

UNIVERSIDADE FEDERAL DO PARANÁ
Adriano Verdério

**CONVERGÊNCIA GLOBAL DE UM MÉTODO DE REGIÃO DE
CONFIANÇA SEM DERIVADAS COM MODELOS BASEADOS EM
INTERPOLAÇÃO POLINOMIAL**

Curitiba, 25 de fevereiro de 2011.

UNIVERSIDADE FEDERAL DO PARANÁ
Adriano Verdério

**CONVERGÊNCIA GLOBAL DE UM MÉTODO DE REGIÃO DE
CONFIANÇA SEM DERIVADAS COM MODELOS BASEADOS EM
INTERPOLAÇÃO POLINOMIAL**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Matemática Aplicada da Universidade Federal do Paraná, como requisito parcial à obtenção do Título de Mestre em Matemática Aplicada.

Orientadora: Profa. Dra. Elizabeth Wegner Karas.
Coorientador: Prof. Dr. Lucas Garcia Pedroso.

Curitiba, 25 de fevereiro 2011.

UNIVERSIDADE FEDERAL DO PARANÁ
Adriano Verdério

**CONVERGÊNCIA GLOBAL DE UM MÉTODO DE REGIÃO DE
CONFIANÇA SEM DERIVADAS COM MODELOS BASEADOS EM
INTERPOLAÇÃO POLINOMIAL**

Profa. Dra. Elizabeth Wegner Karas
Orientadora - Universidade Federal do Paraná - UFPR

Prof. Dr. Lucas Garcia Pedroso
Coorientador - Universidade Federal do Paraná - UFPR

Profa. Dra. Maria Aparecida Diniz Ehrhardt
Universidade Estadual de Campinas - UNICAMP

Prof. Dr. José Mario Martínez
Universidade Estadual de Campinas - UNICAMP

Curitiba, 25 de fevereiro 2011.

Aos meus pais e irmãos, por mesmo longe conseguirem estar sempre perto.

Agradecimentos

Agradeço aos meus orientadores, professora Elizabeth e professor Lucas, pela cobrança, pela orientação e apoio no desenvolvimento desta dissertação e por todos os conhecimentos comigo compartilhados. E por principalmente acreditarem em mim, em horas que nem eu acreditava.

Agradeço ao professor Ademir que muito contribuiu para esse trabalho. E aos professores Mario e Cheti por aceitarem participar de minha banca e pelas sugestões e contribuições ao trabalho.

Aos professores que contribuíram para a minha formação acadêmica.

Aos meus pais, por me ensinar valores e me ajudarem a chegar onde estou.

Ao antigo professor e hoje colega Paulo Conejo, pelas dúvidas compartilhadas e outras sanadas.

Aos amigos que encontrei em Curitiba durante o mestrado, às horas que passamos estudando juntos.

Aos amigos que deixei em Cascavel.

Aos amigos que fazem parte da minha vida, longe ou perto.

Ao Programa de Pós-Graduação em Matemática Aplicada da UFPR pela formação de qualidade propiciada.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico, pelo apoio financeiro, sem o qual dificilmente conseguiria chegar até aqui.

*“Não fosse isso
e era menos.
Não fosse tanto
e era quase.”*

Paulo Leminski

Resumo

Apresentamos um método de região de confiança para resolver problemas de otimização irrestrita sem o uso de derivadas da função objetivo. Em cada iteração minimizamos um modelo quadrático da função em uma região, em torno do ponto corrente, em que confiamos no modelo. Tal modelo é construído por meio de interpolação polinomial, não usando assim as derivadas da função. Provamos que modelos de interpolação polinomial, sob certas condições, são boas aproximações para a função a otimizar, o que nos permite mostrar a convergência global para o Algoritmo. No final do texto, apresentamos alguns testes computacionais.

Palavras-chave: *Região de Confiança, Otimização Sem Derivadas.*

Abstract

We present a derivative-free trust region method for solving unconstrained optimization problems. In each iteration we minimize a quadratic model function in a region around the current point, in which we trust the model. We utilize polynomial interpolation models. We prove that polynomial interpolation models, under certain conditions, are good approximations for the function to be optimized. Then we can show global convergence to the algorithm and present some computational tests.

Keywords: *Trust-Region, Derivative-Free Optimization.*

Notação

x_k	Vetor do \mathbb{R}^n	4
m_k	Modelo que aproxima uma função na região de confiança	4
g_k	Gradiente do modelo m_k no ponto x_k	4
B_k	Hessiana do modelo m_k no ponto x_k	4
f_k	Valor da função f no ponto x_k	4
$\ \cdot\ $	Norma euclidiana de um ponto do \mathbb{R}^n	4
Δ_k	Raio para a região de confiança	4
κ_h	Limitante para B_k	4
p_k	Passo $x - x_k$	5
ρ_k	Razão entre a redução predita e a redução verdadeira	5
η	Parâmetro para aceitação do passo	5
$\bar{\Delta}$	Maior valor admissível para o raio	6
p_k^c	Passo de Cauchy	7
x_k^c	Ponto de Cauchy	7
c_1	Fator de redução do modelo em relação a redução de x_k^c	10
$N(x_0)$	Conjunto de Nível para a função f	10
ν	Constante de Lipschitz da função f	11
$\lambda_{\min}(B)$	Menor autovalor de uma matriz quadrada B	16
$B(x, \Delta)$	Bola de centro em x e raio Δ	16
p_k^E	Autopasso	17
c_2	Fator de redução do modelo em relação a redução de $x_k + p_k^E$	17
c_0	Fator de redução do modelo em relação à maior entre x_k^c e $x_k + p_k^E$	17
N	Conjunto de amostra de f	18
κ_{ef}	Constante para limitação do erro na função f	19
κ_{eg}	Constante para limitação do erro no gradiente da função f	19
κ_{eh}	Constante para limitação do erro na Hessiana da função f	19
ν_1^m	Limitante para constante de Lipschitz do gradiente do modelo m	19
ν_2^m	Limitante para constante de Lipschitz da Hessiana do modelo m	19
$0 \leq \eta_0 \leq \eta_1 < 1$	Parâmetros para aceitação do passo	21
$0 < \gamma_1 < \gamma_{inc}$	Parâmetros para atualização do raio de região de confiança	21
ε_c	Parâmetro para passo de criticalidade	21
$\mu > \beta > 0$	Parâmetros para atualização do raio na etapa de criticalidade	21
m_k^0	Modelo histórico da região de confiança na iteração k	21
Δ_k^0	Raio histórico da região de confiança na iteração k	21
\mathbb{N}_0	Conjunto Números Naturais incluindo o 0	30
\mathcal{P}_n^d	Polinômios de grau menor ou igual a d no \mathbb{R}^n	45
Λ	Constante de posicionamento	53
ν_2	Constante de Lipschitz para a Hessiana da função f	72
$\ A\ _F$	Norma de Frobenius da matriz A	72
L	Limitante para o algoritmo de melhoramento do modelo	80

Sumário

Introdução	1
1 O Método de Região de Confiança	3
1.1 Convergência	7
2 O Método de Região de Confiança sem Derivadas	14
2.1 Condições para os Modelos	17
2.2 Método de Região de Confiança Sem Derivadas Primeira Ordem	19
2.2.1 Algoritmo - Primeira Ordem	19
2.2.2 Convergência Global - Pontos críticos de primeira ordem	22
2.3 Método de Região de Confiança Sem Derivadas - Segunda Ordem	30
2.3.1 Algoritmo - Segunda Ordem	30
2.3.2 Convergência Global - Pontos críticos de segunda ordem	32
3 Interpolação Polinomial	43
3.1 Interpolação Polinomial	45
3.2 Polinômios de Lagrange	46
3.3 Medidas de Posicionamento	51
3.4 Número de Condição como uma medida de bom posicionamento	57
3.5 Λ -posicionamento e o número de condição	61
3.6 Limitante para o erro em termos do número de condição	64
3.7 Algoritmos para bom posicionamento	72
4 Experimentos Numéricos	85
4.1 Algoritmo Implementado	85
4.2 Resultados Computacionais	87
4.2.1 Bateria de Testes	88
4.2.2 Perfil de desempenho	93
Conclusões	99

Introdução

Neste trabalho concentramos nossa atenção no problema de minimizar uma função f definida em \mathbb{R}^n . Nosso objetivo é estudar, do ponto de vista teórico e computacional, um método de região de confiança que não faça uso de derivadas da função. Entre os diversos algoritmos para solucionar tal problema os mais usuais fazem uso de derivadas. Podemos citar o método do Gradiente ou de Cauchy, que será útil no contexto deste trabalho; o método de Newton, que usa informações de primeira e segunda ordem da função e os métodos Quase-Newton. Maiores informações podem ser encontradas em [10, 14, 19].

Embora métodos com derivadas tenham ótimo desempenho, algumas vezes encontrar a derivada de uma função é muito trabalhoso ou computacionalmente inviável, em outras tal cálculo é impossível uma vez que nem sempre possuímos a expressão analítica de f . Porém, mesmo sem conhecer as derivadas, algumas vezes é desejável realizar a otimização. São nesses casos que precisamos utilizar métodos sem derivadas, mesmo conscientes de que o desempenho prático de tais métodos dificilmente supera o de um bom algoritmo baseado em derivadas, especialmente no que tange ao tempo computacional e número de avaliações de função [15].

Conn, Scheinberg e Vicente em [5] consideram a otimização sem derivadas uma área aberta e desafiadora muito importante na ciência da computação e engenharia, com um enorme potencial prático. A fonte de sua importância é a necessidade cada vez maior em resolver problemas de otimização definidos por funções cujas derivadas estão indisponíveis ou disponíveis a um custo proibitivo. O aumento da complexidade em modelagem matemática, a maior sofisticação da computação científica, e uma abundância de códigos relacionados são algumas das razões pelas quais a otimização sem derivadas é atualmente uma área de grande interesse.

Em [12], Lopes, Diniz-Ehrhardt e Pedroso fizeram uma revisão dos métodos sem derivadas para resolver o problema de otimização irrestrita. Em [7], Diniz-Ehrhardt, Martínez e Pedroso apresentaram um método para problemas de programação não linear com restrições simples.

Alguns exemplos onde a otimização sem derivadas pode ser usada na prática são citados em [5]. Entre eles, a otimização dos parâmetros de algoritmos e análise automática de erros, geometria de circuitos, geometria molecular, entre outros.

Apesar de grande utilidade, a otimização sem derivadas apresenta algumas limitações,

talvez uma das principais seja o fato de que geralmente não é razoável tentar otimizar problemas com muitas variáveis. Além disso, mesmo em problemas relativamente simples e bem condicionados, geralmente não é realista esperar soluções exatas.

Entre os métodos existentes na literatura para resolver o problema considerado, nosso enfoque reside nos métodos de região de confiança [2]. De modo geral, tais métodos utilizam um modelo quadrático da função objetivo, visando estabelecer uma redução deste modelo num conjunto fechado em que confiamos neste, por isso o nome *Região de Confiança*. Em geral o modelo quadrático é construído por aproximação por Taylor, utilizando assim as derivadas. Nosso objetivo neste trabalho é discutir uma abordagem que não utilize derivadas para construir os modelos, através da interpolação polinomial. No entanto, precisamos que nossos modelos tenham comportamento local semelhante ao dos modelos construídos com derivadas, por isso mostramos resultados que garantem que modelos por interpolação polinomial são boas aproximações para a função que queremos otimizar e, com isso, conseguimos mostrar resultados semelhantes aos obtido pelos métodos de região de confiança com derivadas.

Em [5], Conn, Scheinberg e Vicente trazem, como apêndice, uma relação de softwares para otimização sem derivadas, inclusive alguns referentes ao método de região de confiança baseado em interpolação polinomial, entre eles os softwares CONDOR, DFO, WEDGE, UOBYQA e NEWUOA. E ainda o BOOSTERS e o ORBIT que são baseados em interpolação com funções radiais. Neste trabalho implementamos em Matlab o método estudado e apresentamos alguns testes computacionais mas não temos qualquer pretensão em compará-lo numericamente com estes softwares profissionais disponíveis na literatura.

Nosso trabalho está organizado da seguinte forma. No Capítulo 1 apresentamos o método de região de confiança no sentido clássico, com aproximações por Taylor. No Capítulo 2, mostramos as modificações necessárias para o método sem derivadas, o respectivo algoritmo e alguns resultados teóricos referentes à convergência global. No Capítulo 3 construímos modelos baseados em interpolação polinomial e mostramos que estes satisfazem as hipóteses para a convergência global, além de apresentar algoritmos para construir tais modelos. Os Capítulos 2 e 3 são baseados essencialmente nos trabalhos de Conn, Scheinberg e Vicente [3, 4, 5]. Por fim, no Capítulo 4 trazemos testes computacionais para calibrar os parâmetros do algoritmo proposto. Para tanto, usamos a biblioteca de problemas de Moré, Garbow e Hillstom [13]. Nossas conclusões são apresentadas no Capítulo final.

Capítulo 1

O Método de Região de Confiança

O método de região de confiança define um modelo da função objetivo e uma região em torno do ponto corrente na qual confiamos no modelo. Calculamos então um minimizador aproximado do modelo na região de confiança. Caso este ponto forneça uma redução razoável no valor da função objetivo, ele é aceito e repetimos o processo. Caso contrário, pode ser que o modelo não represente adequadamente a função. Neste caso, o ponto é recusado, e reduzimos o tamanho da região para encontrar um novo minimizador.

Neste capítulo vamos estudar o método de região de confiança no sentido clássico, com uso de derivadas, para problemas irrestritos. Apresentaremos um algoritmo e mostraremos resultados de convergência, ainda usando derivadas, conforme pode ser visto em [2, 14, 19].

Seja o problema irrestrito

$$\begin{aligned} & \text{minimizar} && f(x) \\ & && x \in \mathbb{R}^n, \end{aligned}$$

onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ é de classe \mathcal{C}^1 .

Dado um ponto $x_k \in \mathbb{R}^n$, consideramos o modelo quadrático $m_k : \mathbb{R}^n \rightarrow \mathbb{R}$ de f em torno de x_k dado por:

$$m_k(x) = f(x_k) + \nabla f(x_k)^\top (x - x_k) + \frac{1}{2}(x - x_k)^\top B_k(x - x_k), \quad (1.1)$$

onde $B_k \in \mathbb{R}^{n \times n}$ é uma matriz simétrica, podendo ser uma aproximação da hessiana $\nabla^2 f(x_k)$ ou qualquer outra matriz simétrica que satisfaça $\|B_k\| \leq \kappa_h$, para alguma constante $\kappa_h > 0$, independente de k .

O modelo definido acima aproxima bem a função f numa vizinhança de x_k . Podemos portanto considerar uma região de raio $\Delta_k > 0$, em que confiamos no modelo

$$\{x \in \mathbb{R}^n \mid \|x - x_k\| \leq \Delta_k\}. \quad (1.2)$$

Para facilitar a notação, denotaremos a função e o gradiente no ponto corrente como

$f_k = f(x_k)$, $g_k = \nabla f(x_k)$ e $p = x - x_k$. Podemos então reescrever (1.1) como

$$m_k(x_k + p) = f_k + g_k^\top p + \frac{1}{2} p^\top B_k p,$$

e a região (1.2) pode ser reescrita como

$$\{p \in \mathbb{R}^n \mid \|p\| \leq \Delta_k\}.$$

Notemos que $\nabla m_k(x_k)$ coincide com $\nabla f(x_k)$.

O próximo passo é encontrar uma solução aproximada $p_k \in \mathbb{R}^n$ para o subproblema

$$\begin{aligned} \text{minimizar} \quad & m_k(x_k + p) = f_k + g_k^\top p + \frac{1}{2} p^\top B_k p \\ \text{sujeito a} \quad & \|p\| \leq \Delta_k. \end{aligned} \tag{1.3}$$

O ponto $x_k + p_k$ minimiza o modelo quadrático (1.1) na região (1.2). Esperamos, portanto, que este ponto proporcione uma redução na função objetivo que seja no mínimo uma fração da redução do modelo. Para formalizar este conceito definimos a redução **predita**, produzida pelo passo p_k , como $pred = m_k(x_k) - m_k(x_k + p_k)$ e a redução **verdadeira** como $ared = f_k - f(x_k + p_k)$ e, então, para $pred \neq 0$, calculamos a razão

$$\rho_k = \frac{ared}{pred}. \tag{1.4}$$

O passo p_k será aceito quando a razão ρ_k for maior que uma constante $\eta > 0$ dada. Neste caso, definimos $x_{k+1} = x_k + p_k$ e repetimos o processo. Caso contrário, recusamos o passo p_k , reduzimos o raio Δ_k e resolvemos o subproblema (1.3) com o novo raio.

A Figura 1.1 mostra um passo do método de região de confiança. O ponto x_{k+1} é a minimização do modelo na região de confiança centrada em x_k . No gráfico da direita o minimizador irrestrito do modelo está na região de confiança, e assim o raio da região não interfere no progresso do algoritmo como veremos a seguir.

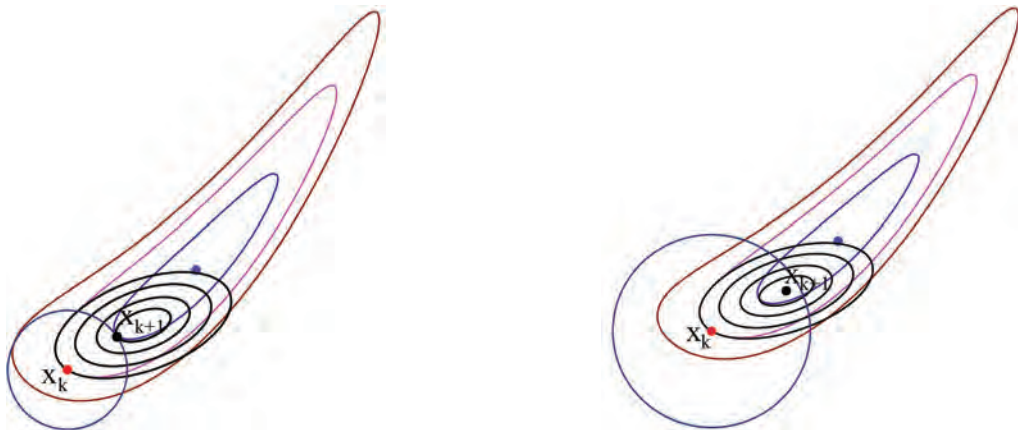


Figura 1.1: Um passo de região de confiança

Vamos agora formalizar a discussão no algoritmo abaixo, que se baseia no proposto em [14].

Algoritmo 1.1. *Região de confiança*

Dados: $x_0 \in \mathbb{R}^n$, $\bar{\Delta} > 0$, $\Delta_0 \in (0, \bar{\Delta})$, $\eta \in [0, \frac{1}{4})$, $\varepsilon > 0$.

$k = 0$.

REPITA enquanto $\|\nabla f(x_k)\| > \varepsilon$

Obtenha p_k resolvendo (1.3) aproximadamente.

Calcule ρ_k por (1.4).

Se $\rho_k < \frac{1}{4}$,

$$\Delta_{k+1} = \frac{1}{4}\Delta_k.$$

Senão

Se $\rho_k > \frac{3}{4}$ e $\|p_k\| = \Delta_k$,

$$\Delta_{k+1} = \min\{2\Delta_k, \bar{\Delta}\}.$$

Senão

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

Se $\rho_k > \eta$,

$$x_{k+1} = x_k + p_k.$$

Senão

$$x_{k+1} = x_k.$$

$k = k + 1$.

FIM

Note que, como o passo p_k é obtido através da minimização do modelo m_k sobre uma região centrada em x_k , a redução predita será sempre não negativa, e não faz sentido quando se anula.

Assim,

- Se ρ_k é negativo, o novo valor $f(x_k + p_k)$ é maior que o anterior f_k e o passo p_k deve ser rejeitado.
- Se ρ_k é positivo mas próximo de zero, digamos $\rho_k \in (0, \frac{1}{4})$, podemos ou não aceitar o passo, uma vez que houve pelos menos um pequeno decréscimo na função objetivo. Neste caso, diminuimos o raio da região de confiança.
- Se ρ_k está próximo de 1, digamos $\rho_k > \frac{3}{4}$, aceitamos o passo, pois a redução verdadeira está próxima da redução do modelo. Se $\|p_k\| = \Delta_k$ podemos aumentar o

raio da região de confiança na próxima iteração. Caso $\|p_k\| < \Delta_k$, não faz sentido aumentar a região de confiança mesmo o modelo sendo uma boa aproximação da função objetivo.

- Se ρ_k é positivo, mas não está próximo de zero e nem de 1, ou seja, $\rho_k \in [\frac{1}{4}, \frac{3}{4}]$, aceitamos o passo e mantemos o raio da região.

No Algoritmo 1.1, $\bar{\Delta}$ é um limitante para o raio da região de confiança, assim, para todo k temos $\Delta_k \leq \bar{\Delta}$. Note que o raio é aumentado somente se p_k está na fronteira da região de confiança. Se o passo fica estritamente dentro da região, podemos inferir que o atual valor de Δ_k não está a interferir no progresso do algoritmo, portanto podemos deixar inalterado o seu valor para a próxima iteração.

O parâmetro $\eta \geq 0$ está relacionado com o critério de aceitação do passo. De modo geral, o passo é aceito quando a redução no valor da função objetivo é pelo menos um percentual η da redução obtida no modelo. Assim, quanto menor o valor do parâmetro η , mais tolerante é o algoritmo. No caso extremo em que $\eta = 0$, qualquer decréscimo no valor da função objetivo é suficiente para aceitar o passo.

Embora a princípio estejamos procurando uma solução ótima de (1.3), é suficiente para a convergência encontrar uma solução aproximada p_k que nos forneça, na região de confiança, uma redução suficiente no modelo. Isto é importante pois muitas vezes não conseguimos resolver o subproblema de forma exata.

Essa redução procurada pode ser quantificada em termos do *ponto de Cauchy*, que é o minimizador de m_k ao longo da direção oposta ao gradiente sujeito à região de confiança, Assim, o *passo de Cauchy* é,

$$p_k^c = -t_k g_k(x),$$

onde $t_k \geq 0$ é solução do problema

$$\begin{aligned} \text{minimizar} \quad & m_k(x_k - t g_k) = f_k - t \|g_k\|^2 + \frac{1}{2} t^2 g_k^\top B_k g_k \\ \text{sujeito a} \quad & \|t g_k\| \leq \Delta_k. \end{aligned} \tag{1.5}$$

O ponto de Cauchy é o ponto obtido a partir de x_k com o passo de Cauchy, ou seja,

$$x_k^c = x_k + p_k^c. \tag{1.6}$$

A Figura 1.2 mostra o ponto de Cauchy numa iteração k . As elipses representam as curvas de nível do modelo m_k . A área hachurada corresponde ao conjunto de pontos na

região de confiança que satisfazem a relação

$$pred \geq m_k(x_k) - m_k(x_k^c).$$

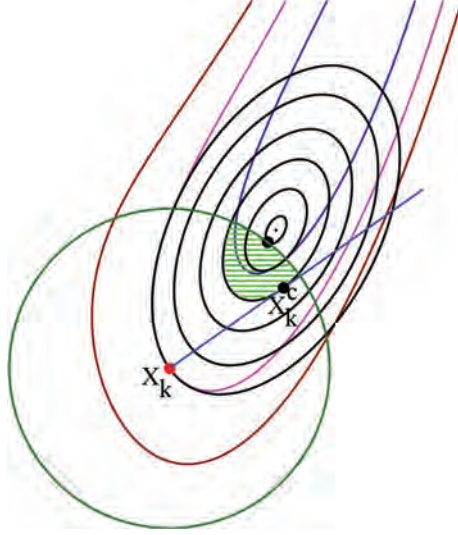


Figura 1.2: Ponto de Cauchy e pontos “melhores”.

Esta condição será a base de uma das hipóteses na análise de convergência, isto é, vamos supor que a solução aproximada do subproblema (1.3) seja pelo menos tão boa quanto a solução de Cauchy.

1.1 Convergência

Nesta seção provaremos que o método de região de confiança é globalmente convergente. Inicialmente, mostramos o lema abaixo que estabelece o decréscimo esperado no modelo com o passo de Cauchy.

Lema 1.1. [19, Lema 5.13] *O ponto de Cauchy, definido em (1.6), satisfaz*

$$m_k(x_k) - m_k(x_k^c) \geq \frac{1}{2} \|g_k\| \min \left\{ \Delta_k, \frac{\|g_k\|}{\|B_k\|} \right\}.$$

Demonstração. Primeiramente, vamos obter t_k , solução do problema (1.5), isto é, o minimizador da função quadrática $\varphi_k : \mathbb{R} \rightarrow \mathbb{R}$ dada por $\varphi_k(t) = m_k(x_k - tg_k)$, ou seja,

$$\varphi_k(t) = f_k - t \|g_k\|^2 + \frac{1}{2} t^2 g_k^\top B_k g_k \quad (1.7)$$

no intervalo $0 \leq t \leq \frac{\Delta_k}{\|g_k\|}$. Para isso, consideramos dois casos: $g_k^\top B_k g_k > 0$ e $g_k^\top B_k g_k \leq 0$.

(i) Quando $g_k^\top B_k g_k > 0$, podemos calcular o parâmetro t^* como sendo o único minimizador de (1.7) sem restrição da região de confiança. Derivando (1.7) com respeito a t e igualando o resultado a zero, obtemos

$$0 = \|g_k\|^2 - t^* g_k^\top B_k g_k,$$

e assim

$$t^* = \frac{\|g_k\|^2}{g_k^\top B_k g_k}. \quad (1.8)$$

Dois subcasos podem ocorrer. O primeiro é quando $t^* \leq \frac{\Delta_k}{\|g_k\|}$. Neste caso temos que $t_k = t^*$, pois o minimizador irrestrito de (1.7) está na região de confiança. E portanto

$$m_k(x_k^c) = m_k(x_k - t_k^* g_k) = f_k - \frac{\|g_k\|^4}{g_k^\top B_k g_k} + \frac{1}{2} \frac{\|g_k\|^4}{g_k^\top B_k g_k}.$$

Como $g_k^\top B_k g_k > 0$, usando a desigualdade de Cauchy-Schwarz

$$m_k(x_k) - m_k(x_k^c) = \frac{\|g_k\|^4}{g_k^\top B_k g_k} - \frac{1}{2} \frac{\|g_k\|^4}{g_k^\top B_k g_k} \geq \frac{\|g_k\|^4}{2\|g_k\|^2 \|B_k\|} = \frac{1}{2} \frac{\|g_k\|^2}{\|B_k\|}. \quad (1.9)$$

No segundo subcaso temos $t^* > \frac{\Delta_k}{\|g_k\|}$, ou seja, o minimizador irrestrito de (1.7) não está na região de confiança o que implica que o minimizador está na fronteira, de onde $t_k \|g_k\| = \Delta_k$.

Usando (1.8) temos

$$t_k = \frac{\Delta_k}{\|g_k\|} < t^* = \frac{\|g_k\|^2}{g_k^\top B_k g_k},$$

o que implica

$$g_k^\top B_k g_k < \frac{\|g_k\|^2}{t_k}.$$

Assim, substituindo em (1.7) e usando que o minimizador está na fronteira

$$\begin{aligned} m_k(x_k) &= f_k - t_k \|g_k\|^2 + \frac{1}{2} t_k^2 g_k^\top B_k g_k \\ &< f_k - \|g_k\| \Delta_k + \frac{1}{2} \|g_k\| \Delta_k \\ &= f_k - \frac{1}{2} \|g_k\| \Delta_k, \end{aligned}$$

de onde segue que

$$m_k(x_k) - m_k(x_k^c) > \frac{1}{2} \|g_k\| \Delta_k. \quad (1.10)$$

(ii) Agora o caso em que $g_k^\top B_k g_k \leq 0$. Neste caso, o ponto de Cauchy está na fronteira da região de confiança, ou seja, $t_k = \frac{\Delta_k}{\|g_k\|}$. Por (1.7), temos que

$$m_k(x_k^c) = f_k - t_k \|g_k\|^2 + \frac{1}{2} t_k^2 g_k^\top B_k g_k \leq f_k - t_k \|g_k\|^2,$$

o que implica

$$m_k(x_k) - m_k(x_k^c) \geq \|g_k\| \Delta_k \geq \frac{1}{2} \|g_k\| \Delta_k. \quad (1.11)$$

Portanto, segue de (1.9), (1.10) e (1.11) que

$$m_k(x_k) - m_k(x_k^c) \geq \frac{1}{2} \|g_k\| \min \left\{ \Delta_k, \frac{\|g_k\|}{\|B_k\|} \right\},$$

o que demonstra o resultado. \square

Para estabelecer a convergência do método de região de confiança, vamos supor que o Algoritmo 1.1 gera uma sequência infinita $\{x_k\}$ em \mathbb{R}^n e que são satisfeitas as seguintes hipóteses:

Hipótese 1. *A função objetivo f é continuamente diferenciável com gradiente Lipschitz.*

Hipótese 2. *A solução aproximada p_k de (1.3) satisfaz*

$$pred = m_k(x_k) - m_k(x_k + p_k) \geq c_1 (m_k(x_k) - m_k(x_k^c)) \geq c_1 \|g_k\| \min \left\{ \Delta_k, \frac{\|g_k\|}{\|B_k\|} \right\},$$

onde $c_1 > 0$ é uma constante.

