}
OSB[19]	11	*	345268	6.0425×10^{-2}	5.5376×10^{-5}
EXT[21]	10	1089	510348	5.2996×10^{-17}	5.8983×10^{-9}
EXT[22]	12	1796	411874	8.3799×10^{-39}	9.3673×10^{-10}
PII[24]	12	2012	51973	1.7949×10^{-31}	5.01789×10^{-13}
VAR[25]	12	831	623975	9.3118×10^{-28}	2.1826×10^{-10}
DIS[28]	12	191	328005	8.0752×10^{-31}	6.7791×10^{-11}
BRO[30]	12	65	4082	1.6393×10^{-30}	4.1620×10^{-13}
BRY[31]	12	446	6888	2.1746×10^{-14}	1.2804×10^{-13}
LIN[33]	12	924	56432	2.1428	2.1428

Tabela 5.5. Comparando $UOBYQA_{mod}$ com $BUSPAD$

O algoritmo de *BUSPAD.f* comparado com o algoritmo $UOBYQA_{mod}$ precisa de mais avaliações da função para chegar à solução ótima do problema ou a uma aproximação dela, como se analisa na função GUL[11], onde o valor final da F é de 4.2247×10^{-31} e no algoritmo $UOBYQA_{mod}$ o valor é de zero. Isso garante que o algoritmo, com os problemas testados, apresenta uma melhor precisão na solução dos problemas. Embora seja uma conclusão verdadeira, não podemos dizer que os métodos de Powell são os melhores sempre, pois trata-se de métodos com filosofia de minimização diferentes, e com complexidade computacional diferente, também. Enfim, podemos concluir que, dependendo do problema, um método pode realmente ser mais eficiente e mais conveniente que o outro.

5.5 Função de Weber

A função de Weber modela o problema de dizer onde deve se localizar um depósito ou produtor de algum bem de modo que o custo associado à distribução desse bem aos diversos interessados seja o menor possível. O custo é proporcional à distância dos interessados ao produtor. Se os locais onde há demanda forem $\vartheta_i \in \Re^2$ com os pesos de custo correspondentes ω_i, a função é dada por

$$f(x) = \sum_i \omega_i \|x - \vartheta_i\| = \sum \omega_i \sqrt{((x)_1 - (\vartheta_i)_1)^2 + ((x)_2 - (\vartheta_i)_2)^2}.$$

Assumiremos que $\sum_i \omega_i > 0$ para garantir a existência de mínimo global. Algum dos pesos ω_i pode ser negativo, e nesse caso, pode haver mais de um minimizador local. A função é não diferenciável para $x = \vartheta_i$, daí o interesse em verificar o comportamento dos algoritmos nesse problema.

Os algoritmos $UOBYQA_{mod}$, foi testado no dois exemplos:

1. $\omega = (2, 4, -5)^T$ e

$$(\vartheta_1, \vartheta_2, \vartheta_3) = \begin{bmatrix} 2 & 90 & 43 \\ 42 & 11 & 88 \end{bmatrix}.$$

2. $\omega = (2, -4, 2, 1)^T$ e

$$(\vartheta_1, \vartheta_2, \vartheta_3) = \begin{bmatrix} -10 & 0 & 5 & 25 \\ -10 & 0 & 8 & 30 \end{bmatrix}.$$

Para o primeiro exemplo, há um mínimo global em $x^* = (90, 11)^T$. Já para o segundo existem dois minimizadores: $(-10, -10)^T$ e $(25, 30)^T$, sendo $(25, 30)^T$ minimizador global.

O algoritmo foi testado usando a origem como aproximação inicial. Apesar da descontinuidade do problema, o algoritmo encontra os minimizadores globais dos exemplos, apresentando os seguintes resultados:

1. ***Exemplo 1***
   Ponto inicial: $(0, 0)$
   Valor da F no ponto inicial: 354.19364.
   Results with $N = 2$

New RHO = 2.0000E-02 Number of function values = 55
Least value of F = $-2.652626866513363E+02$
The corresponding X is: $(8.998723E+01, 1.098719E+01)$.