Hipótese 3. *O passo p_k satisfaz $\|p_k\| \leq \theta_p \Delta_k$, para alguma constante $\theta_p \geq 1$.*

Hipótese 4. *A função f é limitada inferiormente no conjunto de nível*

$$N(x_0) = \{x \mid f(x) \leq f(x_0)\}.$$

Hipótese 5. *As Hessianas B_k são uniformemente limitadas, isto é, existe uma constante $\kappa_h > 0$ tal que $\|B_k\| \leq \kappa_h$ para todo $k \in \mathbb{N}$.*

As Hipóteses 1, 4 e 5 são comuns em análise de convergência. Em vista do Lema 1.1, a Hipótese 2 requer a obtenção de um passo cuja redução no modelo seja proporcional àquela obtida pelo passo de Cauchy. A condição assumida na Hipótese 3 implica que o passo pode exceder a região de confiança, contanto que permaneça dentro de algum múltiplo fixo do raio.

O próximo resultado nos fornece uma estimativa para a razão entre a redução verdadeira e a redução predita do modelo, ou seja, ρ_k , definida em (1.4).

Lema 1.2. [19, Lema 5.14] Suponha que sejam satisfeitas as Hipóteses 1-5. Então existe uma constante $c > 0$ tal que para todo $k \geq 0$

$$|\rho_k - 1| \leq \frac{c\Delta_k^2}{\|g_k\| \min \left\{ \Delta_k, \frac{\|g_k\|}{\kappa_h} \right\}}.$$

Demonstração. Pelo Teorema do Valor Médio, existe $\theta_k \in (0, 1)$ tal que

$$f(x_k + p_k) = f_k + \nabla f(x_k + \theta_k p_k)^\top p_k.$$

Portanto,

$$\begin{aligned} \text{ared} - \text{pred} &= f_k - f(x_k + p_k) - m_k(x_k) + m_k(x_k + p_k) \\ &= m_k(x_k + p_k) - f(x_k + p_k) \\ &= g_k^\top p_k + \frac{1}{2}(p_k)^\top B_k p_k - \nabla f(x_k + \theta_k p_k)^\top p_k \\ &= \frac{1}{2}(p_k)^\top B_k p_k - (\nabla f(x_k + \theta_k p_k) - g_k)^\top p_k. \end{aligned}$$

Usando a Hipótese 5, o fato de que ∇f é Lipschitz e a desigualdade de Cauchy-Schwarz,

$$|\text{ared} - \text{pred}| \leq \frac{\kappa_h}{2} \|p_k\|^2 - \nu \theta_k \|p_k\|^2 \leq \hat{c} \Delta_k^2,$$

com $\hat{c} = \frac{\kappa_h - 2\nu\theta_k}{2} > 0$ e ν a constante de Lipschitz para o gradiente.

Assim, pelas Hipóteses 2 e arefhip5

$$|\rho_k - 1| = \left| \frac{\text{ared} - \text{pred}}{\text{pred}} \right| \leq \frac{\hat{c} \Delta_k^2}{c_1 \|\nabla f_k\| \min \left\{ \Delta_k, \frac{\|\nabla f_k\|}{\kappa_h} \right\}},$$

o que prova o lema para $c = \frac{\hat{c}}{c_1}$. □

O próximo teorema já nos permite concluir algo sobre convergência. Ele garante que se a sequência $\{x_k\}$, gerada pelo algoritmo, for limitada, então ela possui um ponto de acumulação estacionário.

Teorema 1.3. [19, Teorema 5.15] *Suponha que sejam satisfeitas as Hipóteses 1-5. Então*

$$\liminf_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0.$$

Demonstração. Suponha por absurdo que isto seja falso. Então existe $\varepsilon > 0$ tal que $\|\nabla f(x_k)\| \geq \varepsilon$, para todo $k \in \mathbb{N}$. Considere $\tilde{\Delta} = \min \left\{ \frac{\varepsilon}{\kappa_h}, \frac{\varepsilon}{2c} \right\}$, onde κ_h e c são as constantes do Lema 1.2. Se $\Delta_k \leq \tilde{\Delta}$, então

$$\Delta_k \leq \frac{\varepsilon}{\kappa_h} \leq \frac{\|\nabla f_k\|}{\kappa_h} \quad \text{e} \quad \Delta_k \leq \frac{\varepsilon}{2c}.$$

Portanto, pelo Lema 1.2,

$$|\rho_k - 1| \leq \frac{c\Delta_k}{\varepsilon} \leq \frac{1}{2}.$$

Assim, $\rho_k \geq \frac{1}{2} > \frac{1}{4}$ e pelo Algoritmo 1.1 temos $\Delta_{k+1} \geq \Delta_k$. Isto significa que o raio é reduzido somente se $\Delta_k > \tilde{\Delta}$, caso em que $\Delta_{k+1} = \frac{\Delta_k}{2} > \frac{\tilde{\Delta}}{2}$. Podemos então concluir que

$$\Delta_k \geq \min \left\{ \Delta_0, \frac{\tilde{\Delta}}{2} \right\}, \quad (1.12)$$

para todo $k \in \mathbb{N}$. Seja então o conjunto

$$\mathcal{K} = \left\{ k \in \mathbb{N} \mid \rho_k \geq \frac{1}{4} \right\}.$$

Dado $k \in \mathcal{K}$, pelo mecanismo do Algoritmo 1.1 e pela Hipótese 2 temos

$$\begin{aligned} f(x_k) - f(x_{k+1}) &= f(x_k) - f(x_k + p_k) \\ &= \rho_k (m_k(x_k) - m_k(x_k + p_k)) \\ &\geq \frac{1}{4} (m_k(x_k) - m_k(x_k + p_k)) \\ &\geq \frac{1}{4} c_1 \varepsilon \min \left\{ \Delta_k, \frac{\varepsilon}{\kappa_h} \right\}. \end{aligned}$$

Usando isto e (1.12), temos que existe uma constante $\delta > 0$ tal que

$$f(x_k) - f(x_{k+1}) \geq \delta, \quad (1.13)$$

para todo $k \in \mathcal{K}$. Por outro lado, a sequência $\{f_k\}$ é não crescente e, pela Hipótese 4, limitada inferiormente, de onde segue que $f(x_k) - f(x_{k+1}) \rightarrow 0$. Portanto, de (1.13), podemos concluir que o conjunto \mathcal{K} é finito. Assim $\rho_k < \frac{1}{4}$, para todo $k \in \mathbb{N}$ suficientemente grande e então Δ_k será reduzido à metade em cada iteração. Isto implica que $\Delta_k \rightarrow 0$, o que contradiz (1.12). Deste modo, a afirmação do teorema é verdadeira. \square

Vamos agora provar a convergência global do método de região de confiança.

Teorema 1.4. [19, Teorema 5.16] *Suponha que sejam satisfeitas as Hipóteses 1-5. Então*

$$\lim_{k \rightarrow \infty} \nabla f(x_k) = 0.$$

Demonstração. Suponha por absurdo que para algum $\varepsilon > 0$ o conjunto

$$\mathcal{K} = \{k \in \mathbb{N} \mid \|\nabla f(x_k)\| \geq \varepsilon\}$$

seja infinito. Dado $k \in \mathcal{K}$, considere o primeiro índice $l_k > k$ tal que $\|\nabla f(x_{l_k})\| \leq \frac{\varepsilon}{2}$. A existência de l_k é assegurada pelo Teorema 1.3. Como ∇f é Lipschitz, temos

$$\frac{\varepsilon}{2} \leq \|\nabla f(x_k) - \nabla f(x_{l_k})\| \leq \nu \|x_k - x_{l_k}\|,$$

para alguma constante $\nu > 0$. Portanto,

$$\frac{\varepsilon}{2\nu} \leq \|x_k - x_{l_k}\| \leq \sum_{j \in \mathcal{S}_k} \|x_j - x_{j+1}\| \leq \sum_{j \in \mathcal{S}_k} \theta_p \Delta_j, \quad (1.14)$$

onde $\mathcal{S}_k = \{j \in \mathbb{N} \mid k \leq j < l_k \text{ e } x_{j+1} \neq x_j\}$. Pelo mecanismo do Algoritmo 1.1, pela Hipótese 2 e definição de l_k , temos

$$\begin{aligned} f(x_k) - f(x_{l_k}) &= \sum_{j \in \mathcal{S}_k} (f(x_j) - f(x_{j+1})) \\ &\geq \sum_{j \in \mathcal{S}_k} \eta (m_j(x_j) - m_j(x_{j+1})) \\ &\geq \sum_{j \in \mathcal{S}_k} \eta c_1 \varepsilon \min \left\{ \Delta_j, \frac{\varepsilon}{\kappa_h} \right\}. \end{aligned}$$

Definindo $\delta = \min \left\{ \frac{\eta c_1 \varepsilon^2}{2\gamma\nu}, \frac{\eta c_1 \varepsilon^2}{\kappa_h} \right\}$ e usando (1.14), obtemos

$$f(x_k) - f(x_{l_k}) \geq \delta, \quad (1.15)$$

para todo $k \in \mathcal{K}$. Por outro lado, a sequência $\{f_k\}$ é não crescente e, pela Hipótese 4, limitada inferiormente, de onde segue que $f(x_k) - f(x_{k+1}) \rightarrow 0$, contradizendo (1.15). Deste modo, a afirmação do teorema é verdadeira. \square

Uma consequência do Teorema 1.4 é que todo ponto de acumulação \bar{x} de uma sequência gerada pelo Algoritmo 1.1 é estacionário, ou seja, $\nabla f(\bar{x}) = 0$. De fato, se \bar{x} é valor de aderência da sequência $\{x_k\}$ gerada pelo Algoritmo 1.1, existe um subconjunto $\mathbb{N}' \subset \mathbb{N}$

tal que $x_k \xrightarrow{\mathbb{N}'} \bar{x}$. Como f é continuamente diferenciável, temos

$$\nabla f(x_k) \xrightarrow{\mathbb{N}'} \nabla f(\bar{x}),$$

mas do Teorema 1.4, temos que $\nabla f(x_k) \rightarrow 0$, portanto $\nabla f(x_k) \xrightarrow{\mathbb{N}'} 0$ e, pela unicidade do limite, $\nabla f(\bar{x}) = 0$.

Capítulo 2

O Método de Região de Confiança sem Derivadas

Os métodos de região de confiança para otimização irrestrita sem derivadas usam modelos que se baseiam apenas no valor da função objetivo em alguns pontos. Os modelos correspondentes podem ser construídos por meio de interpolação polinomial, regressão, ou ainda por outra técnica de aproximação. No próximo capítulo discutiremos como construir tais modelos. Neste capítulo vamos estudar alguns critérios que devem ser satisfeitos pelos modelos sem derivadas de modo que tenham um comportamento local semelhante ao que é observado pelos modelos de Taylor e critérios que devem ser obedecidos para garantir a convergência do método. Estas condições foram apresentadas por Conn, Scheinberg e Vicente em [4] e [5, Cap. 10].

No caso dos modelos baseados em uma aproximação por Taylor, o modelo torna-se progressivamente melhor quando a região de confiança torna-se menor. Em modelos oriundos de interpolação ou regressão polinomial isso não é necessariamente verdade. Por isso, precisamos garantir que o raio de região de confiança só seja reduzido quando estivermos certos de que o fracasso de uma etapa atual decorre do tamanho da região de confiança e não da má qualidade do modelo. Com esta garantia, podemos provar, mais uma vez, que o modelo é uma boa aproximação da função objetivo e podemos obter, sob certas hipóteses, resultados semelhantes aos obtidos no caso com derivadas, discutidos no capítulo anterior.

Analogamente ao capítulo anterior, tipicamente o modelo de f em torno de x_k é quadrático e pode ser escrito da seguinte forma

$$m_k(x_k + p) = m_k(x_k) + g_k^\top p + \frac{1}{2} p^\top B_k p.$$

As derivadas do modelo com respeito à variável p são dadas por $\nabla m_k(x_k + p) = B_k p + g_k$, $\nabla m_k(x_k) = g_k$ e $\nabla^2 m_k(x_k) = B_k$. Se m_k é um modelo de Taylor de primeira ordem, então $m_k(x_k) = f(x_k)$ e $g_k = \nabla f(x_k)$ e se o modelo de Taylor é de segunda ordem também temos que $B_k = \nabla^2 f(x_k)$. Em geral, no caso com derivadas, B_k pode ser uma aproximação simétrica da Hessiana da função objetivo. Nos casos que não são baseados nas derivadas da função objetivo usamos modelos onde $B_k \neq \nabla^2 f(x_k)$, $g_k \neq \nabla f(x_k)$ e na ausência de interpolação $m_k(x_k) \neq f(x_k)$.

Em cada iteração k desejamos minimizar o modelo em uma vizinhança de x_k , ou seja, em uma região de confiança de raio Δ_k

$$B(x_k, \Delta_k) = \{x_k \in \mathbb{R}^n \mid \|x - x_k\| \leq \Delta_k\}.$$

Esta minimização pode ser aproximada. Para discutir o conceito de solução aproximada, resgatamos a discussão do capítulo anterior sobre o Ponto de Cauchy, que é o ponto que minimiza o modelo ao longo da direção oposta ao gradiente restrito à região de confiança. Sabemos pelo Lema 1.1 que o *ponto de Cauchy* satisfaz

$$m_k(x_k) - m_k(x_k^c) \geq \frac{1}{2} \|g_k\| \min \left\{ \Delta_k, \frac{\|g_k\|}{\|B_k\|} \right\}.$$

Para o propósito de convergência a pontos críticos de primeira ordem, precisamos que a redução do modelo seja pelo menos uma fração $c_1 \in (0, 1)$ da redução do modelo obtida pelo ponto de Cauchy. Em vista do Lema 1.1, a solução aproximada p_k deve satisfazer

$$m_k(x_k) - m_k(x_k + p_k) \geq c_1 (m_k(x_k) - m_k(x_k^c)) \geq c \|g_k\| \min \left\{ \Delta_k, \frac{\|g_k\|}{\|B_k\|} \right\}, \quad (2.1)$$

onde $c = \frac{c_1}{2}$.

Quando consideramos um modelo quadrático e convergência para pontos críticos de segunda ordem, a redução do modelo necessária pode ser alcançada ao longo de uma direção relacionada à maior curvatura negativa, que por sua vez está relacionada com o autovalor negativo de maior módulo da Hessiana do modelo. É nesse sentido que vamos definir o *autopasso*.

Definição 2.1. *Suponha que B_k possui pelo menos um autovalor negativo, e seja $\tau_k = \lambda_{\min}(B_k) =$ o menor autovalor de B_k . Então podemos determinar um passo de curvatura negativa p_k^E , tal que*

$$(p_k^E)^\top (g_k) \leq 0, \quad \|p_k^E\| = \Delta_k \quad e \quad (p_k^E)^\top B_k (p_k^E) = \tau_k \Delta_k^2.$$

Dizemos assim que p_k^E é o autopasso.

Devido à presença da curvatura negativa, p_k^E é o minimizador do modelo quadrático ao longo dessa direção restrito à região de confiança. O autopasso induz a redução no modelo, dada pelo lema abaixo.

Lema 2.2. [5, Lema 10.2] *Suponha que a Hessiana B_k do modelo possui autovalores negativos e que p_k^E e τ_k são como na Definição 2.1. Então temos que*

$$m_k(x_k) - m_k(x_k + p_k^E) \geq -\frac{1}{2}\tau_k\Delta_k^2. \quad (2.2)$$

Demonstração. Pela definição do modelo m_k e do autopasso p_k^E ,

$$\begin{aligned} m_k(x_k) - m_k(x_k + p_k^E) &= -g_k^\top p_k^E - \frac{1}{2}(p_k^E)^\top B_k(p_k^E) \\ &\geq -\frac{1}{2}(p_k^E)^\top B_k(p_k^E) \\ &= -\frac{1}{2}\tau_k\Delta_k^2. \end{aligned}$$

□

Assim, na presença de curvaturas negativas no modelo, o autopasso desempenha um papel similar ao passo de Cauchy. Para a convergência global de segunda ordem precisamos que o decréscimo, em cada passo, seja pelo menos uma fração $c_2 \in (0, 1]$ do decréscimo atingido pelo autopasso, ou seja,

$$m_k(x_k) - m_k(x_k + p_k) \geq c_2[m_k(x_k) - m_k(x_k + p_k^E)].$$

Também esperamos que o passo obtenha uma fração do decréscimo do passo de Cauchy. Assim, assumimos a hipótese de na iteração k ,

$$m_k(x_k) - m_k(x_k + p_k) \geq c_0[m_k(x_k) - \min\{m_k(x_k + p_c^k), m_k(x_k + p_k^E)\}] \quad (2.3)$$

para alguma constante $c_0 \in (0, 1]$.

Um passo que satisfaz esta hipótese é dado pelo cálculo do passo de Cauchy e, na presença de curvatura negativa no modelo, o autopasso, e escolhemos aquele que oferece a maior redução no modelo. Combinando (2.1), (2.2) e (2.3), exigimos que

$$m_k(x_k) - m_k(x_k + p_k) \geq \frac{c_0}{2} \max\left\{\|g_k\| \min\left\{\frac{\|g_k\|}{\|B_k\|}, \Delta_k\right\}, -\tau_k\Delta_k^2\right\}. \quad (2.4)$$

2.1 Condições para os Modelos

Para provarmos a convergência de primeira ordem assumimos que f e seu gradiente são Lipschitz no domínio considerado pelo algoritmo. Para melhor definir essa região, vamos supor que o ponto inicial x_0 é dado e os novos iterados fornecem reduções no valor da função objetivo. Assim, os novos iterados pertencem ao conjunto de nível

$$N(x_0) = \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\}.$$

Entretanto, como consideramos modelos baseados em amostras, é possível que a função objetivo f seja avaliada fora de $N(x_0)$. Então assumimos que a amostra é restrita em regiões da forma $B(x_k; \Delta_k)$ e que Δ_k nunca excede uma constante positiva dada $\bar{\Delta}$. Assim a região onde fazemos a amostra de f é o conjunto

$$N = N(x_0) \cup \bigcup_{x \in N(x_0)} B(x; \bar{\Delta}) = \bigcup_{x \in N(x_0)} B(x; \bar{\Delta}).$$

Hipótese 6. *Dados $x_0 \in \mathbb{R}^n$ e $\bar{\Delta} > 0$, assumamos que f é continuamente diferenciável com gradiente Lipschitz em um domínio aberto $\Omega \subset \mathbb{R}^n$ contendo N .*

Para garantirmos que o modelo torna-se melhor quando diminuimos a região de confiança definimos classe de modelos plenamente lineares e classe de modelos plenamente quadráticos, os quais usaremos para a convergência de primeira ordem e para a convergência de segunda ordem, respectivamente. A construção de modelos que satisfaçam essas definições serão vistas no próximo capítulo.

Definição 2.3 (Classe de Modelos Plenamente Lineares). *[5, Definição 10.3] Considere uma função $f : \mathbb{R}^n \rightarrow \mathbb{R}$ que satisfaz a Hipótese 6. Um conjunto $\mathcal{M} = \{m : \mathbb{R}^n \rightarrow \mathbb{R}, m \in C^1\}$ de modelos da função é chamado uma Classe de Modelos Plenamente Lineares se valem as seguintes condições:*

1. *Existem constantes positivas κ_{ef} , κ_{eg} e ν_1^m tais que para qualquer $x \in N(x_0)$ e $\Delta \in (0, \bar{\Delta}]$ existe um modelo $m \in \mathcal{M}$ da função f , cujo gradiente é Lipschitz com a correspondente constante limitada por ν_1^m , tal que:*

- *o erro entre o gradiente do modelo e o gradiente da função satisfaz*

$$\|\nabla f(x+p) - \nabla m(x+p)\| \leq \kappa_{eg}\Delta, \quad \forall p \in B(0; \Delta),$$

- *o erro entre o modelo e a função satisfaz*

$$|f(x+p) - m(x+p)| \leq \kappa_{ef}\Delta^2, \quad \forall p \in B(0; \Delta).$$

Tal modelo m é chamado *Plenamente Linear* em $B(x; \Delta)$.

2. Para a classe \mathcal{M} existe um algoritmo, que chamamos de Algoritmo de Melhoria do Modelo, que em um número finito e uniformemente limitado (com respeito a x e Δ) de passos pode:

- estabelecer que um modelo $m \in \mathcal{M}$ é plenamente linear em $B(x; \Delta)$.
- ou encontrar um modelo $\bar{m} \in \mathcal{M}$ que é plenamente linear em $B(x; \Delta)$.

Para analisar a convergência para pontos críticos de segunda ordem, nós precisamos que a Hessiana da função objetivo f seja Lipschitz, e esse requerimento de suavidade é dado pela hipótese abaixo.

Hipótese 7. Dados $x_0 \in \mathbb{R}^n$ e $\bar{\Delta} > 0$, assumamos que f é duas vezes continuamente diferenciável com Hessiana Lipschitz em um domínio aberto $\Omega \subset \mathbb{R}^n$ contendo N .

O algoritmo de segunda ordem requer Classe de Modelos Plenamente Quadráticos, que definimos agora.

Definição 2.4 (Classe de Modelos Plenamente Quadráticos). [5, Definição 10.4] Considere uma função $f : \mathbb{R}^n \rightarrow \mathbb{R}$ que satisfaz a Hipótese 7. Um conjunto $\mathcal{M} = \{m : \mathbb{R}^n \rightarrow \mathbb{R}, m \in C^2\}$ de modelos da função f é chamado uma Classe de Modelos Plenamente Quadráticos se valem as seguintes condições:

1. Existem constantes positivas $\kappa_{ef}, \kappa_{eg}, \kappa_{eh}$ e ν_2^m tais que para qualquer $x \in N(x_0)$ e $\Delta \in (0, \bar{\Delta}]$ existe um modelo $m \in \mathcal{M}$ da função f , cuja Hessiana é Lipschitz com a correspondente constante limitada por ν_2^m , tal que:

- o erro entre a Hessiana do modelo e a Hessiana da função satisfaz

$$\|\nabla^2 f(x+p) - \nabla^2 m(x+p)\| \leq \kappa_{eh} \Delta, \quad \forall p \in B(0; \Delta),$$

- o erro entre o gradiente do modelo e o gradiente da função satisfaz

$$\|\nabla f(x+p) - \nabla m(x+p)\| \leq \kappa_{eg} \Delta^2, \quad \forall p \in B(0; \Delta),$$

- o erro entre o modelo e a função satisfaz

$$|f(x+p) - m(x+p)| \leq \kappa_{ef} \Delta^3, \quad \forall p \in B(0; \Delta).$$

Tal modelo m é chamado *Plenamente Quadrático* em $B(x; \Delta)$.

2. Para a classe \mathcal{M} existe um algoritmo, que chamamos de Algoritmo de Melhoria do Modelo, com o qual em um número finito e uniformemente limitado (com respeito a x e Δ) de passos podemos

- estabelecer que um modelo $m \in \mathcal{M}$ é plenamente quadrático em $B(x; \Delta)$.
- ou encontrar um modelo $\bar{m} \in \mathcal{M}$ que é plenamente quadrático em $B(x; \Delta)$.

2.2 Método de Região de Confiança Sem Derivadas Primeira Ordem

Nesta seção apresentamos o algoritmo de região de confiança sem derivadas e a prova de convergência global para pontos críticos de primeira ordem. Tanto o algoritmo quanto os resultados de convergência foram apresentados por Conn, Scheinberg e Vicente em [4] e [5, Cap. 10].

2.2.1 Algoritmo - Primeira Ordem

Agora vamos formalmente apresentar o algoritmo de primeira ordem.

Algoritmo 2.1. [5, Algoritmo 10.1] *Região de Confiança sem Derivadas - 1ª Ordem*

Dados:

- Uma classe de modelos plenamente lineares e o correspondente algoritmo de melhoramento.
- $x_0 \in \mathbb{R}^n$, $\bar{\Delta} > 0$, $\Delta_0^0 \in (0, \bar{\Delta}]$.
- $m_0^0(x_0 + p)$, com gradiente e possivelmente Hessiana em $p = 0$ dados por g_0^0 e H_0^0 .
- $0 \leq \eta_0 \leq \eta_1 < 1$, com $\eta_1 \neq 0$, $0 < \gamma < 1 < \gamma_{inc}$, $\varepsilon_c > 0$, $\mu > \beta > 0$ e $\omega \in (0, 1)$.

$k = 0$.

PASSO 1 - CRITICALIDADE

$$\begin{aligned} \text{SE } \|g_k^0\| &> \varepsilon_c, \\ m_k &= m_k^0. \\ \Delta_k &= \Delta_k^0. \end{aligned}$$

SENÃO

Chame o algoritmo de melhoramento do modelo,

SE m_k^0 não é certificadamente plenamente linear em $B(x_k, \Delta_k^0)$ OU $\Delta_k^0 > \mu \|g_k^0\|$,

Aplique o Algoritmo 2.2 para encontrar um modelo \tilde{m}_k plenamente linear com gradiente e possivelmente Hessiana em $p = 0$ dados por \tilde{g}_k e \tilde{B}_k .

$$\begin{aligned} m_k &= \tilde{m}_k, \\ \Delta_k &= \min\{\max\{\tilde{\Delta}_k, \beta \|\tilde{g}_k\|\}, \Delta_k^0\}. \end{aligned}$$

SENÃO

$$m_k = m_k^0.$$
$$\Delta_k = \Delta_k^0.$$

PASSO 2 - CÁLCULO DO PASSO

Calcule p_k que satisfaça (2.4) de tal modo que $x_k + p_k \in B(x_k, \Delta_k)$.

PASSO 3 - ACEITAÇÃO DO PASSO

Calcule $\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(x_k) - m_k(x_k + p_k)}$.

SE $\rho_k \geq \eta_1$ OU ($\rho_k \geq \eta_0$ E m_k é plenamente linear),

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

Inclua x_{k+1} no conjunto de amostra e obtenha um novo modelo m_{k+1}^0 .

SENÃO

$$x_{k+1} = x_k.$$

$$m_{k+1}^0 = m_k.$$

PASSO 4 - MELHORAMENTO DO MODELO

SE $\rho_k < \eta_1$,

Use o algoritmo de melhoramento do modelo.

Defina m_{k+1}^0 como m_k (possivelmente melhorado).

PASSO 5 - ATUALIZAÇÃO DO RAI0

SE $\rho_k \geq \eta_1$,

Escolha $\Delta_{k+1}^0 \in [\Delta_k, \min\{\gamma_{inc}\Delta_k, \bar{\Delta}\}]$.

SENÃO

SE m_k é plenamente linear,

$$\Delta_{k+1}^0 = \gamma\Delta_k.$$

SENÃO

$$\Delta_{k+1}^0 = \Delta_k.$$

$k = k + 1$.

O algoritmo contempla a aceitação da nova iteração com base em simples diminuição no valor da função objetivo quando escolhermos o parâmetro $\eta_0 = 0$. Destacamos ainda que o modelo m_k e o raio da região de confiança Δ_k são definidos apenas no final do passo de criticalidade. Uma iteração termina por definir um modelo m_{k+1}^0 e um de raio região de confiança Δ_{k+1}^0 históricos para a próxima iteração, que então podem ser alterados ou não pelo passo de criticalidade.

O procedimento invocado na etapa de criticalidade é descrito no próximo algoritmo.

Algoritmo 2.2. [5, Algoritmo 10.2] *Etapa de Criticalidade - 1ª Ordem*

$i = 0.$

$$m_k^{(0)} = m_k^0.$$

REPITA

$$i = i + 1.$$

Melhore o modelo até $m_k^{(i-1)}$ ser plenamente linear em $B(x_k, \omega^{i-1} \Delta_k^0)$.

Denote o novo modelo como $m_k^{(i)}$.

$$\tilde{\Delta}_k = \omega^{i-1} \Delta_k^0.$$

$$\tilde{m}_k = m_k^{(i)}.$$

ATÉ $\tilde{\Delta}_k \leq \mu \|g_k^{(i)}\|.$

Após o passo 3 do Algoritmo 2.1, podemos ter as seguintes possibilidades:

Se $\rho_k \geq \eta_1$, então o novo iterando é aceito e o raio da região de confiança é aumentado ou mantido. Essa é uma iteração de **sucesso**.

Se $\eta_0 \leq \rho_k < \eta_1$ e m_k é plenamente linear, então a nova iteração é aceita e diminuímos o raio da região de confiança. Essa é uma iteração **aceitável**.

Se $\rho_k < \eta_1$ e m_k não é certificado plenamente linear, então o modelo é melhorado e o ponto pode ser incluído no conjunto de amostra para calcular o modelo, mas a iteração não é aceita. Essa é uma iteração de **melhora do modelo**.

Se $\rho_k < \eta_0$ e m_k é plenamente linear, então não obtivemos decréscimo suficiente e o modelo não precisa ser melhorado. Diminuímos o raio da região de confiança e não aceitamos o passo. Essa é uma iteração de **insucesso**.

Essa discussão é resumizada na Figura 2.1.

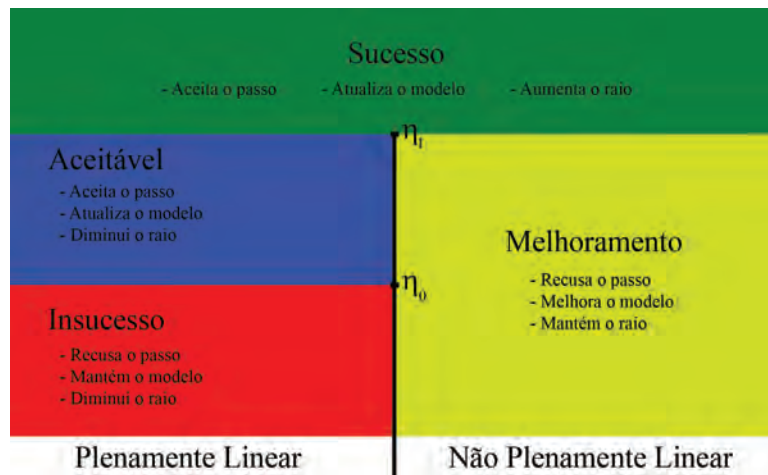


Figura 2.1: Situações possíveis em uma iteração do Algoritmo 2.1

2.2.2 Convergência Global - Pontos críticos de primeira ordem

Vamos mostrar no próximo lema que se o iterando atual é um ponto estacionário de primeira ordem, então o algoritmo não continua infinitamente na etapa de criticalidade do Algoritmo 2.1, ou seja, o Algoritmo 2.2 termina em um número finito de passos.

Lema 2.5. [5, Lema 10.5] *Se $\nabla f(x_k) \neq 0$, o Algoritmo 2.2 termina em um número finito de passos.*

Demonstração. Vamos supor por contradição que o Algoritmo 2.2 é infinito e mostrar que $\nabla f(x_k) = 0$. Para começar, sabemos que m_k^0 não é certificadamente plenamente linear ou $\Delta_k^0 > \mu \|g_k^0\|$. Então definimos $m_k^{(0)} = m_k^0$ e o modelo é melhorado até ser plenamente linear na bola $B(x_k, \omega^0 \Delta_k^0)$, em um número finito de passos de melhoramento. Se o gradiente $g_k^{(1)}$ do modelo resultante $m_k^{(1)}$ satisfaz $\mu \|g_k^{(1)}\| \geq \omega^0 \Delta_k^0$, o processo para com

$$\tilde{\Delta}_k^0 = \omega^0 \Delta_k^0 \leq \mu \|g_k^{(1)}\|.$$

Por outro lado, se $\mu \|g_k^{(1)}\| < \omega^0 \Delta_k^0$, o modelo é melhorado até ser plenamente linear na bola $B(x_k, \omega^0 \Delta_k^0)$. Então, novamente, ou o procedimento para ou o raio é multiplicado outra vez por ω , e assim por diante.

A única maneira para o processo ser infinito, e requerer um número infinito de passos de melhoramento é se

$$\mu \|g_k^{(i)}\| < \omega^{i-1} \Delta_k^0,$$

para todo $i \geq 1$, onde $g_k^{(i)}$ é o gradiente do modelo $m_k^{(i)}$. Esse argumento mostra que

$$\lim_{i \rightarrow +\infty} \|g_k^{(i)}\| = 0.$$

Como temos que o modelo $m_k^{(i)}$ é plenamente linear em $B(x_k, \omega^{i-1} \Delta_k^0)$, com $s = 0$ e $x = x_k$ temos que

$$\|\nabla f(x_k) - g_k^{(i)}\| \leq \kappa_{eg} \omega^{i-1} \Delta_k^0,$$

para todo $i \geq 1$. Pela desigualdade triangular, temos, para todo $i \geq 1$, que

$$\|\nabla f(x_k)\| \leq \|\nabla f(x_k) - g_k^{(i)}\| + \|g_k^{(i)}\| \leq \left(\kappa_{eg} + \frac{1}{\mu} \right) \omega^{i-1} \Delta_k^0.$$

Como $\omega \in (0, 1)$, isso implica que $\nabla f(x_k) = 0$. □

Agora provaremos os resultados relacionados à convergência global para pontos críticos

de primeira ordem. Além de supormos a Hipótese 6 e a existência de modelos plenamente lineares, precisamos de hipóteses adicionais.

Hipótese 8. *Assuma que f é limitada inferiormente em $N(x_0)$, isto é, existe uma constante κ_* tal que, para todo $x \in N(x_0)$, $f(x) \geq \kappa_*$.*

Precisaremos também da hipótese que requer que a Hessiana do modelo B_k seja uniformemente limitada. Em geral, modelos plenamente lineares requerem apenas derivadas contínuas de primeira ordem.

Hipótese 9. *Existe constante $\kappa_h > 0$ tal que, para todo x_k gerado pelo algoritmo,*

$$\|B_k\| \leq \kappa_h.$$

Começaremos a análise com o seguinte lema.

Lema 2.6. *[5, Lema 10.6] Se m_k é plenamente linear em $B(x_k, \Delta_k)$ e*

$$\Delta_k \leq \min \left\{ \frac{\|g_k\|}{\kappa_h}, \frac{c_1 \|g_k\| (1 - \eta_1)}{4\kappa_{ef}} \right\},$$

então a k -ésima iteração é de sucesso.

Demonstração. Como

$$\Delta_k \leq \frac{\|g_k\|}{\kappa_h},$$

pela condição de decréscimo de Cauchy temos que

$$\begin{aligned} m_k(x_k) - m_k(x_k + p_k) &\geq \frac{c_1}{2} \|g_k\| \min \left\{ \frac{\|g_k\|}{\kappa_h}, \Delta_k \right\} \\ &= \frac{c_1}{2} \|g_k\| \Delta_k. \end{aligned}$$

Logo, pelo fato de que m_k é plenamente linear, temos

$$\begin{aligned} |\rho_k - 1| &= \left| \frac{f(x_k) - f(x_k + p_k) - m_k(x_k) + m_k(x_k + p_k)}{m_k(x_k) - m_k(x_k + p_k)} \right| \\ &\leq \left| \frac{f(x_k + p_k) - m_k(x_k + p_k)}{m_k(x_k) - m_k(x_k + p_k)} \right| + \left| \frac{f(x_k) - m_k(x_k)}{m_k(x_k) - m_k(x_k + p_k)} \right| \\ &\leq \frac{\kappa_{ef} \Delta_k^2}{\frac{c_1}{2} \|g_k\| \Delta_k} + \frac{\kappa_{ef} \Delta_k^2}{\frac{c_1}{2} \|g_k\| \Delta_k} \\ &= \frac{4\kappa_{ef} \Delta_k}{c_1 \|g_k\|} \end{aligned}$$

Temos por hipótese que

$$\Delta_k \leq \frac{c_1 \|g_k\| (1 - \eta_1)}{4\kappa_{ef}},$$

assim

$$|\rho_k - 1| \leq \frac{4\kappa_{ef} \Delta_k}{c_1 \|g_k\|} \leq (1 - \eta_1),$$

o que implica $-1 + \eta_1 \leq \rho_k - 1$, e portanto $\eta_1 \leq \rho_k$. O que nos mostra que a k -ésima iteração é de sucesso. \square

O próximo resultado mostra que se a norma do gradiente do modelo é limitada, então o raio da região de confiança também é limitado.

Lema 2.7. [5, Lema 10.7] *Suponha que existe uma constante $\kappa_1 > 0$, tal que $\|g_k\| \geq \kappa_1$ para todo k . Então existe uma constante $\kappa_2 > 0$ tal que*

$$\Delta_k \geq \kappa_2$$

para todo k .

Demonstração. Temos do passo 1 do Algoritmo 2.1 que

$$\Delta_k \geq \min \{ \beta \|g_k\|, \Delta_k^0 \}.$$

Logo por hipótese,

$$\Delta_k \geq \min \{ \beta \kappa_1, \Delta_k^0 \}.$$

Pelo Lema 2.6 sempre que Δ_k é limitado pela constante

$$\bar{\kappa}_2 = \min \left\{ \frac{\kappa_1}{\kappa_h}, \frac{c_1 \kappa_1 (1 - \eta_1)}{4\kappa_{ef}} \right\},$$

a iteração é de sucesso (se m_k é plenamente linear) ou de melhora do modelo (se m_k não é plenamente linear), então pelo mecanismo de atualização do raio temos que $\Delta_{k+1}^0 \geq \Delta_k$.

Por outro lado se Δ_k não é limitado por $\bar{\kappa}_2$ e diminuimos o raio, o novo raio fica limitado por $\gamma \bar{\kappa}_2$. O que nos garante que

$$\Delta_k \geq \min \{ \beta \kappa_1, \Delta_0^0, \gamma \bar{\kappa}_2 \} = \kappa_2,$$

e mostra o resultado. \square

No próximo lema vamos considerar o que acontece quando o número de iterações de sucesso é finito.

Lema 2.8. [5, Lema 10.8] *Se o número de iterações de sucesso é finito, então*

$$\lim_{k \rightarrow +\infty} \|\nabla f(x_k)\| = 0.$$

Demonstração. Vamos considerar as iterações que vêm depois da última iteração de sucesso. Sabemos que só podemos ter um número finito (uniformemente limitado, digamos, por L) de iterações de melhoramento do modelo antes do modelo se tornar plenamente linear, e portanto, há um número infinito de iterações que são aceitáveis ou de insucesso e em ambos os casos, o raio da região de confiança é reduzido. Como não existem mais iterações de sucesso, nunca aumentamos Δ_k para k suficientemente grande. Além disso, Δ_k é reduzido, no mínimo a cada L iterações, por um fator γ . Assim, Δ_k converge para zero.

Agora para cada j , seja i_j o índice da primeira iteração depois da j -ésima iteração onde o modelo m_j é plenamente linear. Então,

$$\|x_j - x_{i_j}\| \leq L\Delta_j \rightarrow 0 \quad \text{quando} \quad j \rightarrow +\infty.$$

Temos que

$$\begin{aligned} \|\nabla f(x_j)\| &= \|\nabla f(x_j) - \nabla f(x_{i_j}) + \nabla f(x_{i_j}) - g_{i_j} + g_{i_j}\| \\ &\leq \|\nabla f(x_j) - \nabla f(x_{i_j})\| + \|\nabla f(x_{i_j}) - g_{i_j}\| + \|g_{i_j}\| \\ &\leq \nu\|x_j - x_{i_j}\| + \kappa_{eg}\Delta_{i_j} + \|g_{i_j}\|, \end{aligned}$$

uma vez que o gradiente é Lipschitz e o modelo a partir da i_j -ésima iteração é plenamente linear. Note que o primeiro termo converge para zero, pois $\|x_j - x_{i_j}\| \rightarrow 0$, o segundo também pois Δ_{i_j} converge para zero. Finalmente $\|g_{i_j}\|$ converge para zero pelo Lema 2.6, pois supondo por absurdo que uma subsequência não converge, então para Δ_{i_j} pequeno o suficiente, i_j seria uma iteração de sucesso, e teríamos uma contradição. Portanto, $\|\nabla f(x_j)\|$ converge para zero quando j vai para o infinito. \square

Vamos agora provar um lema relevante no contexto sem derivadas, onde mostraremos que o raio da região de confiança converge para zero.

Lema 2.9. [5, Lema 10.9]

$$\lim_{k \rightarrow +\infty} \Delta_k = 0.$$

Demonstração. Seja \mathcal{S} o conjunto de índices das iterações de sucesso. Se \mathcal{S} é finito, então pelo lema anterior temos o resultado. Vamos considerar o caso em que \mathcal{S} é infinito. Para

qualquer $k \in \mathcal{S}$, temos

$$f(x_k) - f(x_{k+1}) \geq \eta_1 [m_k(x_k) - m_k(x_k + p_k)].$$

Usando a hipótese sobre a fração do decréscimo pelo passo de Cauchy dado em (2.1), segue que

$$f(x_k) - f(x_{k+1}) \geq \eta_1 \frac{c_1}{2} \|g_k\| \min \left\{ \frac{\|g_k\|}{\|B_k\|}, \Delta_k \right\}. \quad (2.5)$$

Pelo Passo 1 do Algoritmo temos que $\|g_k\| \geq \min\{\varepsilon_c, \mu^{-1}\Delta_k\}$, então

$$f(x_k) - f(x_{k+1}) \geq \eta_1 \frac{c_1}{2} \min\{\varepsilon_c, \mu^{-1}\Delta_k\} \min \left\{ \frac{\min\{\varepsilon_c, \mu^{-1}\Delta_k\}}{\|B_k\|}, \Delta_k \right\}.$$

Como \mathcal{S} é infinito e f é limitada inferiormente, usando a hipótese que a Hessiana do modelo é limitada temos que o lado direito da expressão acima converge para zero. Logo, $\lim_{k \in \mathcal{S}} \Delta_k = 0$, e a prova está completa para todas as iterações de sucesso. Lembrando que pelo Algoritmo 2.1 o raio da região de confiança é aumentado no máximo por uma constante γ_{inc} e apenas nas iterações de sucesso. Agora, seja $k \notin \mathcal{S}$ o índice de uma iteração depois do primeiro sucesso. Então $\Delta_k \leq \gamma_{inc} \Delta_{s_k}$, onde s_k é o índice da última iteração de sucesso antes de k . Como $\Delta_{s_k} \rightarrow 0$, então $\Delta_k \rightarrow 0$ para $k \notin \mathcal{S}$, o que conclui a demonstração. \square

Uma maneira alternativa para a demonstração do lema anterior é analisarmos a expressão (2.5):

$$f(x_k) - f(x_{k+1}) \geq \rho_k \frac{c_1}{2} \|g_k\| \min \left\{ \frac{\|g_k\|}{\|B_k\|}, \Delta_k \right\}.$$

Como a sequência $\{f_k\}$ é decrescente e limitada inferiormente, $f(x_k) - f(x_{k+1}) \rightarrow 0$, e com isso

$$\rho_k \frac{c_1}{2} \|g_k\| \min \left\{ \frac{\|g_k\|}{\|B_k\|}, \Delta_k \right\} \rightarrow 0.$$

O que nos mostra que $\|g_k\| \rightarrow 0$ ou $\Delta_k \rightarrow 0$. Se $\|g_k\| \rightarrow 0$, existe k suficientemente grande para o qual $\|g_k\| \leq \varepsilon_c$, e pelo Algoritmo 2.2, $\Delta_k \leq \mu \|g_k\|$, e consequentemente $\Delta_k \rightarrow 0$.

O lema a seguir segue facilmente como consequência do Lema 2.7 e do lema anterior.

Lema 2.10. [5, Lema 10.10]

$$\liminf_{k \rightarrow +\infty} \|g_k\| = 0.$$

Demonstração. Supondo por absurdo que $\|g_k\| \geq \kappa_1$ para todo k . Pelo Lema 2.7 temos que $\Delta_k \geq \kappa_2$, o que pelo Lema 2.9 é uma contradição quando k tende ao infinito. \square

Vamos provar agora que quando a norma do gradiente do modelo $\|g_k\|$ converge para

zero em uma subsequência, então o gradiente da função $\|\nabla f(x_k)\|$ também converge para zero.

Lema 2.11. [5, Lema 10.11] Para qualquer subsequência $\{k_i\}$ tal que

$$\lim_{i \rightarrow +\infty} \|g_{k_i}\| = 0 \quad (2.6)$$

segue que

$$\lim_{i \rightarrow +\infty} \|\nabla f(x_{k_i})\| = 0.$$

Demonstração. Por (2.6), temos que $\|g_{k_i}\| \leq \varepsilon_c$ para i suficientemente grande. Pelo algoritmo temos que o modelo m_{k_i} é plenamente linear na bola $B(x_{k_i}; \Delta_{k_i})$, com $\Delta_{k_i} \leq \mu \|g_{k_i}\|$ para i suficientemente grande. Usando o fato que m_{k_i} é plenamente linear, temos

$$\|\nabla f(x_{k_i}) - g_{k_i}\| \leq \kappa_{eg} \Delta_{k_i} \leq \kappa_{eg} \mu \|g_{k_i}\|$$

Assim,

$$\|\nabla f(x_{k_i})\| \leq \|\nabla f(x_{k_i}) - g_{k_i}\| + \|g_{k_i}\| \leq (\kappa_{eg} \mu + 1) \|g_{k_i}\|$$

Como $\|g_{k_i}\| \rightarrow 0$, então $\|\nabla f(x_{k_i})\| \rightarrow 0$. □

O Teorema de convergência global, a seguir, segue imediatamente dos lemas anteriores.

Teorema 2.12. [5, Teorema 10.12] Sejam $x_0 \in \mathbb{R}^n$ e $\bar{\Delta} > 0$ dados. Suponha f é continuamente diferenciável com gradiente Lipschitz em um domínio aberto contendo N (o conjunto de amostra dos valores de f) e limitada no conjunto de nível $N(x_0)$, ou seja, $f(x) \geq \kappa_*$ para todo $x \in N(x_0)$. Suponha ainda que existe uma constante $\kappa_h > 0$, tal que $\|B_k\| \leq \kappa_h$. Então

$$\liminf_{k \rightarrow +\infty} \|\nabla f(x_k)\| = 0.$$

Demonstração. Pelo Lema 2.10

$$\liminf_{k \rightarrow +\infty} \|g_k\| = \lim_{k \rightarrow +\infty} \left\{ \inf_{m \geq k} \|g_m\| \right\} = 0,$$

e pelo Lema 2.11

$$\lim_{k \rightarrow +\infty} \left\{ \inf_{m \geq k} \|\nabla f(x_m)\| \right\} = \liminf_{k \rightarrow +\infty} \|\nabla f(x_k)\| = 0.$$

□

Se a sequência de iterandos é limitada, o Teorema 2.12 garante a existência de um ponto crítico de primeira ordem. Na verdade, somos capazes de provar que todos os

pontos limites da sequência de iterandos são pontos críticos de primeira ordem. É o que mostraremos no próximo teorema.

Teorema 2.13. [5, Teorema 10.13] *Sejam $x_0 \in \mathbb{R}^n$ e $\bar{\Delta} > 0$ dados. Se f é continuamente diferenciável com gradiente Lipschitz em um domínio aberto contendo N (o conjunto de amostra dos valores de f) e é limitada no conjunto de nível $N(x_0)$, ou seja, $f(x) \geq \kappa_*$ para todo $x \in N(x_0)$. E ainda se existe uma constante $\kappa_h > 0$, tal que $\|B_k\| \leq \kappa_h$. Então,*

$$\lim_{k \rightarrow +\infty} \nabla f(x_k) = 0.$$

Demonstração. Já mostramos no Lema 2.8 o resultado no caso em que o conjunto \mathcal{S} de iterações de sucesso é finito. Então, assumiremos que o conjunto \mathcal{S} é infinito. Vamos supor por contradição que exista uma subsequência $\{k_i\}$ de iterações de sucesso ou aceitáveis tais que $\|\nabla f(x_{k_i})\| \geq \varepsilon_0 > 0$, para algum $\varepsilon_0 > 0$ e para todo i . As demais iterações podemos ignorar pois nessas k é incrementado mas x_k não muda. Usando a contrapositiva do Lema 2.11 temos que $\|g_{k_i}\| \geq \varepsilon > 0$, para algum ε e para todo i suficientemente grande. Sem perda de generalidade, seja ε tal que

$$\varepsilon \leq \min \left\{ \frac{\varepsilon_0}{2(2 + \kappa_{eg}\mu)}, \varepsilon_c \right\}.$$

O Lema 2.10 assegura a existência, para cada k_i na subsequência, de uma primeira iteração $\ell_i > k_i$ tal que $\|g_{\ell_i}\| < \varepsilon$. Removendo elementos de $\{k_i\}$, obtemos uma outra subsequência indexada por $\{\ell_i\}$ tal que

$$\|g_k\| \geq \varepsilon \quad \text{para } k_i \leq k < \ell_i \quad \text{e} \quad \|g_{\ell_i}\| < \varepsilon \quad (2.7)$$

para i grande o suficiente.

Seja o conjunto

$$\mathcal{K} = \bigcup_{i \in \mathbb{N}_0} \{k \in \mathbb{N}_0 : k_i \leq k < \ell_i\}.$$

Temos que $\|g_k\| \geq \varepsilon$ para todo $k \in \mathcal{K}$. Pelo Lema 2.9

$$\lim_{k \rightarrow +\infty} \Delta_k = 0.$$

Pelo Lema 2.6 concluímos que para algum $k \in \mathcal{K}$ grande o suficiente a iteração k é de sucesso se o modelo é plenamente linear, ou a iteração é de melhoramento do modelo, caso este não seja plenamente linear.

Além disso, para cada $k \in \mathcal{K} \cap \mathcal{S}$ temos que

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\geq \eta_1 [m_k(x_k) - m_k(x_k + p_k)] \\ &\geq \eta_1 \frac{c_1}{2} \|g_k\| \min \left\{ \frac{\|g_k\|}{\kappa_h}, \Delta_k \right\} \\ &\geq \eta_1 \frac{c_1}{2} \varepsilon \min \left\{ \frac{\varepsilon}{\kappa_h}, \Delta_k \right\} \end{aligned}$$

e, para k suficientemente grande, $\Delta_k \leq \frac{\varepsilon}{\kappa_h}$. Então, para $k \in \mathcal{K} \cap \mathcal{S}$ grande

$$f(x_k) - f(x_{k+1}) \geq \eta_1 \frac{c_1}{2} \varepsilon \Delta_k,$$

consequentemente,

$$\Delta_k \leq \frac{2}{\eta_1 c_1 \varepsilon} [f(x_k) - f(x_{k+1})].$$

Como para todo $k \in \mathcal{K}$ grande a iteração é de sucesso ou de melhoramento do modelo, e quando é de melhoramento $x_k = x_{k+1}$, temos para i suficientemente grande

$$\|x_{k_i} - x_{\ell_i}\| \leq \sum_{\substack{j=k_i \\ j \in \mathcal{K} \cap \mathcal{S}}}^{\ell_i-1} \|x_j - x_{j+1}\| \leq \sum_{\substack{j=k_i \\ j \in \mathcal{K} \cap \mathcal{S}}}^{\ell_i-1} \Delta_j.$$

Mas

$$\sum_{\substack{j=k_i \\ j \in \mathcal{K} \cap \mathcal{S}}}^{\ell_i-1} \Delta_j \leq \sum_{\substack{j=k_i \\ j \in \mathcal{K} \cap \mathcal{S}}}^{\ell_i-1} \frac{2}{\eta_1 c_1 \varepsilon} [f(x_j) - f(x_{j+1})] \leq \frac{2}{\eta_1 c_1 \varepsilon} [f(x_{k_i}) - f(x_{\ell_i})].$$

Como a sequência $\{f_k\}$ é limitada e decrescente, segue que

$$\lim_{i \rightarrow +\infty} \|x_{k_i} - x_{\ell_i}\| = 0.$$

Agora, usando o fato que o gradiente é Lipschitz, o modelo é plenamente linear, o mecanismo do passo de criticalidade do Algoritmo 2.1 e (2.7) temos

$$\begin{aligned} \|\nabla f(x_{k_i})\| &\leq \|\nabla f(x_{k_i}) - \nabla f(x_{\ell_i})\| + \|\nabla f(x_{\ell_i}) - g_{\ell_i}\| + \|g_{\ell_i}\| \\ &\leq \nu \|x_{k_i} - x_{\ell_i}\| + \kappa_{eg} \mu \varepsilon + \varepsilon \\ &\leq (2 + \kappa_{eg} \mu) \varepsilon \leq \frac{1}{2} \varepsilon_0, \end{aligned}$$

o que contradiz a suposição inicial e prova o teorema. □

2.3 Método de Região de Confiança Sem Derivadas - Segunda Ordem

Para obter convergência global para pontos críticos de segunda ordem, o algoritmo requer uma ordem a mais de suavidade, de modo que precisamos de uma quantidade que expresse a estacionariedade de segunda ordem. Uma possibilidade é trabalhar com

$$\sigma_k^m = \max\{\|g_k\|, -\lambda_{\min}(B_k)\},$$

como medida de estacionariedade de segunda ordem do modelo.

O algoritmo para pontos críticos de segunda ordem segue basicamente os mesmos argumentos do Algoritmo 2.1. Uma diferença fundamental é que σ_k^m agora desempenha o papel de $\|g_k\|$. Outra diferença é a necessidade de trabalhar com modelos plenamente quadráticos. A terceira modificação principal é a necessidade de sermos capazes de resolver melhor o subproblema da região de confiança, de modo que o decréscimo seja melhor tanto que uma fração do decréscimo de Cauchy como uma fração do decréscimo do autopasso, quando temos a presença de curvatura negativa.

2.3.1 Algoritmo - Segunda Ordem

Vamos agora enunciar a versão do algoritmo de segunda ordem que iremos considerar.

Algoritmo 2.3. [5, Algoritmo 10.3] *Região de Confiança sem Derivadas - 2ª Ordem*

Dados:

- Uma classe de modelos plenamente quadráticos \mathcal{M} .
- O correspondente algoritmo de melhoramento.
- $x_0 \in \mathbb{R}^n$, $\bar{\Delta} > 0$, $\Delta_0^0 \in (0, \bar{\Delta}]$.
- $m_0^0(x_0 + s)$, com gradiente e Hessiana em $p = 0$ dados por g_0^0 e H_0^0 .
- $\sigma_0^{m,0} = \max\{\|g_0^0\|, -\lambda_{\min}(H_0^0)\}$.
- $0 \leq \eta_0 \leq \eta_1 < 1$, com $\eta_1 \neq 0$, $0 < \gamma < 1 < \gamma_{inc}$, $\varepsilon_c > 0$, $\mu > \beta > 0$ e $\omega \in (0, 1)$.

$k = 0$.

PASSO 1 - CRITICALIDADE

$$\begin{aligned} \text{SE } \sigma_0^{m,0} &> \varepsilon_c, \\ m_k &= m_k^0. \\ \Delta_k &= \Delta_k^0. \end{aligned}$$

SENÃO

Chame o algoritmo de melhoramento do modelo.

SE m_k^0 não é plenamente quadrático em $B(x_k, \Delta_k^0)$ OU $\Delta_k^0 > \mu\sigma_0^{m,0}$,

Aplique o Algoritmo 2.4 para encontrar um modelo \tilde{m}_k plenamente quadrático com gradiente e Hessiana em $p = 0$ dados por \tilde{g}_k e \tilde{B}_k , respectivamente.

$$m_k = \tilde{m}_k.$$

$$\Delta_k = \min\{\max\{\tilde{\Delta}_k, \beta\tilde{\sigma}_k^m\}, \Delta_k^0\}.$$

SENÃO

$$m_k = m_k^0.$$

$$\Delta_k = \Delta_k^0.$$

PASSO 2 - CÁLCULO DO PASSO

Calcule p_k que obtenha uma redução suficiente no modelo m_k , tal que $x_k + p_k \in B(x_k, \Delta_k)$.