New RHO = 2.0000E-03 Number of function values = 61
Least value of F = $-2.653381643692633E+02$
The corresponding X is: $(9.003426E+01, 1.101163E+01)$.

New RHO = 1.4142E-04 Number of function values = 72
Least value of F = $-2.653740584879699E+02$
The corresponding X is: $(9.000290E+01, 1.100064E+01)$.

New RHO = 1.0000E-05 Number of function values = 91
Least value of F = $-2.653770366150014E+02$
The corresponding X is: $(9.000003E+01, 1.099998E+01)$.

At the return from $UOBYQA_{mod}$ Number of
function values = 103
Least value of F = $-2.653770814014357E+02$
The corresponding X is: $(9.000001E+01, 1.100000E+01)$.

2. ***Exemplo 2***
Ponto inicial: $(0, 0)$
Valor da F no ponto inicial: 86.2034819
Results with $N = 2$

New RHO = 2.0000E-02 Number of function values = 44
Least value of $F = 9.591743089960302E+00$
The corresponding X is: $(2.497445E+01, 2.998087E+01)$.

New RHO = 2.0000E-03 Number of function values = 49
Least value of $F = 9.572843986409843E+00$
The corresponding X is: $(2.500237E+01, 3.001277E+01)$.

New RHO = 1.4142E-04 Number of function values = 58
Least value of $F = 9.561033688141453E+00$

The corresponding X is: $(2.499991E+012.999974E+01)$.

New RHO = 1.0000E-05 Number of function values = 66
Least value of $F = 9.560771752906831E+00$
The corresponding X is:$(2.499999E+01, 3.000003E+01)$.

At the return from $UOBYQA_{mod}$ Number of function values = 74
Least value of $F = 9.560740504928926E+00$
The corresponding X is: $(2.500000E+01, 3.000000E+01)$.

Quanto ao algoritmo $UOBYQA_{mod}$, verificamos que ele apresenta uma pequena vantagem na relação ao algoritmo $UOBYQA$, se nos referimos ao número de avaliações da função. O algoritmo $UOBYQA_{mod}$ converge para mais problemas, mostrado principalmente em a qualidade do critério de decréscimo suficiente escolhido pelo autor.

O algoritmo *UOBYQA* fracassou em testes realizados em uma função estritamente convexa e não diferenciável, o que indica que talvez não seja possível obter resultados teóricos de convergência para funções não diferenciáveis. O algoritmo de *BUSPAD* apresentou resultados indesejáveis para alguns problemas diferenciáveis, como já era previsto, porém se espera que problemas que são o resultado duma simulação o algoritmo de Virginia Torczon [24] funcione melhor.

CAPÍTULO 6

Conclusões

Só quem persevera é capaz de
imaginar e criar na ciência
os sonhos mais difíceis
e maravilhosos.
L.J.

No presente trabalho estudamos dois métodos para solucionar o problema

$$\begin{array}{ll} \text{minimizar} & F(x) \\ x \in \Re^n & \end{array}$$

que não usam o gradiente da função F. $F : \Re^n \to \Re$ é uma função continuamente diferenciável em $\Re^n$. Basicamente, podemos dizer que o primeiro, *UOBYQA*, proposto por Powell em 2002, [17], e suas modificações constituiu o tema fundamental do trabalho. O segundo, proposto por Virginia Torczon em 1997 [23], foi utilizado para comparar os resultados obtidos. Introduzimos uma pequena modificação em *UOBYQA* e testamos com o sofware *NEWUOA* [19] cedido por o autor, o qual é a última versão do mesmo.

1. O algoritmo *UOBYQA* constrói um modelo quadrático $Q \approx F$ no inicio de cada iteração e trabalha num procedimento de região de confiança para ajustar as variáveis. Quando Q é atualizado, então o novo Q interpola F em m pontos, onde $m = \frac{1}{2}(n+1)(n+2)$. Só um ponto de interpolação é trocado em cada iteração. Muitas questões foram tratadas durante o desenvolvimento deste trabalho, para alcançar uma

boa precisão. Elas incluem a escolha do modelo quadrático inicial, a necessidade de manter a condição de interpolação não singular, com a presença de erros de arredondamento do computador, e a estabilidade das atualizações do modelo Q, das funções de Lagrange e dos raios de região de confiança Δ e ρ.