PASSO 3 - ACEITAÇÃO DO PASSO

Calcule $\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(x_k) - m_k(x_k + p_k)}$.

SE $\rho_k \geq \eta_1$ OU (SE $\rho_k \geq \eta_0$ E m_k é plenamente quadrático),

$$x_{k+1} = x_k + p_k.$$

Inclua x_{k+1} no conjunto de amostra e obtenha um novo modelo m_{k+1}^0 .

SENÃO

$$x_{k+1} = x_k.$$

$$m_{k+1}^0 = m_k.$$

PASSO 4 - MELHORAMENTO DO MODELO

SE $\rho_k < \eta_1$,

Use o algoritmo de melhoramento do modelo.

Defina m_{k+1}^0 como m_k (possivelmente melhorado).

PASSO 5 - ATUALIZAÇÃO DO RAIOS

SE $\rho_k \geq \eta_1$,

SE $\Delta_k < \beta\sigma_k^m$,

$$\Delta_{k+1}^0 = \min\{\gamma_{inc}\Delta_k, \bar{\Delta}\}.$$

SENÃO

Escolha $\Delta_{k+1}^0 \in [\Delta_k, \min\{\gamma_{inc}, \bar{\Delta}\}]$.

SE m_k é plenamente quadrático,

$$\Delta_{k+1}^0 = \gamma\Delta_k.$$

SENÃO

$$\Delta_{k+1}^0 = \Delta_k.$$

$k = k + 1$.

A seguir descrevemos o algoritmo chamado no Passo 1 do algoritmo anterior.

Algoritmo 2.4. [5, Algoritmo 10.4] *Etapa de Criticalidade - 2ª Ordem*

$i = 0.$

$$m_k^{(0)} = m_k^0.$$

REPITA

$$i = i + 1.$$

Melhore o modelo até $m_k^{(i-1)}$ ser plenamente quadrático em $B(x_k, \omega^{i-1} \Delta_k^0)$.

Denote o novo modelo por $m_k^{(i)}$.

$$\tilde{\Delta}_k = \omega^{i-1} \Delta_k^0.$$

$$\tilde{m}_k = m_k^{(i)}.$$

ATÉ $\tilde{\Delta}_k \leq \mu(\sigma_k^m)^{(i)}$.

Novamente vamos usar as definições de iterações de **sucesso**, **aceitável**, **melhoramento do modelo** e **insucesso** que seguem os critérios dados no Algoritmo 2.1.

2.3.2 Convergência Global - Pontos críticos de segunda ordem

Para a convergência global a pontos críticos de segunda ordem, precisamos de uma ordem a mais de suavidade, dada pela Hipótese 7, que exige que a Hessiana de f seja Lipschitz. Será necessário também assumir que a função f é limitada inferiormente (Hipótese 8). Naturalmente, vamos também supor a existência de modelos plenamente quadráticos. Começamos por introduzir a notação

$$\sigma^m(x) = \max \{ \|\nabla m(x)\|, -\lambda_{\min}(\nabla^2 m(x)) \}$$

e

$$\sigma(x) = \max \{ \|\nabla f(x)\|, -\lambda_{\min}(\nabla^2 f(x)) \}.$$

Isso será importante para limitar a diferença entre o valor de $\sigma(x)$ da função e $\sigma^m(x)$ do modelo. Para este propósito, primeiro vamos encontrar um limitante para a diferença entre o menor autovalor da Hessiana da função e do seu correspondente modelo plenamente quadrático.

Proposição 2.14. [5, Proposição 10.14] *Seja f duas vezes continuamente diferenciável com Hessiana Lipschitz e m um modelo plenamente quadrático de f na bola $B(x; \Delta)$. Então, temos que*

$$|\lambda_{\min}(\nabla^2 f(x)) - \lambda_{\min}(\nabla^2 m(x))| \leq \kappa_{eh} \Delta.$$

Demonstração. Seja v um autovetor normalizado correspondente ao menor autovalor da

Hessiana do modelo, então pela Definição 2.4

$$\begin{aligned}
\lambda_{\min}(\nabla^2 f(x)) - \lambda_{\min}(\nabla^2 m(x)) &\leq v^\top \nabla^2 f(x)v - \lambda_{\min}(\nabla^2 m(x)) \\
&= v^\top \nabla^2 f(x)v - v^\top \nabla^2 m(x)v \\
&= v^\top [\nabla^2 f(x) - \nabla^2 m(x)]v \\
&\leq \|\nabla^2 f(x) - \nabla^2 m(x)\| \leq \kappa_{eh}\Delta.
\end{aligned}$$

Analogamente, seja u o autovetor normalizado correspondente ao menor autovalor da Hessiana de f , obtemos que

$$\begin{aligned}
\lambda_{\min}(\nabla^2 m(x)) - \lambda_{\min}(\nabla^2 f(x)) &\leq u^\top [\nabla^2 m(x) - \nabla^2 f(x)]u \\
&\leq \|\nabla^2 m(x) - \nabla^2 f(x)\| \leq \kappa_{eh}\Delta,
\end{aligned}$$

o que demonstra o resultado. □

Antes de provar o próximo resultado, precisamos de dois resultados auxiliares. O primeiro é o lema abaixo.

Lema 2.15. *Seja $a \geq 0$ e $b \in \mathbb{R}$. Então*

$$\max\{a, b\} = \max\{a, \max\{b, 0\}\}.$$

Demonstração. Se $\max\{b, 0\} = 0$, temos que $b \leq 0$. Como $a \geq 0$, temos que $\max\{a, b\} = \max\{a, \max\{b, 0\}\} = a$.

Por outro lado, se $\max\{b, 0\} = b$, é imediato que $\max\{a, \max\{b, 0\}\} = \max\{a, b\}$. □

O segundo resultado é o seguinte lema:

Lema 2.16. *Sejam a, b, c e d números reais não negativos. Então*

$$|\max\{a, b\} - \max\{c, d\}| \leq \max\{|a - c|, |b - d|\}.$$

Demonstração. Vamos separar a demonstração em quatro casos.

Caso I - Se $a \leq b$ e $c \leq d$, então

$$|\max\{a, b\} - \max\{c, d\}| = |b - d| \leq \max\{|a - c|, |b - d|\}.$$

Caso II - Se $b \leq a$ e $d \leq c$ então o resultado segue analogamente ao Caso I.

Caso III - Se $a \leq b$ e $d \leq c$, precisamos mostrar que

$$|\max\{a, b\} - \max\{c, d\}| = |b - c| \leq \max\{|a - c|, |b - d|\}.$$

Vamos supor que $|b - d| \leq |a - c|$ e mostrar que $|b - c| \leq |a - c|$.

Se $|b - d| \leq |a - c|$, então $b - d \leq |a - c|$. Por outro lado, como $d \leq c$, temos que $b - c \leq b - d$ o que implica $b - c \leq |a - c|$. Temos ainda que, $c - b \leq c - a \leq |a - c|$, pois $a \leq b$. Assim, $|b - c| \leq |a - c|$. Analogamente, o resultado vale quando $|a - c| \leq |b - d|$, onde conseguimos mostrar que $|a - c| \leq |b - c|$.

Caso IV - Se $b \leq a$ e $c \leq d$, basta fazer uma análise semelhante a está segunda, que concluímos o resultado. \square

O próximo resultado mostra que a diferença entre $\sigma(x)$ e $\sigma^m(x)$ é da ordem de Δ .

Lema 2.17. [5, Lema 10.15] *Seja Δ limitado por $\bar{\Delta}$. Seja f duas vezes continuamente diferenciável com Hessiana Lipschitz e m um modelo plenamente quadrático de f na bola $B(x; \Delta)$. Então, para algum $\kappa_\sigma > 0$, temos*

$$|\sigma(x) - \sigma^m(x)| \leq \kappa_\sigma \Delta. \quad (2.8)$$

Demonstração. Seja $\tau = -\lambda_{\min}(\nabla^2 f(x))$ e $\tau^m = -\lambda_{\min}(\nabla^2 m(x))$. Pelas definições de $\sigma(x)$ e $\sigma^m(x)$ temos que

$$\sigma(x) = \max\{\|\nabla f(x)\|, \tau\} \quad \text{e} \quad \sigma^m(x) = \max\{\|\nabla m(x)\|, \tau^m\},$$

então

$$|\sigma(x) - \sigma^m(x)| = |\max\{\|\nabla f(x)\|, \tau\} - \max\{\|\nabla m(x)\|, \tau^m\}|,$$

usando o Lema 2.15 temos

$$|\sigma(x) - \sigma^m(x)| = |\max\{\|\nabla f(x)\|, \max\{\tau, 0\}\} - \max\{\|\nabla m(x)\|, \max\{\tau^m, 0\}\}|,$$

usando o Lema 2.16 segue que

$$|\sigma(x) - \sigma^m(x)| \leq \max\{\|\|\nabla f(x)\| - \|\nabla m(x)\|\|, |\max\{\tau, 0\} - \max\{\tau^m, 0\}|\}.$$

Temos que $\|\|\nabla f(x)\| - \|\nabla m(x)\|\| \leq \kappa_{eg} \Delta^2 \leq \kappa_{eg} \Delta \bar{\Delta}$. Por outro lado, o segundo termo da expressão acima é dominado por $|\tau - \tau^m|$, que por sua vez é limitado por $\kappa_{eh} \Delta$,

de acordo com a Proposição 2.14. Portanto,

$$|\sigma(x) - \sigma^m(x)| \leq \max \{ \kappa_{eg} \bar{\Delta}, \kappa_{eh} \} \Delta = \kappa_\sigma \Delta.$$

□

Como acontece no caso de convergência de primeira ordem, vamos mostrar que na etapa de criticalidade é executado um número finito de passos de melhoramento.

Lema 2.18. [5, Lema 10.16] *Se $\sigma(x_k) \neq 0$, o Algoritmo 2.4 termina em um número finito de passos.*

Demonstração. Vamos supor que o Algoritmo 2.4 é infinito e provaremos que $\sigma(x_k) = 0$. Para começar, sabemos que m_k^0 não é certificadamente plenamente quadrático ou $\Delta_k^0 > \mu \sigma_k^{m,0}$. Então definimos $m_k^{(0)} = m_k^0$ e o modelo é melhorado até ser plenamente quadrático na bola $B(x_k, \omega^0 \Delta_k^0)$, em um número finito de passos de melhoramento. Se $(\sigma_k^m)^{(1)}$ do modelo resultante $m_k^{(1)}$ satisfaz $\mu(\sigma_k^m)^{(1)} \geq \omega^0 \Delta_k^0$, o processo para com

$$\tilde{\Delta}_k^0 = \omega^0 \Delta_k^0 \leq \mu(\sigma_k^m)^{(1)}.$$

Por outro lado, se $\mu(\sigma_k^m)^{(1)} < \omega^0 \Delta_k^0$, o modelo é melhorado até ser plenamente quadrático na bola $B(x_k, \omega^0 \Delta_k^0)$. Então, novamente, ou o procedimento para ou o raio é multiplicado outra vez por ω , e assim por diante.

A única situação possível para que o processo seja infinito e requeira um número infinito de passos de melhoramento ocorre se

$$\mu(\sigma_k^m)^{(i)} < \omega^{i-1} \Delta_k^0, \tag{2.9}$$

para todo $i \geq 1$. Esse argumento mostra que

$$\lim_{i \rightarrow +\infty} (\sigma_k^m)^{(i)} = 0.$$

Como temos que o modelo $m_k^{(i)}$ é plenamente quadrático em $B(x_k, \omega^{i-1} \Delta_k^0)$, temos pelo Lema 2.17 que

$$|\sigma(x_k) - (\sigma_k^m)^{(i)}| \leq \kappa_\sigma \omega^{i-1} \Delta_k^0,$$

para todo $i \geq 1$. Pela desigualdade triangular e por (2.9), temos que

$$|\sigma(x_k)| \leq |\sigma(x_k) - (\sigma_k^m)^{(i)}| + |(\sigma_k^m)^{(i)}| \leq \left(\kappa_\sigma + \frac{1}{\mu} \right) \omega^{i-1} \Delta_k^0.$$

Como $\omega \in (0, 1)$, isso implica que $\sigma(x_k) = 0$. □

Agora vamos mostrar que uma iteração pode ser de sucesso se o modelo corrente é plenamente quadrático e a o raio da região de confiança é pequeno o suficiente com respeito a σ_k^m .

Lema 2.19. [5, Lema 10.17] *Se m_k é plenamente quadrático em $B(x_k, \Delta_k)$ e*

$$\Delta_k \leq \min \left\{ \frac{\sigma_k^m}{\kappa_h}, \frac{c_0 \sigma_k^m (1 - \eta_1)}{4\kappa_{ef} \bar{\Delta}}, \frac{c_0 \sigma_k^m (1 - \eta_1)}{4\kappa_{ef}} \right\},$$

então a k -ésima iteração é de sucesso.

Demonstração. Pela condição (2.4) de máximo decréscimo entre o Passo de Cauchy e o autopasso temos

$$m_k(x_k) - m_k(x_k + p_k) \geq \frac{c_0}{2} \max \left\{ \|g_k\| \min \left\{ \frac{\|g_k\|}{\kappa_h}, \Delta_k \right\}, -\tau_k \Delta_k^2 \right\}.$$

Pela definição de $\sigma_k^m(x)$, $\sigma_k^m = \|\nabla m_k(x_k)\| = \|g_k\|$ ou $\sigma_k^m = -\lambda_{\min}(B_k) = -\tau_k$.

Primeiro faremos a análise quando $\sigma_k^m = \|g_k\|$. Como

$$\Delta_k \leq \frac{\sigma_k^m}{\kappa_h},$$

temos que

$$m_k(x_k) - m_k(x_k + p_k) \geq \frac{c_0}{2} \|g_k\| \Delta_k = \frac{c_0}{2} \sigma_k^m \Delta_k.$$

Como m_k é plenamente quadrático temos que $|f(x_k) - m_k(x_k)| \leq \kappa_{ef} \Delta_k^3$, de onde

$$\begin{aligned} |\rho_k - 1| &= \left| \frac{f(x_k) - f(x_k + p_k) - m_k(x_k) + m_k(x_k + p_k)}{m_k(x_k) - m_k(x_k + p_k)} \right| \\ &= \left| \frac{-f(x_k + p_k) + m_k(x_k + p_k)}{m_k(x_k) - m_k(x_k + p_k)} + \frac{f(x_k) - m_k(x_k)}{m_k(x_k) - m_k(x_k + p_k)} \right| \\ &\leq \left| \frac{f(x_k + p_k) - m_k(x_k + p_k)}{m_k(x_k) - m_k(x_k + p_k)} \right| + \left| \frac{f(x_k) - m_k(x_k)}{m_k(x_k) - m_k(x_k + p_k)} \right| \\ &\leq \frac{2\kappa_{ef} \Delta_k^3}{c_0 \sigma_k^m \Delta_k} + \frac{2\kappa_{ef} \Delta_k^3}{c_0 \sigma_k^m \Delta_k} \\ &\leq \frac{4\kappa_{ef} \bar{\Delta}}{c_0 \sigma_k^m} \Delta_k. \end{aligned}$$

Temos por hipótese que

$$\Delta_k \leq \frac{c_0 \sigma_k^m (1 - \eta_1)}{4\kappa_{ef} \bar{\Delta}},$$

assim

$$\frac{4\kappa_{ef} \bar{\Delta}}{c_0 \sigma_k^m} \Delta_k \leq (1 - \eta_1).$$

Logo,

$$|\rho_k - 1| \leq (1 - \eta_1).$$

Por outro lado, considerando o caso $\sigma_k^m = -\tau_k$, temos

$$m_k(x_k) - m_k(x_k + p_k) \geq -\frac{c_0}{2} \tau_k \Delta_k^2 = \frac{c_0}{2} \sigma_k^m \Delta_k^2.$$

Logo,

$$\begin{aligned} |\rho_k - 1| &\leq \left| \frac{f(x_k + p_k) - m_k(x_k + p_k)}{m_k(x_k) - m_k(\Delta_k^2)} \right| + \left| \frac{f(x_k) - m_k(x_k)}{m_k(x_k) - m_k(x_k + p_k)} \right| \\ &\leq \frac{4\kappa_{ef} \Delta_k^3}{c_0 \sigma_k^m \Delta_k^2} \\ &\leq 1 - \eta_1. \end{aligned}$$

Em ambos os casos, $-1 + \eta_1 \leq \rho_k - 1$, e portanto $\eta_1 \leq \rho_k$, o que nos mostra que a k -ésima iteração é de sucesso. \square

O próximo resultado segue do Lema 2.19. Sua demonstração é similar à do Lema 2.7.

Lema 2.20. [5, Lema 10.18] *Suponha que existe uma constante $\kappa_1 > 0$ tal que $\sigma_k^m \geq \kappa_1$ para todo k . Então existe uma constante $\kappa_2 > 0$ tal que*

$$\Delta_k \geq \kappa_2$$

para todo k .

Demonstração. Temos do Passo 1 do Algoritmo 2.3 que

$$\Delta_k \geq \min \{ \beta \sigma_k^m, \Delta_k^0 \}.$$

Pelo Lema 2.19 sempre que Δ_k é limitado pela constante

$$\bar{\kappa}_2 = \min \left\{ \frac{\sigma_k^m}{\kappa_h}, \frac{c_0 \sigma_k^m (1 - \eta_1)}{4\kappa_{ef} \bar{\Delta}}, \frac{c_0 \sigma_k^m (1 - \eta_1)}{4\kappa_{ef}} \right\},$$

a iteração é de sucesso (se m_k é plenamente quadrático) ou de melhora do modelo (se m_k não é plenamente quadrático), então pelo mecanismo de atualização do raio temos que $\Delta_{k+1}^0 \geq \Delta_k$.

Por outro lado se Δ_k não é limitado por $\bar{\kappa}_2$ e diminuimos o raio, o novo raio fica limitado por $\gamma\bar{\kappa}_2$. O que nos garante que

$$\Delta_k \geq \min \{ \beta\kappa_1, \Delta_0^0, \gamma\bar{\kappa}_2 \} = \kappa_2.$$

□

No próximo lema vamos considerar o que acontece quando o número de iterações de sucesso é finito. A demonstração é similar à do Lema 2.8.

Lema 2.21. [5, Lema 10.19] *Se o número de iterações de sucesso é finito, então*

$$\lim_{k \rightarrow +\infty} \sigma(x_k) = 0.$$

Demonstração. Vamos considerar as iterações que vêm depois da última iteração de sucesso. Sabemos que só podemos ter um número finito (uniformemente limitado por, digamos L) de iterações de melhoramento do modelo antes do modelo se tornar plenamente quadrático, e portanto há um número infinito de iterações que são aceitáveis ou de insucesso e, em ambos os casos, o raio da região de confiança é reduzido. Como não existem mais iterações de sucesso, nunca aumentamos Δ_k para k suficientemente grande. Além disso, Δ_k é reduzido, ao menos a cada L iterações, por um fator γ . Assim, Δ_k converge para zero.

Usando o Lema 2.17, temos que

$$\begin{aligned} |\sigma(x_k)| &= |\sigma(x_k) - \sigma^m(x_k) + \sigma^m(x_k)| \\ &\leq |\sigma(x_k) - \sigma^m(x_k)| + |\sigma^m(x_k)| \\ &\leq \kappa_\sigma \Delta_k + |\sigma(x_k)|. \end{aligned}$$

O primeiro termo converge para zero, pois Δ_k converge para zero. Além disso, $|\sigma(x_k)|$ converge para zero pelo Lema 2.19, pois supondo por absurdo que uma subsequência não converge, então para Δ_k pequeno o suficiente, k poderia ser uma iteração de sucesso o que seria uma contradição. Portanto, $\sigma(x_k)$ converge para zero quando j tende para o

infinito. □

Agora vamos provar que a sequência dos raios de região de confiança converge para zero. A demonstração é similar à prova do Lema 2.9.

Lema 2.22. [5, Lema 10.20]

$$\lim_{k \rightarrow +\infty} \Delta_k = 0$$

Demonstração. Seja \mathcal{S} o conjunto de índices das iterações de sucesso. Se \mathcal{S} é finito, então pelo lema anterior temos o resultado. Vamos considerar o caso em que \mathcal{S} é infinito. Para qualquer $k \in \mathcal{S}$, temos

$$f(x_k) - f(x_{k+1}) \geq \eta_1 [m_k(x_k) - m_k(x_k + p_k)].$$

Usando a hipótese sobre a fração do máximo decréscimo entre o autopasso e o passo de Cauchy dado em (2.4), segue que

$$f(x_k) - f(x_{k+1}) \geq \eta_1 \frac{c_0}{2} \max \left\{ \|g_k\| \min \left\{ \frac{\|g_k\|}{\|B_k\|}, \Delta_k \right\}, -\tau_k \Delta_k^2 \right\}.$$

Pelo Passo 1 do Algoritmo 2.3 temos que $\sigma_k^m \geq \min\{\varepsilon_c, \mu^{-1}\Delta_k\}$. Se na iteração k temos que $\|g_k\| \geq \max\{-\tau_k, 0\}$, então $\sigma_k^m = \|g_k\|$ e

$$f(x_k) - f(x_{k+1}) \geq \eta_1 \frac{c_0}{2} \min\{\varepsilon_c, \mu^{-1}\Delta_k\} \min \left\{ \frac{\min\{\varepsilon_c, \mu^{-1}\Delta_k\}}{\kappa_h}, \Delta_k \right\}. \quad (2.10)$$

Caso, $\|g_k\| < -\tau_k$, então $\sigma_k^m = -\tau_k$ e,

$$f(x_k) - f(x_{k+1}) \geq \eta_1 \frac{c_0}{2} \min\{\varepsilon_c, \mu^{-1}\Delta_k\} \Delta_k^2. \quad (2.11)$$

Temos duas subsequências de iterações de sucesso $\{k_i^1\}$ e $\{k_i^2\}$, possivelmente com sobreposição, onde (2.10) e (2.11) são satisfeitas, respectivamente. A união dessas subsequências contém todas as iterações de sucesso. Como \mathcal{S} é infinito e $\{f_k\}$ é limitada inferiormente, temos que ou $\{k_i^1\}$ ($\{k_i^2\}$) é finito ou o lado direito de (2.10) ((2.11)) converge para zero. Logo, $\lim_{k \in \mathcal{S}} \Delta_k = 0$, e a prova está completa se todas as iterações são de sucesso. Lembremos que pelo Algoritmo 2.3 o raio da região de confiança é aumentado no máximo por um fator γ_{inc} e apenas nas iterações de sucesso. Agora, seja $k \notin \mathcal{S}$ o índice de uma iteração depois do primeiro sucesso, então $\Delta_k \leq \gamma_{inc} \Delta_{s_k}$, onde s_k é o índice da última iteração de sucesso antes de k . Como $\Delta_{s_k} \rightarrow 0$, então $\Delta_k \rightarrow 0$ para $k \notin \mathcal{S}$, o que conclui a demonstração. □

O lema a seguir segue facilmente como consequência dos Lema 2.20 e do lema anterior.

Lema 2.23. [5, Lema 10.21]

$$\liminf_{k \rightarrow +\infty} \sigma_k^m = 0$$

Demonstração. Suponha por absurdo que $\sigma_k^m \geq \kappa_1$ para todo k . Pelo Lema 2.20 temos que existe uma constante κ_2 tal que $\Delta_k \geq \kappa_2$, o que contradiz o Lema 2.22. \square

Vamos agora verificar que o passo a criticalidade (Passo 1 do Algoritmo 2.3) garante que uma subsequência de iterados converge para um ponto crítico de segunda ordem, por meio do seguinte resultado auxiliar.

Lema 2.24. [5, Lema 10.22] Para qualquer subsequência $\{k_i\}$ tal que

$$\lim_{i \rightarrow +\infty} \sigma_{k_i}^m = 0 \tag{2.12}$$

segue que

$$\lim_{i \rightarrow +\infty} \sigma(x_{k_i}) = 0.$$

Demonstração. Por (2.12) temos que $\sigma_{k_i}^m \leq \varepsilon_c$ para i suficientemente grande. O Algoritmo garante que o modelo m_{k_i} é plenamente quadrático na bola $B(x_{k_i}; \Delta_{k_i})$, com $\Delta_{k_i} \leq \mu \sigma_{k_i}^m$ para i suficientemente grande. Usando o Lema 2.17, temos

$$\sigma(x_{k_i}) = (\sigma(x_{k_i}) - \sigma_{k_i}^m) + \sigma_{k_i}^m \leq (\kappa_\sigma \mu + 1) \sigma_{k_i}^m.$$

Desse modo, como $\sigma_{k_i}^m \rightarrow 0$ temos que $\sigma(x_{k_i}) \rightarrow 0$. \square

O teorema de convergência global a seguir segue imediatamente dos lemas anteriores.

Teorema 2.25. [5, Teorema 10.23] Sejam $x_0 \in \mathbb{R}^n$ e $\bar{\Delta} > 0$ dados. Se f é duas vezes continuamente diferenciável com Hessiana Lipschitz em um domínio aberto contendo N (o conjunto de amostra dos valores de f) e é limitada no conjunto de nível $N(x_0)$, ou seja, $f(x) \geq \kappa_*$ para todo $x \in N(x_0)$. E ainda existe constante $\kappa_h > 0$, tal que $\|B_k\| \leq \kappa_h$. Então

$$\liminf_{k \rightarrow +\infty} \sigma(x_k) = 0.$$

Se a sequência de iterandos é limitada, o Teorema 2.25 garante a existência de um ponto crítico de segunda ordem. Na verdade, somos capazes de provar, assim como na análise para pontos críticos de primeira ordem, que todos os pontos limites da sequência de iterandos são pontos críticos de segunda ordem. Para isso faremos uso do requisito adicional sobre o Passo 5 do Algoritmo 2.3, que impõe nas iterações de sucesso um aumento no raio da região de confiança se ele for muito pequeno comparado a σ_k^m .

Teorema 2.26. [5, Teorema 10.24] *Sejam $x_0 \in \mathbb{R}^n$ e $\Delta_{\max} > 0$ dados. Se f é duas vezes continuamente diferenciável com Hessiana Lipschitz em um domínio aberto contendo N (o conjunto de amostra dos valores de f) e f é limitada no conjunto de nível $N(x_0)$, ou seja, $f(x) \geq \kappa_*$ para todo $x \in N(x_0)$. E ainda existe constante $\kappa_h > 0$, tal que $\|B_k\| \leq \kappa_h$. Então*

$$\lim_{k \rightarrow +\infty} \sigma(x_k) = 0.$$

Demonstração. Já mostramos no Lema 2.21 o resultado quando o conjunto \mathcal{S} das iterações de sucesso é finito. Então, assumiremos que o conjunto \mathcal{S} é infinito. Vamos supor por contradição que exista uma subsequência $\{k_i\}$ de iterações de sucesso ou aceitáveis, tal que

$$\sigma(x_{k_i}) \geq \varepsilon_0 > 0, \quad (2.13)$$

para algum $\varepsilon_0 > 0$ e para todo i . Usando a contrapositiva do Lema 2.24, temos que $\sigma_k^m \geq \varepsilon > 0$, para algum $\varepsilon > 0$ e para todo i suficientemente grande. Sem perda de generalidade, seja ε tal que

$$\varepsilon \leq \min \left\{ \frac{\varepsilon_0}{2(2 + \kappa_\sigma \mu)}, \varepsilon_c \right\}. \quad (2.14)$$

O Lema 2.23 assegura a existência, para cada k_i na subsequência, de uma primeira iteração de sucesso ou aceitável $\ell_i > k_i$ tal que $\sigma_{\ell_i}^m < \varepsilon$. Removendo elementos de $\{k_i\}$, obtemos que existe uma outra subsequência indexada por $\{\ell_i\}$ tal que

$$\sigma_k^m \geq \varepsilon \quad \text{para } k_i \leq k < \ell_i \text{ e } \sigma_{\ell_i}^m < \varepsilon \quad (2.15)$$

para i grande o suficiente.

Seja o conjunto

$$\mathcal{K} = \bigcup_{i \in \mathbb{N}_0} \{k \in \mathbb{N}_0 \mid k_i \leq k < \ell_i\}.$$

Pelo Lemas 2.19 e 2.22 segue que, para algum $k \in \mathcal{K}$ grande o suficiente a iteração k é de sucesso se o modelo é plenamente quadrático, ou a iteração é de melhoramento do modelo, caso contrário. Vamos considerar a situação onde o índice $k \in \mathcal{K} \cap \mathcal{S} \setminus \mathcal{S}_+$, onde \mathcal{S}_+ são as iterações de sucesso em que $\Delta_k < \beta \sigma_k^m$.

Nesse caso, $\Delta_k \geq \beta \sigma_k^m \geq \beta \varepsilon$. Como, $\Delta_k \rightarrow 0$ para $k \in \mathcal{K}$, então $\mathcal{K} \cap \mathcal{S} \setminus \mathcal{S}_+$ contém apenas um número finito de iterações. Temos também que quando $k \in \mathcal{K} \cap \mathcal{S}$ então $k \in \mathcal{S}_+$ quando k é suficientemente grande.

Vamos mostrar que para $k \in \mathcal{K} \cap \mathcal{S}_+$ grande o suficiente, temos $\Delta_{k+1} = \gamma_{inc} \Delta_k$. Sabemos que quando $k \in \mathcal{S}_+$, $\Delta_{k+1}^0 = \gamma_{inc} \Delta_k$ depois do Passo 5 do Algoritmo 2.3. No entanto, Δ_{k+1}^0 pode ser reduzido no Passo 1, de alguma iteração seguinte, e para isso $\Delta_{k+1} \geq \beta \tilde{\sigma}_{k+1}^m = \beta \sigma_{k+1}^m \geq \beta \varepsilon$, mas isso só acontece um número finito de vezes, pois temos que $\Delta_k \rightarrow 0$. Então, para $k \in \mathcal{K} \cap \mathcal{S}_+$ grande o suficiente, temos $\Delta_{k+1} = \gamma_{inc} \Delta_k$.

Seja $\mathcal{S}_+^i = [k_i, \ell_i - 1] \cap \mathcal{S}_+ = \{j_i^1, j_i^2, \dots, j_i^*\}$ o conjunto de índices de iterações de sucesso que estão no intervalo $[k_i, \ell_i - 1]$. Pelo passo de atualização do raio e fato de que o ponto e o raio são mantidos nas iterações de melhoramento do modelo, para i grande temos que

$$\begin{aligned} \|x_{k_i} - x_{\ell_i}\| &\leq \sum_{j \in \mathcal{S}_+^i} \|x_j - x_{j+1}\| \leq \sum_{j \in \mathcal{S}_+^i} \Delta_j \\ &\leq \sum_{j \in \mathcal{S}_+^i} \left(\frac{1}{\gamma_{inc}} \right)^{j_i^* - j} \Delta_{j_i^*} \leq \frac{\gamma_{inc}}{\gamma_{inc} - 1} \Delta_{j_i^*}. \end{aligned}$$

Como $\Delta_{j_i^*} \rightarrow 0$, concluímos que

$$\lim_{i \rightarrow +\infty} \|x_{k_i} - x_{\ell_i}\| = 0.$$

Além disso,

$$\sigma(x_{k_i}) = (\sigma(x_{k_i}) - \sigma(x_{\ell_i})) + (\sigma(x_{\ell_i}) - \sigma_{\ell_i}^m) + \sigma_{\ell_i}^m.$$

O primeiro termo do lado direito tende a zero por causa da continuidade Lipschitz de $\sigma(x)$, e assim é limitado por ε para i suficientemente grande. O terceiro termo é limitado por ε por (2.15). Para o segundo termo, usamos o fato de que, a partir de (2.14) e o mecanismo do passo de criticalidade na iteração ℓ_i , o modelo m_{ℓ_i} é plenamente quadrático em $B(x_{\ell_i}, \mu \sigma_{\ell_i}^m)$. Usando (2.8) e (2.15), também deduzimos que o segundo termo é limitado por $k_\sigma \mu \varepsilon$ (para i suficientemente grande). Como consequência, obtemos a partir destes limites e (2.14), que

$$\sigma(x_{k_i}) \leq (2 + k_\sigma \mu) \varepsilon \leq \frac{1}{2} \varepsilon_0$$

para i grande o suficiente, o que contradiz (2.13). Então nossa hipótese inicial deve ser falsa, o que prova o teorema. \square

Capítulo 3

Interpolação Polinomial

Neste capítulo abordamos as propriedades dos modelos de interpolação por polinômios quadráticos. O que discutiremos pode ser aplicado à interpolação polinomial de qualquer grau. No entanto, os polinômios quadráticos são de maior interesse para nós, pois podem ser considerados os modelos não-lineares mais simples, e muitas vezes os mais eficientes. Por essa razão, todos os exemplos e imagens são limitadas ao caso quadrático.

O objetivo deste capítulo é estabelecer uma base teórica para a utilização de interpolação polinomial como aproximação da função a otimizar sem o uso das derivadas. Para assegurar a convergência global de um algoritmo de otimização que utiliza um modelo da função objetivo, é normalmente necessário garantir uma certa qualidade deste modelo. Os resultados aqui apresentados seguem as ideias de Conn, Scheinberg e Vicente em [3] e [5].

Quando um modelo é uma expansão por Taylor de primeira ou segunda ordem, a qualidade do modelo é facilmente analisada em termos do erro da expansão. Já no caso de interpolação polinomial existem limites semelhantes, mas, ao contrário da expansão de Taylor, não dependem só do centro da expansão e da função que está sendo aproximada, mas também do conjunto de pontos de interpolação. Assim, a fim de manter a qualidade do modelo de interpolação, é necessário manter a qualidade do conjunto de interpolação. Examinaremos constantes que caracterizam a qualidade de um conjunto de interpolação. Vamos estudar a relação entre essas constantes e o papel que desempenham no erro entre a interpolação polinomial e a função (e entre os de suas derivadas).

Vamos considerar alguns conceitos básicos da interpolação polinomial. Diferentes bases para o espaço de polinômios apresentam também diferentes graus de dificuldade para a interpolação. Seja \mathcal{P}_n^d o espaço dos polinômios de grau menor ou igual a d no \mathbb{R}^n . A

dimensão desse espaço é $p + 1$. Se $d = 1$, então $p + 1 = n + 1$ e se $d = 2$ temos que $p + 1 = (n + 1)(n + 2)/2$. Em geral, $\dim(\mathcal{P}_n^d) = \frac{(n+d)!}{n!d!}$.

Uma base $\phi = \{\phi_0(x), \phi_1(x), \dots, \phi_p(x)\}$ de \mathcal{P}_n^d é um conjunto de $p + 1$ polinômios de grau menor ou igual a d que gera \mathcal{P}_n^d . Como ϕ é uma base de \mathcal{P}_n^d , então qualquer polinômio $m(x) \in \mathcal{P}_n^d$ pode ser escrito como

$$m(x) = \sum_{j=0}^p \alpha_j \phi_j(x),$$

onde α_j são coeficientes reais.

Podemos considerar várias bases para o espaço dos polinômios. Por exemplo, uma base canônica, com monômios de coeficiente 1, pode ser escrita como

$$\phi = \{1, x_1, x_2, \dots, x_n, x_1^2, x_1x_2, \dots, x_1x_n, x_2^2, x_2x_3, \dots, x_2x_n, \dots, x_n^d\}.$$

Uma base bastante importante no contexto que vamos abordar é a chamada base natural. A base natural $\bar{\phi}$ pode ser convenientemente descrita via o uso de *multiíndices*. Seja um vetor $\alpha^i = (\alpha_1^i, \dots, \alpha_n^i) \in \mathbb{N}_0^n$ chamado de multiíndice, e para qualquer $x \in \mathbb{R}^n$, seja x^{α^i} definido como

$$x^{\alpha^i} = \prod_{j=1}^n x_j^{\alpha_j^i}.$$

Também, definamos

$$|\alpha^i| = \sum_{j=1}^n \alpha_j^i \quad \text{e} \quad (\alpha^i)! = \prod_{j=1}^n (\alpha_j^i!).$$

Então os elementos da base natural são

$$\bar{\phi}_i(x) = \frac{1}{(\alpha^i)!} x^{\alpha^i}, \quad i = 0, \dots, p, \quad \text{e} \quad |\alpha^i| \leq d.$$

A base natural pode ser escrita como

$$\bar{\phi} = \{1, x_1, x_2, \dots, x_n, x_1^2/2, x_1x_2, \dots, x_1x_n, \dots, x_{n-1}^{d-1}/(d-1)!, x_n^d/d!\},$$

onde cada termo da base canônica é dividido pelo produtório dos fatoriais de cada expoente.

A base natural é a base de polinômios que aparece na expansão por Taylor.

3.1 Interpolação Polinomial

Dizemos que o polinômio m interpola uma função f em um ponto y se $m(y) = f(y)$. Se temos um conjunto $Y = \{y^0, y^1, \dots, y^p\} \subset \mathbb{R}^n$ de pontos de interpolação, e m denota um polinômio de grau menor ou igual a d que interpola f nos pontos de Y , os coeficientes $\alpha_0, \dots, \alpha_p$ em termos de uma base ϕ podem ser determinados pelas condições de interpolação

$$m(y^i) = \sum_{j=0}^p \alpha_j \phi_j(y^i) = f(y^i), \quad i = 0, \dots, p.$$

Esta condição forma um sistema linear de equações em termos dos coeficientes de interpolação, o qual podemos escrever na forma matricial como

$$M(\phi, Y)\alpha_\phi = f(Y),$$

onde

$$M(\phi, Y) = \begin{bmatrix} \phi_0(y^0) & \phi_1(y^0) & \cdots & \phi_p(y^0) \\ \phi_0(y^1) & \phi_1(y^1) & \cdots & \phi_p(y^1) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(y^p) & \phi_1(y^p) & \cdots & \phi_p(y^p) \end{bmatrix},$$

$$\alpha_\phi = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_p \end{bmatrix} \quad \text{e} \quad f(Y) = \begin{bmatrix} f(y^0) \\ f(y^1) \\ \vdots \\ f(y^p) \end{bmatrix}.$$

Para que o sistema acima tenha uma única solução, é necessário que a matrix $M(\phi, Y)$ seja não singular.

Definição 3.1. [5, Definição 3.1] *O conjunto $Y = \{y^0, y^1, \dots, y^p\}$ está posicionado para interpolação polinomial em \mathbb{R}^n se a matriz correspondente $M(\phi, Y)$ é não singular para alguma base ϕ em \mathcal{P}_n^d .*

Como todas as bases em espaços vetoriais de dimensão finita são equivalentes, a definição acima vale se $M(\phi, Y)$ for não singular, qualquer que seja a base ϕ de \mathcal{P}_n^d .

Dado o espaço dos polinômios \mathcal{P}_n^d e uma base ϕ , denotaremos

$$\phi(x) = [\phi_0(x), \phi_1(x), \dots, \phi_p(x)]^\top$$

como o vetor em \mathbb{R}^{p+1} cujas entradas são os valores dos elementos da base polinomial no

ponto x .

Lema 3.2. [5, Lema 3.2] Dada uma função $f : \mathbb{R}^n \rightarrow \mathbb{R}$ e um conjunto posicionado para interpolação polinomial $Y \in \mathbb{R}^n$, o polinômio interpolador m de f em Y existe e é único.

Demonstração. Como Y é posicionado para interpolação polinomial, m existe e é único para uma base dada ϕ . O que precisamos mostrar é que m não depende da escolha da base. Assim, seja $\psi(x) = P^\top \phi(x)$ uma outra base, onde P é uma matriz $(p+1) \times (p+1)$, não singular.

Temos que,

$$M(\psi, Y) = M(P^\top \phi, Y) = M(\phi, Y)P,$$

de fato,

$$\psi_i(x) = \sum_{j=1}^{p+1} P_{j,i+1} \phi_{j-1}, \quad i = 0, \dots, p.$$

E

$$M(\phi, Y)P = \begin{bmatrix} \phi_0(y^0) & \phi_1(y^0) & \cdots & \phi_p(y^0) \\ \phi_0(y^1) & \phi_1(y^1) & \cdots & \phi_p(y^1) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(y^p) & \phi_1(y^p) & \cdots & \phi_p(y^p) \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1,(p+1)} \\ P_{21} & P_{22} & \cdots & P_{2,(p+1)} \\ \vdots & \vdots & \ddots & \vdots \\ P_{(p+1),1} & P_{(p+1),2} & \cdots & P_{(p+1),(p+1)} \end{bmatrix}.$$

Seja α_ϕ a solução de $M(\phi, Y)\alpha_\phi = f(Y)$ e α_ψ a solução de $M(\psi, Y)\alpha_\psi = f(Y)$, então

$$M(\phi, Y)P\alpha_\psi = f(Y),$$

o que implica

$$\alpha_\psi = P^{-1}(M(\phi, Y))^{-1}f(Y).$$

Mas,

$$\alpha_\psi^\top \psi(x) = f(Y)^\top (M(\phi, Y))^{-\top} P^{-\top} P^\top \phi(x) = f(Y)^\top (M(\phi, Y))^{-\top} \phi(x) = \alpha_\phi^\top \phi(x).$$

De onde, $m(x) = \alpha_\psi^\top \psi(x) = \alpha_\phi^\top \phi(x)$. □

3.2 Polinômios de Lagrange

Agora vamos estudar os polinômios de Lagrange, que representam papel fundamental na teoria de interpolação. Uma base ℓ formada pelos polinômios de Lagrange será chamada

de *Base de Lagrange* e, como veremos, se Y é um conjunto posicionado então a matriz $M(\ell, Y)$ é a identidade. Começamos pela definição de tais polinômios.

Definição 3.3. [5, Definição 3.3] *Dado um conjunto de pontos de interpolação $Y = \{y^0, y^1, \dots, y^p\}$, uma base de $p + 1$ polinômios $\ell_j(x)$, $j = 0, 1, \dots, p$, em \mathcal{P}_n^d é chamada base de Polinômios de Lagrange se*

$$\ell_j(y^i) = \delta_{ij} = \begin{cases} 1, & \text{se } i = j, \\ 0, & \text{se } i \neq j. \end{cases}$$

Se Y é posicionado, então os Polinômios de Lagrange existem e são únicos. Os polinômios de Lagrange gozam de propriedades interessantes, algumas relacionadas à seguir.

Lema 3.4. [5, Lema 3.4] *Se Y é posicionado, então a base de Polinômios de Lagrange existe e está unicamente definida.*

Demonstração. A prova da existência segue do fato de que cada polinômio de Lagrange $\lambda_j(x)$ é um polinômio interpolador da função que se anula em todos os pontos de Y exceto em y^j onde é igual a 1, ou seja, $\lambda_j(x)$ é polinômio interpolador da função $h_j : Y \rightarrow \mathbb{R}$ tal que

$$h_j(y^i) = \begin{cases} 1, & \text{se } i = j, \\ 0, & \text{se } i \neq j. \end{cases}$$

Logo, pelo Lema 3.2, $\lambda_j(x)$ existem e são únicos para $j = 0, 1, \dots, p$. Como temos $p + 1$ polinômios de Lagrange, estes polinômios formam uma base unicamente determinada. \square

A existência dos Polinômios de Lagrange implica que um conjunto é posicionado. De fato, se os polinômios de Lagrange existem, então existe uma matriz A_ϕ , onde as colunas são os coeficientes dos polinômios em uma base ϕ qualquer, tal que $M(\phi, Y)A_\phi = I$, ou seja, $M(\phi, Y)$ é não singular. Verifiquemos tal afirmação. Seja $\ell_j(x) = \alpha_0^j(x) + \dots + \alpha_p^j(x)$ o j -ésimo polinômio de Lagrange escrito como combinação linear dos elementos da base ϕ . Então definamos A_ϕ como

$$A_\phi = \begin{bmatrix} \alpha_0^0 & \alpha_0^1 & \cdots & \alpha_0^p \\ \alpha_1^0 & \alpha_1^1 & \cdots & \alpha_1^p \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_p^0 & \alpha_p^1 & \cdots & \alpha_p^p \end{bmatrix}$$

e

$$M(\phi, Y) = \begin{bmatrix} \phi_0(y^0) & \phi_1(y^0) & \cdots & \phi_p(y^0) \\ \phi_0(y^1) & \phi_1(y^1) & \cdots & \phi_p(y^1) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(y^p) & \phi_1(y^p) & \cdots & \phi_p(y^p) \end{bmatrix}.$$

Logo,

$$M(\phi, Y)A_\phi = \begin{bmatrix} \sum_{i=0}^p \alpha_i^0 \phi_i(y^0) & \sum_{i=0}^p \alpha_i^1 \phi_i(y^0) & \cdots & \sum_{i=0}^p \alpha_i^p \phi_i(y^0) \\ \sum_{i=0}^p \alpha_i^0 \phi_i(y^1) & \sum_{i=0}^p \alpha_i^1 \phi_i(y^1) & \cdots & \sum_{i=0}^p \alpha_i^p \phi_i(y^1) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^p \alpha_i^0 \phi_i(y^p) & \sum_{i=0}^p \alpha_i^1 \phi_i(y^p) & \cdots & \sum_{i=0}^p \alpha_i^p \phi_i(y^p) \end{bmatrix}.$$

Ou seja,

$$M(\phi, Y)A_\phi = \begin{bmatrix} \ell_0(y^0) & \ell_0(y^1) & \cdots & \ell_0(y^p) \\ \ell_1(y^0) & \ell_1(y^1) & \cdots & \ell_1(y^p) \\ \vdots & \vdots & \ddots & \vdots \\ \ell_p(y^0) & \ell_p(y^1) & \cdots & \ell_p(y^p) \end{bmatrix} = I.$$

Para mostrar que o conjunto de polinômios de Lagrange forma uma base para \mathcal{P}_n^d , nós precisamos mostrar que qualquer polinômio m em \mathcal{P}_n^d é uma combinação linear desses polinômios. Sabemos que qualquer polinômio é unicamente determinado por seus valores em Y , desde que Y seja posicionado. O próximo resultado mostra que esses valores, de fato, são os coeficientes de m quando escrito em termos da base de Lagrange.

Lema 3.5. [5, Lema 3.5] *Para qualquer função $f : \mathbb{R}^n \rightarrow \mathbb{R}$ e qualquer conjunto posicionado $Y = \{y^0, y^1, \dots, y^p\} \subset \mathbb{R}^n$, o único polinômio que interpola f em Y pode ser escrito como*

$$m(x) = \sum_{i=0}^p f(y^i) \ell_i(x),$$

onde $\{\ell_i(x), i = 0, 1, \dots, p\}$ é a base de Polinômios de Lagrange para Y .

Demonstração. Como Y é posicionado, m é único. Seja ϕ uma base, então $m(x) = \alpha_\phi^\top \phi(x)$, onde α_ϕ é solução de $M(\phi, Y)\alpha_\phi = f(Y)$. Então, $\alpha_\phi = M(\phi, Y)^{-1}f(Y)$, o que implica que

$$\alpha_\phi^\top = f(Y)^\top M(\phi, Y)^{-\top}.$$

Por outro lado, $\ell_j(x) = \alpha_j^\top \phi_j(x)$, ou matricialmente $\ell(x) = A_\phi^\top \phi(x)$. Assim

$$\begin{aligned} m(x) &= f(Y)^\top M(\phi, Y)^{-\top} \phi(x) \\ &= f(Y)^\top M(\phi, Y)^{-\top} A_\phi^{-\top} \ell(x). \end{aligned}$$

Temos que $M(\phi, Y)A_\phi = I$, logo

$$m(x) = f(Y)^\top \ell(x) = \sum_{i=0}^p f(y^i) \ell_i(x).$$

□

Para melhor entendimento ilustraremos os resultados acima com um exemplo.

Exemplo 3.1.

Seja a função $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ definida por $f(x) = 3 + x_1 + x_2^2 + 2x_1x_2 + 4x_1^3$ e o conjunto de interpolação $Y = \{(0, 0), (1, 0), (0, 1), (2, 0), (0, 2), (1, 1)\}$. Queremos encontrar o polinômio quadrático m que interpola f nos pontos de Y . Primeiramente vamos encontrar os polinômios de Lagrange referentes ao conjunto Y . Para isso resolvemos os 6 sistemas lineares:

$$\sum_{i=0}^5 \alpha_i^j \phi(y^i) = \ell_j(y^i),$$

para todo $j = 0, 1, \dots, 5$, onde $\phi = \{1, x_1, x_2, x_1^2, x_2^2, x_1x_2\}$ é uma base para \mathcal{P}_2^2 e $\ell_j(y^i)$ são os polinômios de Lagrange nos pontos y^i . Ao resolver os sistemas obtemos:

$$\begin{aligned} \ell_0(x) &= 1 - \frac{3}{2}x_1 - \frac{3}{2}x_2 + \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2 + x_1x_2, \\ \ell_1(x) &= 2x_1 - x_1^2 - x_1x_2, \\ \ell_2(x) &= 2x_2 - x_2^2 - x_1x_2, \\ \ell_3(x) &= -\frac{1}{2}x_1 + \frac{1}{2}x_1^2, \\ \ell_4(x) &= -\frac{1}{2}x_2 + \frac{1}{2}x_2^2, \\ \ell_5(x) &= x_1x_2. \end{aligned}$$

É fácil verificar que $\ell_j(y^i) = \delta_{ij}$ como esperado. E como $f(y^0) = 3$, $f(y^1) = 7$, $f(y^2) = 4$, $f(y^3) = 27$, $f(y^4) = 10$, $f(y^5) = 12$, pelo Lema 3.5 temos que

$$\begin{aligned} m(x) &= 3 \left(1 - \frac{3}{2}x_1 - \frac{3}{2}x_2 + \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2 + x_1x_2 \right) + 7 (2x_1 - x_1^2 - x_1x_2) + \\ &\quad + 4 (2x_2 - x_2^2 - x_1x_2) + 27 \left(-\frac{1}{2}x_1 + \frac{1}{2}x_1^2 \right) + 10 \left(-\frac{1}{2}x_2 + \frac{1}{2}x_2^2 \right) + 12x_1x_2 \\ &= 3 - 4x_1 - \frac{3}{2}x_2 + 8x_1^2 + \frac{5}{2}x_2^2 + 4x_1x_2. \end{aligned}$$

Conseguimos assim o polinômio $m(x) = 3 - 4x_1 - \frac{3}{2}x_2 + 8x_1^2 + \frac{5}{2}x_2^2 + 4x_1x_2$ que interpola a função $f(x) = 3 + x_1 + x_2^2 + 2x_1x_2 + 4x_1^3$ nos pontos do conjunto Y . Veremos na Seção 3.7

um algoritmo para encontrar os polinômios de Lagrange sem a necessidade de resolvermos $p + 1$ sistemas lineares. \square

Os Polinômios de Lagrange são importantes na teoria de interpolação polinomial, tanto unidimensional como em espaços de dimensão 2 ou mais. Além disso algumas de suas propriedades são bastante úteis em modelos da otimização sem derivadas. Por isso, é importante estudarmos as propriedades dos Polinômios de Lagrange. Iremos agora discutir maneiras alternativas para defini-los.

Dado um conjunto $Y = \{y^0, y^1, \dots, y^p\} \subset \mathbb{R}^n$ e um vetor $x \in \mathbb{R}^n$, como a matriz $M(\phi, Y)$ é não singular, podemos expressar o vetor $\phi(x)$ unicamente em termos do vetor $\phi(y^i)$, $i = 0, \dots, p$, como

$$\sum_{i=0}^p \lambda_i(x) \phi(y^i) = \phi(x)$$

ou, da forma matricial,

$$M(\phi, Y)^\top \lambda(x) = \phi(x), \quad \text{onde } \lambda(x) = [\lambda_0(x), \dots, \lambda_p(x)]^\top. \quad (3.1)$$

Com isso, temos a primeira maneira alternativa para encontrar os polinômios de Lagrange: basta resolvermos o sistema linear dado em (3.1).

Para a segunda alternativa, dado um conjunto Y e um ponto x , considere o conjunto $Y_i(x) = Y \setminus \{y^i\} \cup \{x\}$, $i = 0, \dots, p$. Ao aplicarmos a Regra de Cramer em (3.1) obtemos

$$\lambda_i(x) = \frac{\det(M(\phi, Y_i(x)))}{\det(M(\phi, Y))}.$$

Esse fato mostra que $\lambda(x)$ não depende da escolha da base ϕ , uma vez que o espaço \mathcal{P}_n^d é fixado.

Considere um conjunto $\phi(Y) = \{\phi(y^i), i = 0, \dots, p\} \in \mathbb{R}^{p+1}$. Seja $\text{vol}(\phi(Y))$ o volume do simplex de vértices em $\phi(Y)$, dado por

$$\text{vol}(\phi(Y)) = \frac{|\det(M(\phi, Y))|}{(p+1)!}.$$

Então,

$$|\lambda_i(x)| = \frac{\text{vol}(\phi(Y_i(x)))}{\text{vol}(\phi(Y))}.$$

Ou seja, o valor absoluto do i -ésimo polinômio de Lagrange em um ponto dado x é a mudança no volume de $\phi(Y)$, quando y^i é trocado por x .

3.3 Medidas de Posicionamento

Vimos que a não singularidade de $M(\phi, Y)$ é um indicador de posicionamento. Uma questão natural é saber se existe conjunto melhor posicionado que outros, e a resposta é sim. Outra pergunta que podemos fazer é: será que o número de condição de $M(\phi, Y)$ é um indicador de bom posicionamento? A resposta, em geral, é não, uma vez que o número de condição de $M(\phi, Y)$ depende da escolha de ϕ . Além disso, para qualquer conjunto de interpolação Y , é possível escolhermos a base ϕ de modo que o número de condição de $M(\phi, Y)$ seja igual a qualquer número entre 1 e $+\infty$. Por outro lado, para qualquer escolha de ϕ fixa, o número de condição de $M(\phi, Y)$ depende da escala de Y . Assim, o número de condição de $M(\phi, Y)$ não é considerado um bom indicativo para o nível de posicionamento de um conjunto de pontos. No entanto, vamos voltar a esta questão e mostrar que, para uma escolha específica de ϕ , a base natural, e para \hat{Y} , uma versão reduzida de Y , o número de condição de $M(\bar{\phi}, \hat{Y})$ é uma medida significativa de bom posicionamento.

Uma clássica medida de posicionamento de um conjunto de pontos Y em uma região B é dada em [1], onde é mostrado que um polinômio interpolador de \mathcal{P}_n^d satisfaz

$$\|\mathcal{D}^r f(x) - \mathcal{D}^r m(x)\| \leq \frac{1}{(d+1)!} \nu_d \sum_{i=0}^p \|y^i - x\|^{(d+1)} \|\mathcal{D}^r \ell_i(x)\|,$$

onde \mathcal{D}^r denota a r -ésima derivada da função e ν_d é um limitante superior para $\mathcal{D}^{(d+1)} f(x)$. Notemos que f precisa ter $(d+1)$ derivadas em x . Quando $r = 0$, o limitante do valor da função se reduz a

$$|f(x) - m(x)| \leq \frac{1}{(d+1)!} (p+1) \nu_d \Lambda_\ell \Delta^{d+1},$$

onde

$$\Lambda_\ell = \max_{0 \leq i \leq p} \max_{x \in B(Y)} |\ell_i(x)|,$$

e Δ é o diâmetro da menor bola $B(Y)$ contendo Y .

A partir desses resultados vamos definir Λ -posicionamento, onde Λ é a constante de posicionamento do conjunto Y .

Definição 3.6. [5, Definição 3.6] *Sejam $\Lambda > 0$ e um conjunto $B \in \mathbb{R}^n$ dados. Seja $\phi = \{\phi_0(x), \phi_1(x), \dots, \phi_p(x)\}$ uma base em \mathcal{P}_n^d . Um conjunto $Y = \{y^0, y^1, \dots, y^p\}$ é dito Λ -posicionado em B se e somente se*

1. para a base de Polinômios de Lagrange associado com Y

$$\Lambda \geq \max_{0 \leq i \leq p} \max_{x \in B(Y)} |\ell_i(x)|,$$

ou, equivalentemente,

2. para qualquer $x \in B$ existe $\lambda(x) \in \mathbb{R}^{p+1}$ tal que

$$\sum_{i=0}^p \lambda_i(x) \phi(y^i) = \phi(x) \quad \text{com} \quad \|\lambda(x)\|_\infty \leq \Lambda,$$

ou, equivalentemente,

3. ao substituir qualquer ponto em Y por $x \in B$, aumentamos o volume do conjunto $\{\phi_0(x), \phi_1(x), \dots, \phi_p(x)\}$ no máximo por um fator Λ .