2. O algoritmo *UOBYQA*, com as modificações efetuadas, *UOBYQA* $_{mod}$, apresenta uma melhor precisão, de um modo geral. A primeira característica deste algoritmo é que o valor inicial de $\rho_{end} = 0.2$ permite iniciar com um bom passo para determinar o valor de x_k na primeira iteração. A segunda característica, é o valor da constante $\frac{1}{6}M$, que determina uma cota de erro do modelo quadrático em cada iteração garantindo que Q seja uma boa aproximação para a função F.

3. O número de avaliações da função objetivo F, $m = n = 1/2(n+1)(n+2)$ que *UOBYQA* e $UOBYQA_{mod}$ precisam para construir o primeiro polinômio de interpolação é muito grande, em geral. No algoritmo *NEWUOA* [19], Powell trata deste assunto, e mostra que uma boa aproximação para a solução do problema (1.1) é encontrada com apenas $m = 2n + 1$ pontos de interpolação.

4. Os dois algoritmos *UOBYQA* e *NEWUOA* são feitos em Fortran por Powell, e estão disponíveis na internet pelo autor.

5. De acordo com a definição de *derivative-free*, podemos classificar todos os algoritmos estudados e implementados, como métodos *derivative-free*, já que nenhum deles usa o gradiente da função F no desenvolvimento do algoritmo. O método de Virginia Torczon pode ser classificado como método de busca direta *derivative-free*, já que ele só usa comparações dos valores da função F como estratégia para definir o próximo ponto factível com melhor valor, utilizando uma estratégia de decréscimo simples para determinar o próximo ponto factível com melhor valor da função F.

Os próximos passos a seguir neste trabalho são:

1. Tentar medir o tempo de execução de cada algoritmo.

2. Programar $NEWUOA_{mod}$, com as mesmas modificações feitas em $UOBYQA_{mod}$ e comparar os resultados obtidos.

3. Terminar a programação dos métodos em MatLab. Esta parte do trabalho está bastante adiantada, mas ainda não conseguimos rodar os

programas a contento.

Este trabalho teve como resultados: (i) um grande amadurecimento em termos de pesquisa, haja visto a bibliografia consultada e o total entendimento dos métodos estudados. (ii) Uma outra proposta que consideramos totalmente atingida foi uma prática exaustiva de programação, onde pudemos praticar o uso de um software, o MatLab, bem como aprender a programar em uma linguagem científica, o Fortran. (iii) Salientamos ainda a experiência em comparar resultados numéricos obtidos por diferentes algoritmos.

Finalmente, gostaríamos de agradecer ao Professor M. J. D. Powell, pela sua atenção e prontidão em nos atender, quando entramos em contato com ele, a fim de obter os softwares *UOBYQA* e *NEWUOA*, de sua autoria.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] Bertsekas, D. P., 1995. *"Nonlinear programing"*, Athena Scientific, second edition.

[2] Cline, A. K.,C. B. Moler, G. W. Stewart and J. H. Wilkinson. 1979. *"An estimate for the condition number of a matrix"*, SIAM Journal Numerical Analysis, vol. 16, pp. 368-375.

[3] Conn A. R., N. I. M. Gould and Ph. L. Toint. 2000. *"Trust-Region Methods"*, MPS-SIAM Series on Optimization. Philadelfia, USA.

[4] Coope I. D. and C. J. Price. 2001. *"On the convergence of grid-based methods for unconstrained optimization"*, SIAM Journal on Optimization, vol. 11, pp. 859-869.

[5] Cunha C. M. Cristina. 2003. *"Métodos numéricos"*, Editora da Unicamp, Campinas, SP. Segunda edição revisada. Cap. No. 5, pp. 93-120.