Essa definição não implica ou requer que o conjunto de amostra Y esteja contido no conjunto B onde o valor absoluto dos polinômios de Lagrange são maximizados. De fato, o conjunto Y pode estar arbitrariamente longe de B desde que todos os polinômios de Lagrange estejam limitados em valor absoluto em B . No entanto, veremos que para garantir a validade dos algoritmos que constroem conjuntos Λ -posicionados iremos gerar pontos em B .

Mostraremos agora algumas propriedades do Λ -posicionamento.

Lema 3.7. [5, Lema 3.7]

1. Se B contém pontos em Y e Y é Λ -posicionado em B , então $\Lambda \geq 1$.
2. Se Y é Λ -posicionado em B , então Y é Λ -posicionado em um subconjunto de B .
3. Se Y é Λ -posicionado em B para um constante Λ , então Y é $\tilde{\Lambda}$ -posicionado em B para qualquer $\tilde{\Lambda} \geq \Lambda$.
4. para qualquer $x \in \mathbb{R}^n$, se $\lambda(x)$ é solução de

$$\sum_{i=0}^p \lambda_i(x) \phi(y^i) = \phi(x)$$

então

$$\sum_{i=0}^p \lambda_i(x) = 1.$$

Demonstração.

1. Como

$$\Lambda \geq \max_{0 \leq i \leq p} \max_{x \in B(Y)} |\ell_i(x)|$$

e $y^i \in B$, então $\max_{x \in B(Y)} |\ell_i(x)| \geq |\ell_i(y^i)|$, logo

$$\Lambda \geq \max_{0 \leq i \leq p} |\ell_i(y^i)|.$$

Pela definição dos polinômios de Lagrange segue que $\max_{0 \leq i \leq p} |\ell_i(y^i)| = 1$, o que mostra o resultado.

2. Seja $A \subset B$. Como Y é Λ -posicionado em B , temos pela Definição 3.6 que para qualquer $x \in B$ existe $\lambda(x) \in \mathbb{R}^{p+1}$ tal que

$$\sum_{i=0}^p \lambda_i(x) \phi(y^i) = \phi(x) \quad \text{com} \quad \|\lambda(x)\|_\infty \leq \Lambda.$$

O mesmo vale, em particular, para todo $x \in A$.

3. O resultado é imediato uma vez que para qualquer $x \in B$ existe $\lambda(x) \in \mathbb{R}^{p+1}$ tal que

$$\sum_{i=0}^p \lambda_i(x) \phi(y^i) = \phi(x) \quad \text{com} \quad \|\lambda(x)\|_\infty \leq \Lambda \leq \tilde{\Lambda}.$$

4. Pelo Lema 3.5, dados $f : \mathbb{R}^n \rightarrow \mathbb{R}$ e um conjunto Y posicionado, o único polinômio interpolador é dado por

$$m(x) = \sum_{i=0}^p f(y^i) \lambda_i(x) \quad \text{e} \quad m(y^i) = f(y^i).$$

Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$, onde $f(x) = 1$ para todo x . Então $f(y^i) = 1$, $i = 0, 1, \dots, p$, e com isso

$$m(x) = \sum_{i=0}^p \lambda_i(x).$$

Por outro lado, como f é um polinômio de grau zero, $m(x) = f(x)$, logo $m(x) = 1$ de onde

$$\sum_{i=0}^p \lambda_i(x) = 1.$$

□

A seguir mostramos alguns exemplos de conjuntos posicionados e suas respectivas constantes de posicionamento. Não é necessário utilizar a função que queremos interpolar para sabermos a constante de posicionamento de determinado conjunto, mas para a melhor visualização as figuras abaixo foram construídas de maneira que interpolassem a função $f(x, y) = \cos(x) + \sin(y)$ numa vizinhança do ponto $(1, 1)$. Nas figuras a seguir a função está traçada em azul enquanto o modelo é mostrado em cinza. Conseguimos ver pelas figuras que quanto mais próximo de 1 a constante Λ , melhor o modelo aproxima a função. Ainda neste capítulo veremos como encontrar a constante de posicionamento para um conjunto dado, e, mais que isso como melhorá-la quando necessário.

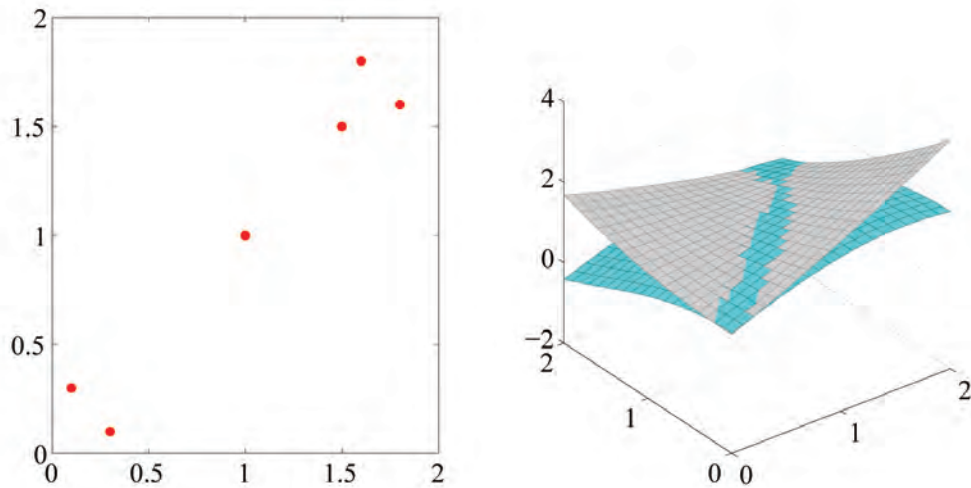


Figura 3.1: $Y = \{(0.1, 0.3); (0.3, 0.1); (1, 1); (1.5, 1.5); (1.6, 1.8); (1.8, 1.6)\}$, com $\Lambda = 186$.

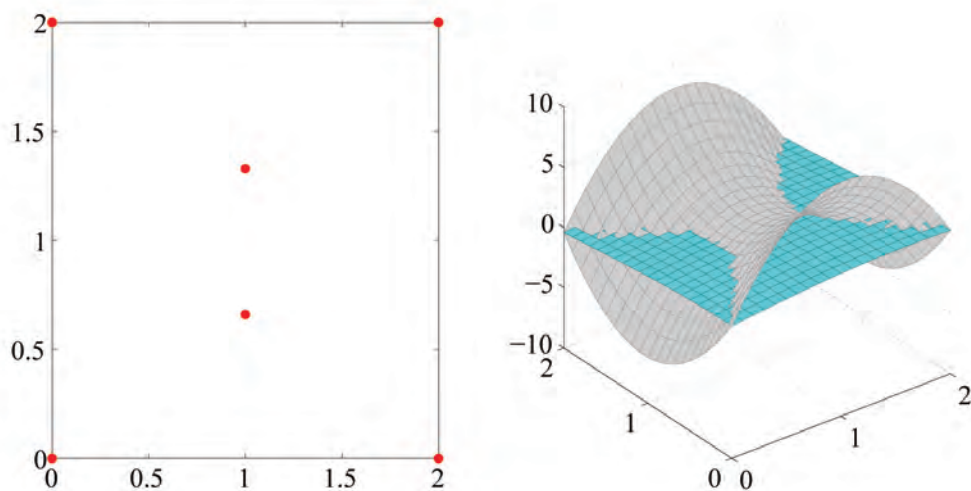


Figura 3.2: $Y = \{(0, 0); (1, 0.66); (1, 1.33); (0, 2); (2, 0); (2, 2)\}$, com $\Lambda = 150$.

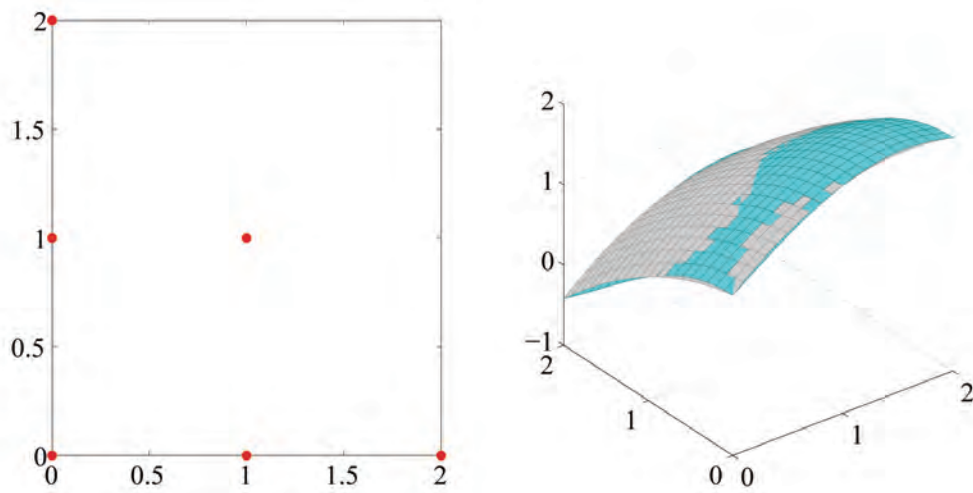


Figura 3.3: $Y = \{(0,0); (0,1); (1,0); (1,1); (0,2); (2,0)\}$, com $\Lambda = 4$.

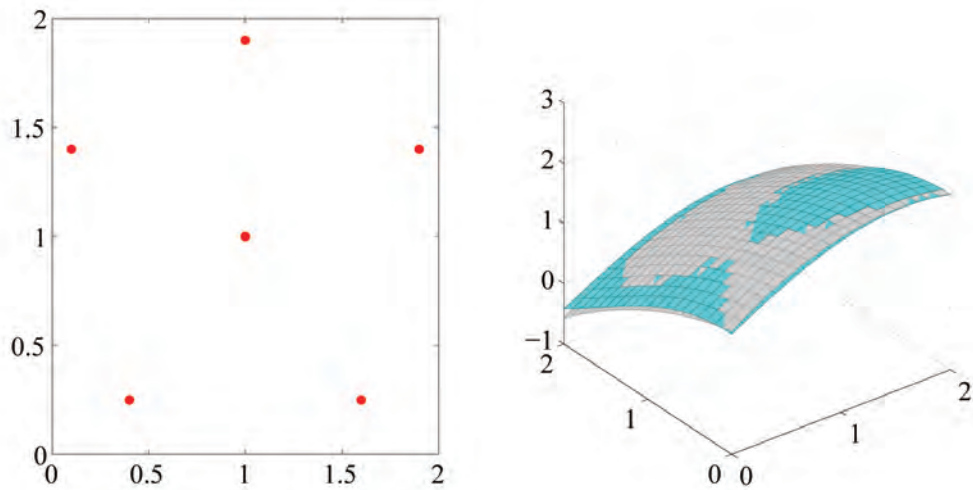


Figura 3.4: $Y = \{(0.1, 1.4); (0.4, 0.25); (1, 1); (1, 1.9); (1.6, 0.25); (1.9, 1.4)\}$, com $\Lambda = 1.82$.

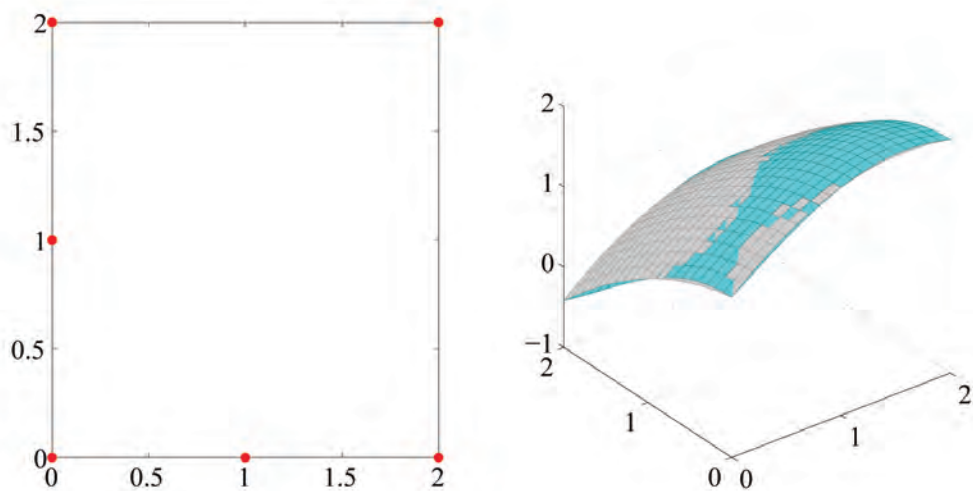


Figura 3.5: $Y = \{(0,0); (0,1); (1,0); (0,2); (2,0); (2,2)\}$, com $\Lambda = 1$.

Mostraremos a seguir que a constante Λ não depende da escala do conjunto de amostra.

Lema 3.8. [5, Lema 3.8] *Sejam $Y = \{y^0, y^1, \dots, y^p\}$ um conjunto de pontos de interpolação e $\lambda_i(x)$, $i = 0, \dots, p$, a base dos polinômios de Lagrange para Y , isto é, solução de (3.1) para Y . Então, para qualquer $\Delta > 0$, $\lambda_i(x/\Delta)$, $i = 0, \dots, p$, é solução de (3.1) para \hat{Y} , onde $\hat{Y} = \{y^0/\Delta, y^1/\Delta, \dots, y^p/\Delta\}$.*

Demonstração. Como a solução $\lambda(x)$ de (3.1) não depende da escolha da base, vamos considerar a base natural $\bar{\phi}$. Temos que $\lambda_i(x)$, $i = 0, \dots, p$, satisfaz

$$\sum_{i=0}^p \lambda_i \bar{\phi}(y^i) = \bar{\phi}(x). \quad (3.2)$$

Se multiplicamos cada y^i e x por $1/\Delta$, isto corresponde a multiplicar as linhas do sistema (3.2) por diferentes escalares, a saber $(1, 1/\Delta, 1/\Delta^2, \dots, 1/\Delta^d)$ os quais dependem do grau do polinômio da base natural que está em cada linha. E temos que $\lambda(x/\Delta)$ satisfaz este novo sistema para \hat{Y} . \square

Segue diretamente do lema anterior o seguinte corolário.

Corolário 3.9. *Se Y é Λ -posicionado em B , então $\hat{Y} = Y/\Delta$ é Λ -posicionado em $\hat{B} = B/\Delta$.*

O próximo resultado mostra que translações do conjunto de interpolação não alteram Λ -posicionamento.

Lema 3.10. [5, Lema 3.9] *Sejam $Y = \{y^0, y^1, \dots, y^p\}$ um conjunto de pontos de interpolação e $\lambda_i(x)$, $i = 0, \dots, p$, a solução de (3.1) para x dado. Então, para qualquer $a \in \mathbb{R}^n$, $\lambda_i(x)$ também é solução de (3.1) para $Y_a = \{y^0 + a, y^1 + a, \dots, y^p + a\}$ e $x_a = x + a$.*

Demonstração. A solução de (3.1) não depende da escolha da base, trabalhando com a base natural com a notação de multiíndice

$$\bar{\phi}_j(x) = \frac{1}{(\alpha_j)!} x^{\alpha_j},$$

temos que

$$\begin{aligned} \bar{\phi}_j(x+a) &= \frac{1}{(\alpha_j)!} (x+a)^{\alpha_j} \\ &= \frac{1}{(\alpha_j)!} (x^{\alpha_j} + x^{\alpha_j-1}a + x^{\alpha_j-2}a^2 + \dots + xa^{\alpha_j-1} + a^{\alpha_j}) \\ &= \bar{\phi}_j(x) + \bar{\phi}_j(a) + \sum_{k: |\alpha_k| < |\alpha_j|} \gamma_k(a) \bar{\phi}_k(x), \end{aligned}$$

onde $\gamma_k(a)$ são coeficientes que dependem de a mas não de x . Então, $\bar{\phi}_j(x+a)$ define uma base em \mathcal{P}_n^d . Novamente usando o fato que a solução de (3.1) não depende da base, temos

$$\sum_{i=0}^p \lambda_i(x) \bar{\phi}(y^i + a) = \bar{\phi}(x + a).$$

□

3.4 Número de Condição como uma medida de bom posicionamento

O número de condição da matriz $M(\phi, Y)$ dada pelas condições de interpolação, em geral, não é uma boa medida de posicionamento de Y , uma vez que podemos ter mudanças grandes quando mudamos ϕ ou mudamos a escala de Y . No entanto, vamos considerar uma específica escolha de base, a base natural $\bar{\phi}$, e vamos olhar o número de condição de $M(\bar{\phi}, \hat{Y})$, onde \hat{Y} é uma versão transladada e com mudança de escala de Y de tal modo que

$$\hat{y} = \{0, \hat{y}^1, \dots, \hat{y}^p\} \subset B(0, 1),$$

e

$$\max_{0 \leq i \leq p} \|\hat{y}^i - \hat{y}^0\| = \max_{0 \leq i \leq p} \|\hat{y}^i\| = 1.$$

Para isso, seja $Y = \{y^0, y^1, \dots, y^p\}$. Primeiro transladamos y^0 para a origem, obtendo

$$\{0, y^1 - y^0, \dots, y^p - y^0\}.$$

Então consideramos

$$\Delta = \Delta(Y) = \max_{0 \leq i \leq p} \|y^i - y^0\|$$

e multiplicamos todos os elementos do conjunto por $1/\Delta$, obtendo

$$\hat{Y} = \{0, \hat{y}^1, \dots, \hat{y}^p\} = \{0, (y^1 - y^0)/\Delta, \dots, (y^p - y^0)/\Delta\} \subset B(0, 1).$$

Para entendermos como o número de condição de $M(\bar{\phi}, \hat{Y})$ pode ser uma medida de posicionamento, precisaremos alguns resultados auxiliares. O primeiro deles mostra que qualquer polinômio pode ser expresso por $\phi(x)$ com coeficientes que não são tão pequenos que o polinômio possa zerar, ou quase, na bola unitária.

Lema 3.11. [5, Lema 3.10] Existe um número $\xi_\infty > 0$ tal que, para qualquer escolha de v satisfazendo $\|v\|_\infty = 1$, existe um ponto $y \in B(0, 1)$ tal que $|v^\top \phi(y)| \geq \xi_\infty$.

Demonstração. Consideremos

$$\psi(v) = \max_{x \in B(0,1)} |v^\top \phi(x)|.$$

Primeiramente, vamos mostrar que ψ é uma norma:

(i) $\psi(v) \geq 0$ para todo $v \in \mathbb{R}^{p+1}$ pois é o máximo de valores absolutos.

(ii) $\psi(v) = 0$ se, e somente se $v = 0$. Temos que,

$$\max_{x \in B(0,1)} |v^\top \phi(x)| = 0 \Rightarrow |v^\top \phi(z)| = 0 \Rightarrow v^\top \phi(z) = 0,$$

para todo $z \in B(0, 1)$. Logo, $v = 0$. Por outro lado, se $v = 0$, temos que para todo $x \in B(0, 1)$

$$v^\top \phi(x) = 0 \Rightarrow |v^\top \phi(x)| = 0 \Rightarrow \max_{x \in B(0,1)} |v^\top \phi(x)| = 0.$$

(iii)

$$\psi(av) = \max_{x \in B(0,1)} |av^\top \phi(x)| = \max_{x \in B(0,1)} |a| |v^\top \phi(x)| = |a| \max_{x \in B(0,1)} |v^\top \phi(x)| = |a| \psi(v).$$

(iv)

$$\begin{aligned} \psi(u+v) &= \max_{x \in B(0,1)} |(u+v)^\top \phi(x)| = \max_{x \in B(0,1)} |u^\top \phi(x) + v^\top \phi(x)| \\ &\leq \max_{x \in B(0,1)} |u^\top \phi(x)| + |v^\top \phi(x)| \leq \max_{x \in B(0,1)} |u^\top \phi(x)| + \max_{x \in B(0,1)} |v^\top \phi(x)| \\ &= \psi(u) + \psi(v) \end{aligned}$$

E como duas normas de dimensão finita são equivalentes, então existe uma constante $\xi_\infty > 0$ tal que

$$\psi(v) \geq \xi_\infty \|v\|_\infty,$$

onde $\xi_\infty = \min_{\|v\|_\infty=1} \psi(v)$. Por fim, quando $\|v\|_\infty = 1$, temos $\psi(v) \geq \xi_\infty$, logo existe $y \in B(0, 1)$ tal que $|v^\top \phi(y)| \geq \xi_\infty$. \square

Se restringimos o resultado anterior para o caso em que ϕ é a base natural e $d \leq 2$, então podemos explicitar um limite inferior para ξ_∞ . É o que mostramos nos próximos

resultados.

Lema 3.12. [5, Lema 3.11] *Seja $v^\top \bar{\phi}(x)$ um polinômio linear, onde $\|v\|_\infty = 1$ e $\bar{\phi}$ é a base natural. Então*

$$\max_{x \in B(0,1)} |v^\top \bar{\phi}(x)| \geq 1.$$

Demonstração. Para o caso linear, temos que $\bar{\phi} = \{1, x_1, \dots, x_n\}$ e $v = [v_0, v_1, \dots, v_n]^\top$. Seja $w = [v_1, v_2, \dots, v_n]^\top$ então o máximo de $|v_0 + v_1 x_1, \dots, v_n x_n|$ em $B(0, 1)$ acontece em $\frac{w}{\|w\|}$ ou $-\frac{w}{\|w\|}$. Quando o maximizador é $\frac{w}{\|w\|}$, temos o valor máximo $v_0 + \|w\|$ e quando o maximizador é $-\frac{w}{\|w\|}$, temos o valor ótimo $-v_0 + \|w\|$. Assim, temos o valor máximo $|v_0| + \|w\| \geq 1$, pois ou $|v_0| = 1$ ou algum $|v_i| = 1$ para $i = 1, \dots, n$, pois $\|v\|_\infty = 1$. \square

Lema 3.13. [5, Lema 3.12] *Seja $v^\top \bar{\phi}(x)$ um polinômio quadrático, onde $\|v\|_\infty = 1$ e $\bar{\phi}$ é a base natural. Então*

$$\max_{x \in B(0,1)} |v^\top \bar{\phi}(x)| \geq \frac{1}{4}.$$

Demonstração. [5, pág. 111] Como $\|v\|_\infty = 1$, pelo menos um dos componentes de v é 1 ou -1 . Então o polinômio $q(x) = v^\top \bar{\phi}(x)$ tem um coeficiente igual a $-1, 1, -\frac{1}{2}$ ou $\frac{1}{2}$. Vamos analisar quando esses coeficientes são positivos, o caso em que são negativos podem ser analisados de modo análogo.

O maior coeficiente em valor absoluto em v corresponde ao termo constante, ou a um termo linear x_i ou ainda a um termo quadrático $x_i^2/2$ ou $x_i x_j$. Vamos restringir como zero todas as variáveis onde isso não ocorre. Vamos mostrar que o máximo valor absoluto do polinômio é pelo menos $\frac{1}{4}$ considerando os 4 casos que correspondem os maiores coeficientes. Em cada caso vamos avaliar o polinômio restrito em alguns pontos da bola unitária e mostrar que em pelo menos um desses pontos do polinômio alcança o valor absoluto de pelo menos $\frac{1}{4}$. O que nos fornece uma garantia de que o máximo valor absoluto do polinômio irrestrito é também limitado inferiormente por $\frac{1}{4}$.

(i) O primeiro caso é quando $q(x) = 1$, em que trivialmente temos $|q(x)| \geq \frac{1}{4}$.

(ii) No segundo caso $q(x) = x_i^2/2 + \alpha x_i + \beta$. Neste caso temos

$$q(1) = \frac{1}{2} + \alpha + \beta, \quad q(-1) = \frac{1}{2} - \alpha + \beta, \quad q(0) = \beta.$$

Se $|q(1)| \geq 1/4$ ou $|q(-1)| \geq 1/4$, conseguimos o resultado. Por outro lado, se $|q(1)| < 1/4$ e $|q(-1)| < 1/4$, temos

$$|q(1) + q(-1)| \leq |q(1)| + |q(-1)| < \frac{1}{2},$$

logo

$$\left| \frac{1}{2} + \alpha + \beta + \frac{1}{2} - \alpha + \beta \right| < \frac{1}{2},$$

que por sua vez implica em $|1 + 2\beta| < 1/2$, logo $\beta < -1/4$. Como $q(0) = \beta$, temos $|q(0)| > 1/4$.

(iii) No terceiro caso $q(x) = \alpha x_i^2/2 + x_i + \beta$. Temos que

$$q(1) = \alpha/2 + 1 + \beta \quad \text{e} \quad q(-1) = \alpha/2 - 1 + \beta.$$

Nesse caso temos que o $\max\{|q(-1)|, |q(1)|\} \geq 1$.

(iv) O quarto caso acontece quando $q(x) = \alpha x_i^2/2 + \beta x_j^2/2 + x_i x_j + \gamma x_i + \delta x_j + \epsilon$. Neste caso vamos considerar quatro pontos na bola $B(0, 1)$:

$$p_1 = (1/\sqrt{2}, 1/\sqrt{2}), p_2 = (1/\sqrt{2}, -1/\sqrt{2}), p_3 = (-1/\sqrt{2}, 1/\sqrt{2}) \text{ e } p_4 = (-1/\sqrt{2}, -1/\sqrt{2}).$$

Assim,

$$q(p_1) = \frac{\alpha}{4} + \frac{\beta}{4} + \frac{1}{2} + \frac{\gamma}{\sqrt{2}} + \frac{\delta}{\sqrt{2}} + \epsilon,$$

$$q(p_2) = \frac{\alpha}{4} + \frac{\beta}{4} - \frac{1}{2} + \frac{\gamma}{\sqrt{2}} - \frac{\delta}{\sqrt{2}} + \epsilon,$$

$$q(p_3) = \frac{\alpha}{4} + \frac{\beta}{4} - \frac{1}{2} - \frac{\gamma}{\sqrt{2}} + \frac{\delta}{\sqrt{2}} + \epsilon,$$

$$q(p_4) = \frac{\alpha}{4} + \frac{\beta}{4} + \frac{1}{2} - \frac{\gamma}{\sqrt{2}} - \frac{\delta}{\sqrt{2}} + \epsilon.$$

E com isso

$$q(p_1) - q(p_2) = 1 + \delta\sqrt{2} \quad \text{e} \quad q(p_3) - q(p_4) = -1 + \delta\sqrt{2}.$$

Se $\delta \geq 0$, então $q(p_1) - q(p_2) \geq 1$, o que implica que se $|q(p_1)| < 1/2$, então $q(p_2) \leq -1/2$. O caso $\delta < 0$ é análogo, ao analisarmos $q(p_3) - q(p_4) \leq -1$.

Considerando os quatro casos, provamos o lema. □

Podemos ainda fazer esta estimativa em termos da norma euclidiana, uma vez que $\|u\|_\infty \leq \|u\| \leq \sqrt{n}\|u\|_\infty$, onde n é a dimensão de u . Então, dado \bar{v} tal que $\|\bar{v}\| = 1$ podemos multiplicar \bar{v} por no máximo $\sqrt{p+1}$ para que $v = \alpha\bar{v}$, $0 < \alpha < \sqrt{p+1}$, tal que $\|v\|_\infty = 1$. Então,

$$\xi_2 = \min_{\|\bar{v}\|=1} \max_{x \in B(0,1)} |\bar{v}^\top \bar{\phi}(x)| \geq \frac{1}{\sqrt{p+1}} \min_{\|v\|_\infty=1} \max_{x \in B(0,1)} |v^\top \bar{\phi}(x)| \geq \frac{1}{\sqrt{p+1}} \xi_\infty.$$

3.5 Λ -posicionamento e o número de condição

Vamos agora relacionar o número de condição da matriz $M(\bar{\phi}, \hat{Y})$ com a constante de posicionamento Λ do conjunto \hat{Y} . Da definição de Λ -posicionamento em $B(y^0, \Delta(Y))$, usando a base natural, temos que para todo $x \in B(y^0, \Delta(Y))$, existe $\lambda(x)$ tal que

$$M^\top \lambda(x) = \bar{\phi}(x) \quad \text{com} \quad \|\lambda(x)\|_\infty \leq \Lambda.$$

Usando os Lemas 3.8 e 3.10, temos que isso é equivalente a dizer que para todo $x \in B(0, 1)$ existe $\lambda(x)$ tal que

$$\hat{M}^\top \lambda(x) = \bar{\phi}(x) \quad \text{com} \quad \|\lambda(x)\|_\infty \leq \Lambda.$$

O número de condição da matriz $\hat{M} = M(\bar{\phi}, \hat{Y})$ é dado por $\text{cond}(\hat{M}) = \|\hat{M}\| \|\hat{M}^{-1}\|$. Com o auxílio do próximo resultado vamos mostrar que para relacionar $\text{cond}(\hat{M})$ com a constante de posicionamento Λ é suficiente encontrar um limitante para $\|\hat{M}^{-1}\|$.

Lema 3.14. *Seja $A \in \mathbb{R}^{m \times n}$, então*

$$\max_{i,j} |a_{ij}| \leq \|A\| \leq n\sqrt{m} \max_{i,j} |a_{ij}|,$$

onde, $\|A\| = \max_{\|x\|=1} \|Ax\|$ para $x \in \mathbb{R}^n$.

Demonstração. Temos que

$$(Ax)_i = \sum_{j=1}^n a_{ij} x_j,$$

como para $x \in \mathbb{R}^n$

$$\|A\| = \max_{\|x\|=1} \|Ax\|,$$

temos que

$$\|A\| = \max_{\|x\|=1} \left(\sum_{i=1}^m \left(\sum_{j=1}^n a_{ij} x_j \right)^2 \right)^{\frac{1}{2}} = \left(\sum_{i=1}^m \left(\sum_{j=1}^n a_{ij} \bar{x}_j \right)^2 \right)^{\frac{1}{2}},$$

para algum $\bar{x} \in \mathbb{R}^n$, com $\|\bar{x}\| = 1$.

Seja $\bar{x}_{\max} = \max_{1 \leq i \leq n} \bar{x}_i$. Como $\|\bar{x}\| = 1$ temos que $|\bar{x}_{\max}| \leq 1$ e com isso

$$\|A\| = \left(\sum_{i=1}^m \left(\sum_{j=1}^n a_{ij} \bar{x}_j \right)^2 \right)^{\frac{1}{2}} \leq \left(\sum_{i=1}^m \left(\sum_{j=1}^n a_{ij} \bar{x}_{\max} \right)^2 \right)^{\frac{1}{2}}.$$

Então,

$$\begin{aligned}
\|A\| &\leq |\bar{x}_{\max}| \left(\sum_{i=1}^m \left(\sum_{j=1}^n a_{ij} \right)^2 \right)^{\frac{1}{2}} \leq \left(\sum_{i=1}^m \left(\sum_{j=1}^n a_{ij} \right)^2 \right)^{\frac{1}{2}} \\
&\leq \left(\sum_{i=1}^m \left(\sum_{j=1}^n \max_j |a_{ij}| \right)^2 \right)^{\frac{1}{2}} \leq \left(\sum_{i=1}^m n^2 (\max_j |a_{ij}|)^2 \right)^{\frac{1}{2}} \\
&\leq (mn^2 (\max_{i,j} |a_{ij}|)^2)^{\frac{1}{2}} = n\sqrt{m} \max_{i,j} |a_{ij}|.
\end{aligned}$$

Por outro lado, seja \bar{x}^* um vetor com coordenada 1 na posição j_0 e zero nas demais, de modo que $\max_{i,j} |a_{ij}| = |a_{i_0 j_0}|$, logo $\|A\bar{x}^*\| \leq \max_{\|x\|=1} \|Ax\| = \|A\|$. Assim,

$$\max_{i,j} |a_{ij}| = \max_i |a_{i j_0}| \leq \left(\sum_{i=1}^m |a_{i j_0}|^2 \right)^{\frac{1}{2}} = \|A\bar{x}^*\| \leq \|A\|.$$

□

Com isso, conseguimos um limitante para a norma de \hat{M} , uma vez que $\max_{i,j} |\hat{m}_{ij}| = 1$, temos

$$1 \leq \|\hat{M}\| \leq (p+1)^{\frac{3}{2}}.$$

Para o próximo resultado auxiliar lembramos que uma matriz $A \in \mathbb{R}^{m \times n}$, sem perda de generalidade $m \geq n$, pode ser decomposta em um produto $U\Sigma V^\top$, onde $U \in \mathbb{R}^{m \times m}$ e $V \in \mathbb{R}^{n \times n}$ são matrizes ortogonais, $\Sigma \in \mathbb{R}^{m \times n}$ é uma matriz diagonal cujos elementos diagonais σ_i são as raízes quadradas positivas dos autovalores da matriz $A^\top A$, ou seja, os valores singulares de A dispostos em ordem não decrescente. Comparando as j -ésimas colunas de cada lado da equação

$$AV = U\Sigma$$

obtemos

$$Av_j = \sigma_j u_j \quad j = 1, \dots, n.$$

Analogamente,

$$A^\top U = V\Sigma^\top$$

e, portanto,

$$\begin{aligned}
A^\top u_j &= \sigma_j v_j & \text{para } j &= 1, \dots, n \\
A^\top u_j &= 0 & \text{para } j &= n+1, \dots, m.
\end{aligned}$$

Os vetores v_j são chamados de vetores singulares à direita de A e os u_j de vetores singulares

à esquerda de A . [11]

Lema 3.15. [5, Lema 3.13] *Seja w um vetor singular à direita normalizado de uma matriz quadrada não-singular $A \in \mathbb{R}^{n \times n}$. Então, para qualquer vetor $r \in \mathbb{R}^n$,*

$$\|Ar\| \geq |w^\top r| \|A\|.$$

Demonstração. Temos que $\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|} \leq \frac{\|Ax\|}{\|x\|}$, para todo $x \in \mathbb{R}^n$ com $x \neq 0$. Como por hipótese w é um vetor singular à direita, existe σ um valor singular de A associado a um autovetor unitário u de AA^\top tal que $Aw = \sigma u$ e $\sigma w^\top = u^\top A$. Assim

$$\begin{aligned} |w^\top r| \|A\| &\leq |w^\top r| \|Aw\| \\ &= |w^\top r| \|\sigma u\| \\ &= |\sigma w^\top r| \|u\| \\ &= |\sigma w^\top r| \\ &= |u^\top Ar| \\ &\leq \|u\| \|Ar\| = \|Ar\|, \end{aligned}$$

completando a demonstração. □

Com esses resultados, agora conseguimos enunciar o teorema que relaciona o número de condição de \hat{M} e a constante de posicionamento Λ .

Teorema 3.16. [5, Teorema 3.14] *Se \hat{M} é não singular e $\|\hat{M}^{-1}\| \leq \Lambda$, então \hat{Y} é $\Lambda\sqrt{p+1}$ -posicionado em $B(0,1)$. Reciprocamente, se \hat{Y} é Λ -posicionado em $B(0,1)$, então $\|\hat{M}^{-1}\| \leq \theta\Lambda\sqrt{p+1}$, onde $\theta > 0$ depende de n e d , mas não depende de \hat{Y} ou Λ .*

Demonstração. (\Rightarrow) Se \hat{M} é não singular e $\|\hat{M}^{-1}\| \leq \Lambda$, então, como $\hat{M}^\top \lambda(x) = \bar{\phi}(x)$, temos

$$\|\lambda(x)\|_\infty \leq \|\hat{M}^{-\top}\|_\infty \|\bar{\phi}(x)\|_\infty \leq (p+1)^{\frac{1}{2}} \|\hat{M}^{-\top}\| \|\bar{\phi}(x)\|_\infty.$$

Como $\max_{x \in B(0,1)} \bar{\phi}(x) \leq 1$, temos

$$\|\lambda(x)\|_\infty \leq (p+1)^{\frac{1}{2}} \|\hat{M}^{-\top}\| \leq (p+1)^{\frac{1}{2}} \Lambda.$$

(\Leftarrow) Seja \bar{v} um vetor singular à direita normalizado correspondente ao maior valor singular de $\hat{M}^{-\top}$. Como $\|\bar{v}\| = 1$, pelo Lema 3.11 existe $y \in B(0,1)$ tal que

$$|\bar{v}^\top \bar{\phi}(y)| \geq \xi_2 \geq \frac{\xi_\infty}{\sqrt{p+1}}.$$

Temos também por (3.1) que $\|\lambda(x)\| = \|\hat{M}^{-\top} \bar{\phi}(x)\|$, usando o Lema 3.15 com $A = \hat{M}^{-\top}$, $w = \bar{v}$ e $r = \bar{\phi}(y)$,

$$\|\lambda(x)\| = \|\hat{M}^{-\top} \bar{\phi}(y)\| \geq |\bar{v}^\top \bar{\phi}(y)| \|\hat{M}^{-\top}\|,$$

e conseqüentemente

$$|\bar{v}^\top \bar{\phi}(y)| \|\hat{M}^{-\top}\| \leq \|\lambda(x)\| \leq \sqrt{p+1} \|\lambda(x)\|_\infty.$$

Portanto,

$$\|\hat{M}^{-\top}\| \leq \frac{\sqrt{p+1}}{|\bar{v}^\top \bar{\phi}(y)|} \|\lambda(x)\|_\infty \leq \theta \Lambda \sqrt{p+1},$$

com $\theta = \frac{\sqrt{p+1}}{\xi_\infty}$, uma vez que $\|\lambda(x)\|_\infty \leq \Lambda$. □

Em vista dos Lemas 3.12 e 3.13 podemos estimar o valor de θ . Abaixo, apresentamos um corolário que será útil para provar que conseguimos construir conjuntos Λ -posicionados em um número finito de passos.

Corolário 3.17. [5, Corolário 3.15] *Se \hat{Y} é Λ -posicionado em $B(0, 1)$, então o volume $\text{vol}(\bar{\phi}(\hat{Y})) \geq \Theta(p, \theta) > 0$, onde $\Theta(p, \theta)$ depende somente de p , (ou seja, de n e d) e da base $\bar{\phi}$.*

Demonstração. O Teorema 3.16 garante a existência da constante θ que depende somente de n , d e $\bar{\phi}$ tal que $\|\hat{M}^{-\top}\| \leq \theta \Lambda \sqrt{p+1}$. Como o valor absoluto do determinante de uma matriz quadrada é o produto de seus valores singulares, temos que

$$\text{vol}(\bar{\phi}(\hat{Y})) = \frac{1}{(p+1)! |\det(\hat{M}^{-1})|} \geq \frac{1}{(p+1)! (\theta \Lambda \sqrt{p+1})^{p+1}}.$$

□

3.6 Limitante para o erro em termos do número de condição

Vamos agora encontrar um limitante para o erro entre o polinômio interpolador e a função interpolada, entre seus gradientes e entre suas hessianas.

Seja $Y = \{y^0, y^1, \dots, y^p\} \subset \mathbb{R}^n$, onde $p = \frac{(n+1)(n+2)}{2}$ um conjunto posicionado para interpolação polinomial contido em $B(y^0, \Delta(Y))$. Vamos supor que f é interpolada por

um polinômio m , em termos da base natural, ou seja,

$$m(y) = \sum_{k=0}^p \gamma_k \bar{\phi}_k(y).$$

Além disso vamos assumir como hipótese que f é duas vezes continuamente diferenciável em um domínio Ω aberto contendo $B(y^0, \Delta(Y))$ e $\nabla^2 f$ é Lipschitz contínua em Ω , com constante $\nu_2 > 0$. Com isso, podemos escrever o modelo quadrático de interpolação como

$$m(y) = c + g^\top y + \frac{1}{2} y^\top H y = c + \sum_{1 \leq k \leq n} g_k y_k + \frac{1}{2} \sum_{1 \leq k, l \leq n} h_{kl} y_k y_l, \quad (3.3)$$

onde H é simétrica de ordem n . Os coeficientes c, g_1, \dots, g_n e h_{kl} , $1 \leq l \leq k \leq n$ são unicamente determinados pelas condições de interpolação, uma vez que o conjunto Y é posicionado.

Vamos assumir sem perda de generalidade, por um momento, que $y^0 = 0$. Considerando agora um ponto y na bola $B(0, \Delta(Y))$ para o qual vamos estimar o erro no valor da função,

$$m(y) = f(y) + e^f(y), \quad (3.4)$$

do gradiente

$$\nabla m(y) = H y + g = \nabla f(y) + e^g(y), \quad (3.5)$$

e também da Hessiana

$$H = \nabla^2 f(y) + E^H(y).$$

Como as Hessianas de f e m são simétricas, vamos apenas considerar o erro de derivadas de segunda ordem nos elementos da diagonal e abaixo delas, ou seja,

$$h_{kl} = \nabla_{kl}^2 f(y) + E_{kl}^H(y), \quad 1 \leq l \leq k \leq n.$$

Das condições de interpolação,

$$m(y^i) = \sum_{k=0}^p \gamma_k \bar{\phi}_k(y^i) = f(y^i), \quad i = 0, \dots, p.$$

Subtraindo (3.4) de cada uma das $(p+1)$ equações das condições de interpolação, temos

$$m(y^i) - m(y) = f(y^i) - f(y) - e^f(y).$$

Usando (3.3), segue

$$c + g^\top y^i + \frac{1}{2}(y^i)^\top Hy^i - c - g^\top y - \frac{1}{2}y^\top Hy = f(y^i) - f(y) - e^f(y),$$

o que simplificando resulta

$$g^\top (y^i - y) + \frac{1}{2}((y^i)^\top Hy^i - y^\top Hy) = f(y^i) - f(y) - e^f(y). \quad (3.6)$$

Temos que

$$(y^i)^\top Hy^i = \sum_{k=1}^n \sum_{l=1}^n h_{kl} y_k^i y_l^i \quad \text{e} \quad y^\top Hy = \sum_{k=1}^n \sum_{l=1}^n h_{kl} y_k y_l,$$

assim

$$(y^i)^\top Hy^i - y^\top Hy = \sum_{k=1}^n \sum_{l=1}^n h_{kl} (y_k^i y_l^i - y_k y_l),$$

por outro lado,

$$\begin{aligned} y_k^i y_l^i - y_k y_l &= y_k^i y_l^i - y_k^i y_l - y_k y_l^i + y_k y_l + y_k^i y_l + y_k y_l^i - 2y_k y_l \\ &= (y_k^i - y_k)(y_l^i - y_l) + (y_k^i - y_k)y_l + (y_l^i - y_l)y_k, \end{aligned}$$

então

$$\begin{aligned} (y^i)^\top Hy^i - y^\top Hy &= \sum_{k=1}^n \sum_{l=1}^n (h_{kl} (y_k^i - y_k)(y_l^i - y_l) + h_{kl} (y_k^i - y_k)y_l + h_{kl} (y_l^i - y_l)y_k) \\ &= (y^i - y)^\top H(y^i - y) + (y^i - y)^\top Hy + y^\top H(y^i - y) \\ &= (y^i - y)^\top H(y^i - y) + 2(y^i - y)^\top Hy. \end{aligned}$$

Substituindo em (3.6),

$$g^\top (y^i - y) + \frac{1}{2}(y^i - y)^\top H(y^i - y) + (y^i - y)^\top Hy = f(y^i) - f(y) - e^f(y).$$

Pela aproximação de Taylor de segunda ordem de f em torno do ponto y^i temos

$$f(y^i) = f(y) + (y^i - y)^\top \nabla f(y) + \frac{1}{2}(y^i - y)^\top \nabla^2 f(y)(y^i - y) + \mathcal{O}(\Delta^3).$$

Usando (3.5),

$$\begin{aligned} (y^i - y)^\top (\nabla f(y) + e^g(y) - Hy) + \frac{1}{2}(y^i - y)^\top H(y^i - y) + (y^i - y)^\top Hy \\ = (y^i - y)^\top \nabla f(y) + \frac{1}{2}(y^i - y)^\top \nabla^2 f(y)(y^i - y) + \mathcal{O}(\Delta^3) - e^f(y), \end{aligned}$$

então

$$(y^i - y)^\top e^g(y) + \frac{1}{2}(y^i - y)^\top [H - \nabla^2 f(y)](y^i - y) = \mathcal{O}(\Delta^3) - e^f(y) \quad (3.7)$$

Como $y^0 = 0$,

$$-y^\top e^g(y) + \frac{1}{2}y^\top [H - \nabla^2 f(y)]y = \mathcal{O}(\Delta^3) - e^f(y),$$

subtraindo de (3.7)

$$y^{i\top} e^g(y) + \frac{1}{2}y^{i\top} [H - \nabla^2 f(y)]y^i - y^{i\top} E^H(y)y = \mathcal{O}(\Delta^3),$$

ou ainda,

$$y^{i\top} (e^g(y) - E^H(y)y) + \frac{1}{2}y^{i\top} [H - \nabla^2 f(y)]y^i = \mathcal{O}(\Delta^3).$$

Este é um sistema linear que pode ser escrito como

$$\sum_{1 \leq k \leq n} y_k^i t_k(y) + \frac{1}{2} \sum_{1 \leq k \leq n} (y_k^i)^2 E_{kk}^H(y) + \sum_{1 \leq l < k \leq n} (y_k^i y_l^i) E_{kj}^H(y) = \mathcal{O}(\Delta^3),$$

onde $i = 1, \dots, p$ e

$$t(y) = e^g(y) - E^H(y)y = e^g(y) - [H - \nabla^2 f(y)]y. \quad (3.8)$$

Podemos escrevê-lo matricialmente como

$$Q \begin{bmatrix} t(y) \\ e^H(y) \end{bmatrix} = \mathcal{O}(\Delta^3), \quad (3.9)$$

onde $e^H(y)$ é o vetor de dimensão $n + n(n+1)/2$ que são os elementos de $E_{kk}^H(y)$, $k = 1, \dots, n$ e $E_{kl}^H(y)$, $1 \leq l < k \leq n$. A matriz Q é a matriz $M(\bar{\phi}, Y)$, quando excluímos a primeira linha e a primeira coluna e $d = 2$ e $y^0 = 0$, desde que $e^H(y)$ esteja devidamente ordenado.

O próximo resultado nos auxilia a encontrar um limitante para cada entrada do vetor dado pelo lado direito de (3.9).

Lema 3.18. [6, Lema 4.1.14] *Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função duas vezes continuamente diferenciável em um aberto convexo $D \subset \mathbb{R}^n$ com $\nabla^2 f$ Lipschitz contínua em D com constante γ . Então para todo $x + p \in D$,*

$$|f(x + p) - (f(x) + p^\top \nabla f(x) + \frac{1}{2}p^\top \nabla^2 f(x)p)| \leq \frac{\gamma}{6} \|p\|^3.$$

Demonstração. Seja $h : \mathbb{R} \rightarrow \mathbb{R}$ uma parametrização de f ao longo do segmento $[x, x + p]$,

com $h(t) = f(x + tp)$. Seja $r(t) = x + tp$. Então pela Regra da Cadeia, para $0 \leq \theta \leq 1$,

$$\frac{dh}{dt}(\theta) = \sum_{i=1}^n \frac{\partial f}{\partial r_i(t)}(r(\theta)) \frac{dr_i(t)}{dt}(\theta) = \nabla f(x + \theta p)^\top p,$$

e

$$\frac{d^2h}{dt^2}(\theta) = \sum_{i=1}^n \sum_{j=1}^n \frac{\partial^2 f}{\partial r_i(t) \partial r_j(t)}(r(\theta)) \left(\frac{dr_i(t)}{dt}(\theta) \right)^2 + \sum_{i=1}^n \frac{\partial f}{\partial r_i(t)}(r(\theta)) \frac{d^2r_i(t)}{dt^2} = p^\top \nabla^2 f(x + \theta p) p.$$

Pelo Teorema Fundamental do Cálculo, dados $a, b \in \mathbb{R}$,

$$\int_a^{a+b} h'(t) dt = h(a+b) - h(a).$$

Por outro lado, através de integração por partes, vemos que

$$\int_a^{a+b} h'(t) dt = (a+b)h'(a+b) - ah'(a) - \int_a^{a+b} th''(t) dt,$$

Com isso e novamente o Teorema Fundamental do Cálculo,

$$\begin{aligned} h(a+b) - h(a) &= (a+b)h'(a+b) - ah'(a) - \int_a^{a+b} th''(t) dt \\ &= \int_a^{a+b} (a+b)h''(t) dt + (a+b)h'(a) - ah'(a) - \int_a^{a+b} th''(t) dt \\ &= bh'(a) + \int_a^{a+b} (a+b-t)h''(t) dt. \end{aligned}$$

Fazendo a mudança de variáveis $t = a + sb$,

$$h(a+b) - h(a) - bh'(a) = \int_0^1 (b-sb)h''(a+sb)b ds = \int_0^1 b^2(1-s)h''(a+sb) ds.$$

Por outro lado,

$$\frac{1}{2}b^2h''(a) = \int_0^1 b^2(1-s)h''(a) ds.$$

Portanto,

$$h(a+b) - h(a) - bh'(a) - \frac{1}{2}b^2h''(a) = \int_0^1 b^2(1-s)[h''(a+sb) - h''(a)] ds.$$

Para $a = 0$ e $b = 1$,

$$h(1) - h(0) - h'(0) - \frac{1}{2}h''(0) = \int_0^1 (1-s)[h''(s) - h''(0)] ds.$$

Substituindo $h(t) = f(x+tp)$, $h'(t) = \nabla f(x+tp)^\top p$ e $h''(t) = p^\top \nabla^2 f(x+tp)p$ na equação acima

$$f(x+p) - f(x) - \nabla f(x)^\top p - \frac{1}{2}p^\top \nabla^2 f(x)p = \int_0^1 (1-s)[p^\top \nabla^2 f(x+sp)p - p^\top \nabla^2 f(x)p] ds.$$

Com isso,

$$\left| f(x+p) - f(x) - p^\top \nabla f(x) - \frac{1}{2}p^\top \nabla^2 f(x)p \right| \leq \int_0^1 |(1-s)p^\top [\nabla^2 f(x+sp) - \nabla^2 f(x)]p| ds$$

$$\left| f(x+p) - f(x) - p^\top \nabla f(x) - \frac{1}{2}p^\top \nabla^2 f(x)p \right| \leq \gamma \|p\|^3 \int_0^1 (s-s^2) ds = \frac{\gamma}{6} \|p\|^3,$$

o que completa a demonstração. \square

O sistema (3.7) pode ser escrito como

$$(y^i - y)^\top e^g(y) + \frac{1}{2}(y^i - y)^\top [H - \nabla^2 f(y)](y^i - y) + e^f(y) = \mathcal{O}(\Delta^3).$$

Substituindo $e^f(y)$ e $e^g(y)$ no lado esquerdo obtemos

$$(y^i - y)^\top (\nabla m(y) - \nabla f(y)) + \frac{1}{2}(y^i - y)^\top H(y^i - y) - \frac{1}{2}(y^i - y)^\top \nabla^2 f(y)(y^i - y) + m(y) - f(y).$$

Como $m(y^i) = f(y^i) = f(y + y^i - y)$

$$\begin{aligned} & f(y + y^i - y) - f(y) - (y^i - y)^\top \nabla f(y) - \frac{1}{2}(y^i - y)^\top \nabla^2 f(y)(y^i - y) \\ & -m(y^i) + m(y) + (y^i - y)^\top \nabla m(y) + \frac{1}{2}(y^i - y)^\top H(y^i - y). \end{aligned}$$

No entanto,

$$\begin{aligned} & -m(y^i) + m(y) + (y^i - y)^\top \nabla m(y) + \frac{1}{2}(y^i - y)^\top H(y^i - y) \\ & = -m(y^i) + m(y) + (y^i)^\top \nabla m(y) - y^\top \nabla m(y) + \frac{1}{2}(y^i)^\top H y^i - (y^i)^\top H y + \frac{1}{2}y^\top H y. \end{aligned}$$

Como $\nabla m(y) = Hy + g$,

$$\begin{aligned} & -m(y^i) + m(y) + (y^i)^\top H y + (y^i)^\top g - y^\top H y - y^\top g + \frac{1}{2}(y^i)^\top H y^i - (y^i)^\top H y + \frac{1}{2}y^\top H y \\ & = -m(y^i) + m(y) + (y^i)^\top g - y^\top g + \frac{1}{2}(y^i)^\top H y^i - \frac{1}{2}y^\top H y. \end{aligned}$$

Como $m(y + y) = c + 2y^\top g + 2y^\top Hy$, subtraindo e somando da equação acima

$$-m(y + y) + c + y^\top g - m(y^i) + m(y) + (y^i)^\top g + \frac{1}{2}(y^i)^\top Hy^i + \frac{3}{2}y^\top Hy$$

Como $m(y^i) = c + (y^i)^\top g + \frac{1}{2}(y^i)^\top Hy^i$, substituindo acima

$$-m(y + y) + y^\top Hy + y^\top g + m(y) + \frac{1}{2}y^\top Hy = -m(y + y) + m(y) + y^\top \nabla m(y) + \frac{1}{2}y^\top Hy$$

Então (3.7) pode ser escrito como

$$\begin{aligned} f(y + y^i - y) - f(y) - (y^i - y)^\top \nabla f(y) - \frac{1}{2}(y^i - y)^\top \nabla^2 f(y)(y^i - y) \\ -m(y + y) + m(y) + y^\top \nabla m(y) + \frac{1}{2}y^\top Hy = \mathcal{O}(\Delta^3) \end{aligned} \quad (3.10)$$

Usando o Lema (3.18) temos que o lado esquerdo é a diferença entre dois termos que podem ser limitados por $\nu_2 \|y^i - y\|^3/6$ e $\nu_2 \|y\|^3/6$ respectivamente. Como $\|y^i - y\| \leq 2\Delta$ e $\|y\| \leq \Delta$, o lado esquerdo de (3.10) é limitado por $3\Delta^3/2$.

Então,

$$\left\| Q \begin{bmatrix} t(y) \\ e^H(y) \end{bmatrix} \right\| \leq \frac{3}{2}(p+1)^{\frac{1}{2}} \nu_2 \Delta^3. \quad (3.11)$$

Para eliminar a dependência de Q em Λ vamos considerar a matriz

$$\hat{Q} = Q \begin{bmatrix} D_\Delta^{-1} & 0 \\ 0 & D_{\Delta^2}^{-1} \end{bmatrix}, \quad (3.12)$$

onde D_Δ é uma matriz diagonal de ordem n com Δ nas entradas e D_{Δ^2} é uma matriz diagonal de ordem $(p - n)$ com Δ^2 na entradas. A matriz \hat{Q} é análoga a Q para o caso $\hat{Y} = Y/\Delta \subset B(0, 1)$.

O próximo teorema nos garante limitantes para o erro entre os valores da função e do modelo quadrático, no sentido da definição de Plenamente Quadrático.