[6] Dennis J. E. and V. Torczon, 1991. *"Direct search methods on parallel machines"*, SIAM Journal on optimization, vol. 1, No. 4, pp. 448-474.

[7] Higham J. Nicholas. 1987. *"A survey of condition number estimation for triangular matrices"*, SIAM Review, vol. 29, No. 4, pp. 575-596.

[8] Hock, W. and Schittkowski, K., 1981. *"Test examples for nonlinear programming codes"*, Springer, Berlin.

[9] Hooke, R. and Jeeves, T. A., 1961. *"Direct search solution of numerical and statistical problems"*, Journal of the Association for Computing Machinery 8, pp.212-229.

[10] Moré Jorge J. and D.C. Sorensen. 1983. *"Computing a trust-region step"*, SIAM journal Scientific Statistical Computation, vol. 4, pp. 553-572.

[11] Nelder J. A. and R. Mead, 1965. *"A simplex method for function minimization"*, Computer Journal, vol. 7, pp. 308-313.

[12] Nocedal, J. and Wright S. J., 1999. *"Numerical optimization"*, Springer Series in Operations Research.

[13] Powell, M. J. D. 1994. *"A direct search optimization method that models the objective and constraint functions by linear interpolations"*, Advances in Optimization and Numerical Analysis, Proceeding of the Sixth Workshop on Optimization and Numerical Analysis, Oaxaca, México, vol. 275, pp. 51-67.

[14] Powell, M. J. D. 1994. *"A direct search optimization method that models the objective by quadrátic interpolation"*, Presentation at the 5th Stockholm Optimization Days, Stockholm.

[15] Powell, M. J. D. 1998. *"Direct search algoritms for optimization and calculations"*, Acta Numerica, pp. 287-336.

[16] Powell, M. J. D. 2001. *"On the Lagrange functions of quadratic models that are defined by interpolation"*, Optimization Methods Software, vol. 16, pp. 289-309.

[17] Powell, M. J. D. 2002. *"UOBYQA: unconstrained optimization by quadratic approximation"*, Mathematical Programming, vol. 92, pp. 555-582.

[18] Powell, M. J. D. 2003. *"On trust region methods for unconstrained minimization without derivative"*, Mathematical Programming, vol. 97, pp. 605-623.

[19] Powell, M. J. D. 2006. *"The NEWUOA sofware for for unconstrained minimization without derivatives"*, In Large-Scale Nonlinear

Optimization, eds. G. Di Pillo and M. Roma, Springer (New York), pp. 255-297.

[20] Price C. J., I. D. Coope and D. Byatt, 2002. *"A convergent variant of the Nelder-Mead algoritm"*, Journal of Optimization Theory and Applications, vol. 113, No. 1, pp. 5-19.

[21] Schittkowski, K. 1987. *"More test examples for nonlinear programming codes"*, Springer, Berlin.

[22] Scheinberg, Katya. 1997. *"Derivative free optimization method"*, IBM Watson Research Center. pp. 1-10.

[23] Torczon V., 1991. *"On the convergence of the multidirectional search algorithms"*, SIAM Journal on Optimization, Vol 1, No. 1, pp.123-145.

[24] Torczon V., 1997. *"On the convergence of pattern search algorithms"*, SIAM Journal on optimization, vol. 7, No. 1, pp. 1-25.

[25] Th. Sauer and Yuan Xu. 1995. *"On multivariate Lagrange interpolation"*. Mathematics of Computation, vol. 64, pp. 1147-1170.

[26] Winfield D., 1969. *"Function and functional optimization by interpolation in data tables"*, PhD thesis, Harvard University, Cambridge, USA.

[27] Watkins David S. 1991. *"Fundamentals of matrix computations"*, Jhon Wiley & Sons, Washington State University, pp. 210.

[28] Winfield D., 1973. *"Function minimization by interpolation in a data table"*, Journal of the Institute of Mathematics and its Applications, vol. 12, pp. 339-347.