Teorema 3.19. [5, Teorema 3.16] *Seja $Y = \{y^0, y^1, \dots, y^p\} \subset \mathbb{R}^n$, onde $p = \frac{(n+1)(n+2)}{2}$ um conjunto posicionado para interpolação polinomial contido em $B(y^0, \Delta(Y))$. Se f é duas vezes continuamente diferenciável em um domínio Ω aberto contendo $B(y^0, \Delta(Y))$, $\nabla^2 f$ é Lipschitz contínua em Ω , com constante $\nu_2 > 0$ e f é interpolada por um polinômio m . Então, para todo ponto $y \in B(y^0, \Delta(Y))$, temos que*

- o erro entre a Hessiana do modelo e a Hessiana da função satisfaz

$$\|\nabla^2 f(y) - \nabla^2 m(y)\| \leq \kappa_{eh}\Delta,$$

- o erro entre o gradiente do modelo e o gradiente da função satisfaz

$$\|\nabla f(y) - \nabla m(y)\| \leq \kappa_{eg}\Delta^2,$$

- o erro entre o modelo e a função satisfaz

$$|f(y) - m(y)| \leq \kappa_{ef}\Delta^3,$$

onde κ_{eh} , κ_{eg} e κ_{ef} são constantes dadas por

$$\begin{aligned}\kappa_{eh} &= \frac{3\sqrt{2}}{2}(p+1)^{\frac{1}{2}}\nu_2\|\hat{Q}^{-1}\|, \\ \kappa_{eg} &= \frac{3(1+\sqrt{2})}{2}(p+1)^{\frac{1}{2}}\nu_2\|\hat{Q}^{-1}\|, \\ \kappa_{ef} &= \frac{(6+9\sqrt{2})}{4}(p+1)^{\frac{1}{2}}\nu_2\|\hat{Q}^{-1}\| + \frac{\nu_2}{6}.\end{aligned}$$

Demonstração. O lado esquerdo de (3.9) pode ser escrito como

$$Q \begin{bmatrix} D_{\Delta}^{-1} & 0 \\ 0 & D_{\Delta^2}^{-1} \end{bmatrix} \begin{bmatrix} D_{\Delta}t(y) \\ D_{\Delta^2}e^H(y) \end{bmatrix}.$$

Usando (3.11) e (3.12), temos que

$$\left\| \begin{bmatrix} D_{\Delta}t(y) \\ D_{\Delta^2}e^H(y) \end{bmatrix} \right\| \leq \frac{3}{2}(p+1)^{\frac{1}{2}}\nu_2\|\hat{Q}^{-1}\|\Delta^3. \quad (3.13)$$

Então

$$\|D_{\Delta^2}e^H(y)\| \leq \frac{3}{2}(p+1)^{\frac{1}{2}}\nu_2\|\hat{Q}^{-1}\|\Delta^3,$$

o que implica

$$\|e^H(y)\| \leq \frac{3}{2}(p+1)^{\frac{1}{2}}\nu_2\|\hat{Q}^{-1}\|\Delta.$$

O erro das Hessianas é, portanto, dado por

$$\|E^H(y)\| \leq \|E^H(y)\|_F \leq \sqrt{2}\|e^H(y)\| \leq \frac{3\sqrt{2}}{2}(p+1)^{\frac{1}{2}}\nu_2\|\hat{Q}^{-1}\|\Delta,$$

onde $\|\cdot\|_F$ denota a norma de Frobenius.

De (3.13)

$$\|D_{\Delta}t(y)\| \leq \frac{3}{2}(p+1)^{\frac{1}{2}}\nu_2\|\hat{Q}^{-1}\|\Delta^3,$$

e conseqüentemente

$$\|t(y)\| \leq \frac{3}{2}(p+1)^{\frac{1}{2}}\nu_2\|\hat{Q}^{-1}\|\Delta^2.$$

Por outro lado, por (3.8)

$$\begin{aligned} \|e^g(y)\| &\leq \|t(y)\| + \|E^H(y)\|\|y\| \\ &\leq \frac{3}{2}(p+1)^{\frac{1}{2}}\nu_2\|\hat{Q}^{-1}\|\Delta^2 + \left(\frac{3\sqrt{2}}{2}(p+1)^{\frac{1}{2}}\nu_2\|\hat{Q}^{-1}\|\Delta\right)\Delta \\ &= \frac{3(1+\sqrt{2})}{2}(p+1)^{\frac{1}{2}}\nu_2\|\hat{Q}^{-1}\|\Delta^2. \end{aligned}$$

Finalmente, da versão de (3.7) para $i = 0$, temos

$$\begin{aligned} |e^f(y)| &\leq \|e^g(y)\|\Delta + \frac{1}{2}\|E^H(y)\|\Delta^2 + \frac{\nu_2}{6}\Delta^3 \\ &\leq \frac{3(1+\sqrt{2})}{2}(p+1)^{\frac{1}{2}}\nu_2\|\hat{Q}^{-1}\|\Delta^3 + \frac{3\sqrt{2}}{2}(p+1)^{\frac{1}{2}}\nu_2\|\hat{Q}^{-1}\|\Delta^3 + \frac{\nu_2}{6}\Delta^3 \\ &= \frac{(6+9\sqrt{2})}{4}(p+1)^{\frac{1}{2}}\nu_2\|\hat{Q}^{-1}\|\Delta^3 + \frac{\nu_2}{6}\Delta^3. \end{aligned}$$

Se $y \neq 0$, fazemos uma mudança de variável $\hat{f}(z) = f(z + y^0)$, com $z = y - y^0$. Aplicamos então o resultado provado e encontramos limitantes para $\|\nabla^2\hat{f}(z) - \nabla^2\hat{m}(z)\|$, $\|\nabla\hat{f}(z) - \nabla\hat{m}(z)\|$ e $|\hat{f}(z) - \hat{m}(z)|$. Definimos $m(y) = \hat{m}(y^0 + z)$ e mudamos a função e o polinômio modelo novamente para as variáveis originais. E com isso encontramos os limitantes desejados, uma vez que $\nabla^2\hat{m}(z) = \nabla^2m(y)$, $\nabla^2\hat{f}(z) = \nabla^2f(y)$, $\nabla\hat{m}(z) = \nabla m(y)$ e $\nabla\hat{f}(z) = \nabla f(y)$. \square

3.7 Algoritmos para bom posicionamento

Até o momento, apresentamos resultados teóricos que garantem que interpolação polinomial é uma boa alternativa para construir modelos quadráticos de uma função a ser minimizada. Agora vamos mostrar que conseguimos algoritmos para obter a base de Lagrange de um conjunto de pontos, melhorar o posicionamento desse conjunto e garantir que esse algoritmo de melhoramento termine em um número finito e limitado de iterações.

Com isto, garantimos que os modelos baseados em interpolação polinomial serão plenamente lineares e plenamente quadráticos, de acordo com as definições vistas no capítulo anterior.

O primeiro algoritmo que apresentamos é para encontrar a base de Lagrange de um conjunto Y de pontos que tenha cardinalidade igual à dimensão do espaço de polinômios \mathcal{P}_n^d .

Algoritmo 3.1. [5, Algoritmo 6.1] *Calcula Base de Lagrange*

Dados:

- Uma aproximação inicial para a base de Lagrange $\ell_i = \phi_i$, $i = 0, 1, \dots, p$.

PARA $i = 0, \dots, p$

PASSO 1 - SELEÇÃO

Encontre $j_i = \arg \max_{i \leq j \leq p} |\ell_i(y^j)|$.

SE $\ell_i(y^{j_i}) = 0$,

Pare, pois o conjunto Y não é posicionado.

SENÃO

Troque os pontos y^i e y^{j_i} de posição no conjunto Y .

PASSO 2 - NORMALIZAÇÃO

$$\ell_i(x) = \frac{\ell_i(x)}{\ell_i(y^i)}$$

PASSO 3 - ORTOGONALIZAÇÃO

PARA $j = 0, \dots, p$, $j \neq i$, FAÇA

$$\ell_j(x) = \ell_j(x) - \ell_j(y^i)\ell_i(x).$$

Mostremos que o Algoritmo 3.1 calcula a base de polinômios de Lagrange para conjuntos posicionados. Na primeira iteração temos

$$\ell_0(x) = \frac{\phi_0(x)}{\phi_0(y^0)},$$

logo $\ell_0(y^0) = 1$. Para $j = 1, \dots, p$,

$$\ell_j(x) = \phi_j(x) - \phi_j(y^0)\ell_0(x),$$

então $\ell_j(y^0) = 0$.

Na segunda iteração,

$$\ell_1(x) = \frac{\ell_1(x)}{\ell_1(y^1)},$$

logo $\ell_1(y^1) = 1$, e $\ell_0(y^1) = 0$. Para $j = 2, \dots, p$,

$$\ell_j(x) = \ell_j(x) - \ell_j(y^1)\ell_1(x),$$

então $\ell_j(y^1) = 0$.

Temos também que para $j = 0$

$$\ell_0(x) = \ell_0(x) - \ell_0(y^1)\ell_1(x),$$

o que garante que $\ell_0(y^0) = 1$, pois $\ell_0(y^1) = 0$. Portanto uma iteração não destrói a anterior, e com isso ao fim de $(p + 1)$ iterações temos

$$\ell_j(y^i) = \delta_{ij} = \begin{cases} 1, & \text{se } i = j, \\ 0, & \text{se } i \neq j, \end{cases}$$

o que por definição é uma base de Lagrange.

Para ilustrar o funcionamento do Algoritmo 3.1 vamos construir um exemplo.

Exemplo 3.2.

Sejam o conjunto $Y = \{(0, 0), (1, 0), (0, 1), (2, 0), (0, 2), (1, 1)\}$ e a base inicial $\phi = \{1, x_1, x_2, x_1^2, x_2^2, x_1x_2\}$. Então para $i = 0$, precisamos encontrar o ponto que atinge o máximo do polinômio $\ell_0(x) = 1$ entre os pontos de Y . Nesse caso todos os pontos de Y são pontos de máximo e com isso consideramos $j_0 = 0$ e não precisamos trocar os pontos de lugar. O segundo passo é fazermos a Normalização, assim

$$\ell_0(x) = \frac{1}{1} = 1.$$

O próximo passo é a Ortogonalização que vai atualizar os demais polinômios da base inicial. Ou seja,

$$\ell_1(x) = \ell_1(x) - \ell_1(y^0)\ell_0(x) = x_1 - 0 \cdot 1 = x_1,$$

$$\ell_2(x) = \ell_2(x) - \ell_2(y^0)\ell_0(x) = x_2 - 0 \cdot 1 = x_2,$$

$$\ell_3(x) = \ell_3(x) - \ell_3(y^0)\ell_0(x) = x_1^2 - 0 \cdot 1 = x_1^2,$$

$$\ell_4(x) = \ell_4(x) - \ell_4(y^0)\ell_0(x) = x_2^2 - 0 \cdot 1 = x_2^2,$$

$$\ell_5(x) = \ell_5(x) - \ell_5(y^0)\ell_0(x) = x_1x_2 - 0 \cdot 1 = x_1x_2.$$

Após a primeira iteração nenhuma alteração foi feita.

Para a segunda iteração precisamos encontrar o ponto que atinge o máximo do polinômio $\ell_1(x) = x_1$, que neste caso é o ponto $(2, 0)$, e com isso trocamos este com o segundo ponto do conjunto, logo $Y = \{(0, 0), (2, 0), (0, 1), (1, 0), (0, 2), (1, 1)\}$. Normalizando, obtemos

$$\ell_1(x) = \frac{\ell_1(x)}{\ell_1(y^1)} = \frac{x_1}{2}.$$

Após o terceiro passo temos

$$\ell_0(x) = \ell_0(x) - \ell_0(y^1)\ell_1(x) = 1 - 1 \cdot \frac{x_1}{2} = 1 - \frac{x_1}{2},$$

$$\ell_2(x) = \ell_2(x) - \ell_2(y^1)\ell_1(x) = x_2 - 0 \cdot \frac{x_1}{2} = x_2,$$

$$\ell_3(x) = \ell_3(x) - \ell_3(y^1)\ell_1(x) = x_1^2 - 4 \cdot \frac{x_1}{2} = x_1^2 - 2x_1,$$

$$\ell_4(x) = \ell_4(x) - \ell_4(y^1)\ell_1(x) = x_2^2 - 0 \cdot \frac{x_1}{2} = x_2^2,$$

$$\ell_5(x) = \ell_5(x) - \ell_5(y^1)\ell_1(x) = x_1x_2 - 0 \cdot \frac{x_1}{2} = x_1x_2.$$

Logo, ao final da segunda iteração a nova base é $\ell = \{1 - 0.5x_1, 0.5x_1, x_2, x_1^2 - 2x_1, x_2^2, x_1x_2\}$.

Quando $i = 2$, o ponto que atinge o máximo de $\ell_2(x) = x_2$ é o ponto $y^4 = (0, 2)$ e ao atualizar Y temos $Y = \{(0, 0), (2, 0), (0, 2), (1, 0), (0, 1), (1, 1)\}$, ao fazermos a normalização

$$\ell_2(x) = \frac{\ell_2(x)}{\ell_2(y^2)} = \frac{x_2}{2}.$$

Após a Ortogonalização temos

$$\ell_0(x) = \ell_0(x) - \ell_0(y^2)\ell_2(x) = 1 - \frac{x_1}{2} - 1 \cdot \frac{x_2}{2} = 1 - \frac{x_1}{2} - \frac{x_2}{2},$$

$$\ell_1(x) = \ell_1(x) - \ell_1(y^2)\ell_2(x) = \frac{x_1}{2} - 0 \cdot \frac{x_2}{2} = \frac{x_1}{2},$$

$$\ell_3(x) = \ell_3(x) - \ell_3(y^2)\ell_2(x) = x_1^2 - 2x_1 - 0 \cdot \frac{x_2}{2} = x_1^2 - 2x_1,$$

$$\ell_4(x) = \ell_4(x) - \ell_4(y^2)\ell_2(x) = x_2^2 - 4 \cdot \frac{x_2}{2} = x_2^2 - 2x_2,$$

$$\ell_5(x) = \ell_5(x) - \ell_5(y^2)\ell_2(x) = x_1x_2 - 0 \cdot \frac{x_2}{2} = x_1x_2.$$

Ao final da terceira iteração a nova base é $\ell = \{1 - 0.5x_1 - 0.5x_2, 0.5x_1, 0.5x_2, x_1^2 - 2x_1, x_2^2 - 2x_2, x_1x_2\}$.

Para $i = 3$, temos ao final da iteração a nova base $\ell = \{1 - 1.5x_1 - 0.5x_2 + 0.5x_1^2, 0.5x_1^2 - 0.5x_1, 0.5x_2, -x_1^2 + 2x_1, x_2^2 - 2x_2, x_1x_2\}$.

Quando $i = 4$, $\ell = \{1 - 1.5x_1 - 1.5x_2 + 0.5x_1^2 + 0.5x_2^2, 0.5x_1^2 - 0.5x_1, 0.5x_2^2 - 0.5x_2, -x_1^2 + 2x_1, -x_2^2 + 2x_2, x_1x_2\}$.

E finalmente para $i = 5$, $\ell = \{1 - 1.5x_1 - 1.5x_2 + 0.5x_1^2 + 0.5x_2^2 + x_1x_2, 0.5x_1^2 - 0.5x_1, 0.5x_2^2 - 0.5x_2, 2x_1 - x_1^2 - x_1x_2, 2x_2 - x_2^2 - x_1x_2, x_1x_2\}$ que é a base de Lagrange para o conjunto $Y = \{(0, 0), (1, 0), (0, 1), (2, 0), (0, 2), (1, 1)\}$.

O segundo algoritmo é para o caso que o conjunto Y tem menos pontos do que $(p+1)$. E no caso em que a cardinalidade de Y é igual a $(p+1)$, ele segue o mesmo procedimento do Algoritmo 3.1, produzindo a mesma base de Lagrange.

Algoritmo 3.2. [5, Algoritmo 6.2] *Completa um conjunto não posicionado*

Dados:

- Uma aproximação inicial para a base de Lagrange $\ell_i = \phi_i$, $i = 0, 1, \dots, p$.
- $p_{ini} + 1$ o número de pontos no conjunto Y inicial.

PARA $i = 0, \dots, p$

PASSO 1 - SELEÇÃO

Encontre $j_i = \arg \max_{i \leq j \leq p_{ini}} |\ell_i(y^j)|$.

SE $\ell_i(y^{j_i}) > 0$ E $i \leq p_{ini}$,

Troque os pontos y^i e y^{j_i} de posição no conjunto Y .

SENÃO

Calcule $y^i \in \arg \max_{x \in B} |\ell_i(x)|$.

PASSO 2 - NORMALIZAÇÃO

$$\ell_i(x) = \frac{\ell_i(x)}{\ell_i(y^i)}$$

PASSO 3 - ORTOGONALIZAÇÃO

PARA $j = 0, \dots, p$, $j \neq i$, FAÇA

$$\ell_j(x) = \ell_j(x) - \ell_j(y^i)\ell_i(x).$$

Este algoritmo primeiramente inclui os pontos em Y para formar um conjunto posicionado, em seguida encontra pontos que maximizam os polinômios de Lagrange até que o

conjunto Y tenha $p + 1$ pontos. Ao final temos um conjunto posicionado e sua respectiva base de Lagrange.

Uma diferença entre os Algoritmos 3.1 e 3.2 é que o segundo precisa de um conjunto B onde os pontos de Y devem estar, já no primeiro esse requisito não é necessário.

Exemplo 3.3.

Vamos agora aplicar o Algoritmo 3.2 para encontrar um conjunto posicionado. Sejam o conjunto $Y = \{(0, 0), (2, 0), (0, 2), (1, 0), (0, 1)\}$ que é subconjunto de

$$B = \{x \in R^2 \mid \|x - (1, 1)\|_\infty \leq 1\},$$

e a base inicial $\phi = \{1, x_1, x_2, x_1^2, x_2^2, x_1x_2\}$. Vamos completar o conjunto Y . Até a quinta iteração o Algoritmo 3.2 segue como o Algoritmo 3.1 e portanto teremos a seguinte base $\ell = \{1 - 1.5x_1 - 1.5x_2 + 0.5x_1^2 + 0.5x_2^2, 0.5x_1^2 - 0.5x_1, 0.5x_2^2 - 0.5x_2, -x_1^2 + 2x_1, -x_2^2 + 2x_2, x_1x_2\}$.

Agora precisamos encontrar um ponto (x_1, x_2) que maximize o polinômio $\ell_5(x) = x_1x_2$ em B , que é o ponto $y^5 = (2, 2)$. E com isso conseguimos um conjunto posicionado $Y = \{(0, 0), (2, 0), (0, 2), (1, 0), (0, 1), (2, 2)\}$ e a seguinte base de Lagrange $\ell = \{1 - 1.5x_1 - 1.5x_2 + 0.5x_1^2 + 0.5x_2^2 + 0.25x_1x_2, 0.5x_1^2 - 0.5x_1 - 0.25x_1x_2, 0.5x_2^2 - 0.5x_2 - 0.25x_1x_2, 2x_1 - x_1^2 - x_1x_2, 2x_2 - x_2^2 - x_1x_2, 0.25x_1x_2\}$.

O terceiro algoritmo que apresentamos é para o caso em que temos um conjunto posicionado com exatos $p + 1$ pontos e queremos verificar se Y é Λ -posicionado, para um constante $\Lambda > 1$ dada. Se isso não acontecer, o algoritmo muda os pontos do conjunto Y para que o conjunto seja Λ -posicionado em um conjunto B .

Algoritmo 3.3. [5, Algoritmo 6.3] Melhora do Posicionamento

Dados:

- Uma constante $\Lambda > 1$.
- Um conjunto posicionado Y , com a respectiva base de Lagrange.

PARA $k = 1, 2, 3, \dots$

PASSO 1

$$\text{Estime } \Lambda_{k-1} = \max_{0 \leq i \leq p} \max_{x \in B} |\ell_i(x)|.$$

PASSO 2

$$\text{SE } \Lambda_{k-1} > \Lambda$$

Seja $i_k \in \{0, \dots, p\}$ um índice para o qual $\max_{x \in B} |\ell_{i_k}(x)| > \Lambda$,

Seja $y_*^{i_k} \in B$ um ponto que maximiza $|\ell_{i_k}(x)|$ em B .

Atualize Y realizando a mudança de ponto $Y = Y \cup \{y_*^{i_k}\} \setminus \{y^{i_k}\}$.

SENÃO

Y é Λ -posicionado, pare.

PASSO 3

Atualize todos os coeficientes dos polinômios de Lagrange.

Denote por $\ell_j^*(x)$, $j = 0, \dots, p$ os novos polinômios de Lagrange.

Faça $\ell_{i_k}^* = \frac{\ell_{i_k}}{\ell_{i_k}(y_*^{i_k})}$.

Faça $\ell_j^* = \ell_j(x) - \ell_j(y_*^{i_k})\ell_{i_k}^*(x)$, para todo $j \in \{0, \dots, p\}$ e $j \neq k$.

Este algoritmo depende da precisão para encontrar Λ_{k-1} no Passo 1. Os valores absolutos máximos dos polinômios de Lagrange em B podem ser estimados, desde que possamos garantir que $\Lambda_{k-1} > \Lambda$ ou $\Lambda_{k-1} \leq \Lambda$. O Passo 2 também pode ser estimado da mesma maneira. No Passo 3 podemos fazer o passo de normalização pois, uma vez que, se $\ell_{i_k}(y_*^{i_k}) = 0$, o conjunto não é posicionado, e assim precisamos aplicar o Algoritmo 3.2 antes de aplicar o Algoritmo 3.3.

Provaremos que para qualquer $\Lambda > 1$ o Algoritmo 3.3 para em um número finito e uniformemente limitado de passos.

Teorema 3.20. [5, Teorema 6.3] *Dados $\Lambda > 1$, uma bola fechada B e uma base de polinômios ϕ fixa, o Algoritmo 3.3 termina com um conjunto Y Λ -posicionado em até $L = L(\Lambda, \phi)$ iterações, onde L é uma constante que depende de Λ e ϕ .*

Demonstração. Vamos dividir a prova em três partes.

Parte 1. Vamos relembrar a definição equivalente dos polinômios de Lagrange:

$$|\ell_i(x)| = \frac{\text{vol}(\phi, Y_i(x))}{\text{vol}(\phi, Y)}$$

para uma base fixa ϕ , onde $Y_i(x) = Y \setminus \{y^i\} \cup \{x\}$. Temos que $|\ell_{i_k}(y_*^{i_k})|$ é igual à razão do volume do simplex de vértices em $\phi(Y^*)$ e o volume do simplex de vértices em $\phi(Y)$. Isso significa que cada vez que um ponto y^{i_k} é substituído por um novo ponto $y_*^{i_k}$ tal que $|\ell_{i_k}(y_*^{i_k})| > \Lambda > 1$, o volume do simplex é incrementado por pelo menos um fator de Λ .

Todos os pontos são considerados na bola fechada B , e portanto $\phi(B)$ é um conjunto compacto em \mathbb{R}^{p+1} . Isso significa que o volume do simplex definido por $\phi(Y)$ não pode ser aumentado infinitamente por um fator constante $\Lambda > 1$. Seja $V_{\max} = V(\phi, Y_{\max})$ o

volume máximo de todos os simplex formados por $\phi(Y)$ para $Y \subset B$, e seja $V(\phi, Y_0)$ o volume do simplex definido pelos vértices em $\phi(Y_0)$ para o conjunto inicial Y_0 .

Assim se L_0 é o número de iterações necessárias para chegar ao simplex de volume máximo temos que

$$\Lambda^{L_0} \leq \frac{V_{\max}}{V(\phi, Y_0)},$$

portanto o número de passos do Algoritmo 3.3 não é maior que

$$\log_{\Lambda} \left(\frac{V_{\max}}{V(\phi, Y_0)} \right).$$

Parte 2. Vamos mostrar agora que, depois de no máximo $p+1$ passos o Algoritmo 3.3 calcula um conjunto Y tal que o volume do simplex satisfaz $V(\phi, Y) \geq \Theta(p, \phi) > 0$.

Primeiro vamos mostrar após no máximo $(p+1)$ iterações, o algoritmo obtém um conjunto Y que é 2^p -posicionado. Vamos assumir que em uma iteração do Algoritmo 3.3 uma atualização é realizada, o ponto y^{i_1} é trocado, e os polinômios de Lagrange atualizados. Depois da atualização, como ℓ_{i_1} é dividido pelo seu valor máximo, temos que $\max_{x \in B} |\ell_{i_1}(x)| \leq 1$. Em seguida, na próxima iteração onde há troca de pontos, $i_2 \neq i_1$, uma vez que escolhemos um polinômio cujo valor é maior do que 1 em B . Então após a atualização temos novamente que $\max_{x \in B} |\ell_{i_2}(x)| \leq 1$, e também que $\max_{x \in B} |\ell_{i_1}(x)| \leq 2$, uma vez que $\ell_{i_1}(x) = \ell_{i_1}(x) - \ell_{i_1}(y^{i_2})\ell_{i_2}(x)$. Depois da terceira iteração ou temos um conjunto 2-posicionado ou $i_3 \notin \{i_1, i_2\}$, e então depois da atualização temos que $\max_{x \in B} |\ell_{i_3}(x)| \leq 1$, $\max_{x \in B} |\ell_{i_2}(x)| \leq 2$ e $\max_{x \in B} |\ell_{i_1}(x)| \leq 4$. Estendendo esse argumento e aplicando o processo de indução temos que após $(p+1)$ passos teremos um conjunto Y 2^p -posicionado.

Parte 3. Usando as propriedades dos polinômios de Lagrange com respeito a translação e escala temos que dado um conjunto Y em uma bola B o Algoritmo 3.3 trabalha idênticamente para o caso quando é aplicado na versão \hat{Y} e na bola $B(0, 1)$. Por idêntico comportamento queremos dizer que o algoritmo gera a mesma sequência de pontos e que o novo conjunto de amostra é idêntico ao com translação e mudança de escala. Assim quando rodamos o Algoritmo $p+1$ passos, produzimos o mesmo conjunto (com translação e mudança de escala) como se aplicássemos o algoritmo em \hat{Y} na bola $B(0, 1)$, logo podemos aplicar o Corolário 3.17. Isso significa que o volume $V(\phi, Y)$ do simplex em vértices em Y obtido após $p+1$ iterações é uniformemente limitado com $V(\phi, Y) \geq \Theta(p, \phi) > 0$, onde $\Theta(p, \phi)$ depende somente de p e ϕ . Assim,

$$\log_{\Lambda} \left(\frac{V_{\max}}{V(\phi, Y_0)} \right)$$

é uniformemente limitado por uma constante L_1 que depende apenas de Λ , p e ϕ . Tomando $L = L_1 + p + 1$ concluímos que o Algoritmo 3.3 termina em no máximo L iterações. \square

Para ilustrar o Algoritmo 3.3 vamos apresentar o seguinte exemplo.

Exemplo 3.4.

Seja o conjunto $Y = \{(0, 0), (2, 0), (0, 1), (1, 0), (0, 2), (1, 1)\}$, que é um conjunto posicionado em $B = \{x \in R^2 \mid \|x - (1, 1)\|_\infty \leq 1\}$. Primeiramente vamos encontrar a constante de posicionamento em B . Ao aplicarmos o Algoritmo 3.1, encontramos a seguinte base de Lagrange: $\ell = \{1 - 1.5x_1 - 1.5x_2 + 0.5x_1^2 + 0.5x_2^2 + x_1x_2, 0.5x_1^2 - 0.5x_1, 0.5x_2^2 - 0.5x_2, 2x_1 - x_1^2 - x_1x_2, 2x_2 - x_2^2 - x_1x_2, x_1x_2\}$.

Pela definição de Λ , precisamos encontrar o máximo valor absoluto dos polinômios da base de Lagrange em B , assim

$$\max_{x \in B} |1 - 1.5x_1 - 1.5x_2 + 0.5x_1^2 + 0.5x_2^2 + x_1x_2| = 3, \text{ em } (2, 2).$$

$$\max_{x \in B} |0.5x_1^2 - 0.5x_1| = 1, \text{ em } (2, x_2).$$

$$\max_{x \in B} |0.5x_2^2 - 0.5x_2| = 1, \text{ em } (x_1, 2).$$

$$\max_{x \in B} |2x_1 - x_1^2 - x_1x_2| = 2, \text{ em } (2, 0) \text{ e } (2, 2).$$

$$\max_{x \in B} |2x_2 - x_2^2 - x_1x_2| = 2, \text{ em } (0, 2) \text{ e } (2, 2).$$

$$\max_{x \in B} |x_1x_2| = 4, \text{ em } (2, 2).$$

Portanto $\Lambda = 4$ é a constante de posicionamento do conjunto Y em B . Queremos melhorar o posicionamento de Y para que este tenha constante de posicionamento $\Lambda = 1$ em B . Por simplicidade, vamos, durante os Passos 1 e 2 do Algoritmo 3.3, procurar o máximo em B através de um conjunto amostra aleatório com um número finito de pontos, para este exemplo usaremos $t = 1000$ para esta quantidade de pontos.

A Figura 3.6 mostra o resultado da aplicação do Algoritmo 3.3, que em apenas uma iteração consegue um conjunto 1-posicionado em B . O conjunto encontrado é o conjunto $Y = \{(0, 0), (2, 0), (0, 1), (1, 0), (0, 2), (1.9935, 1.9964)\}$ e o ponto que realmente maximizaria os polinômios seria o ponto $(2, 2)$.

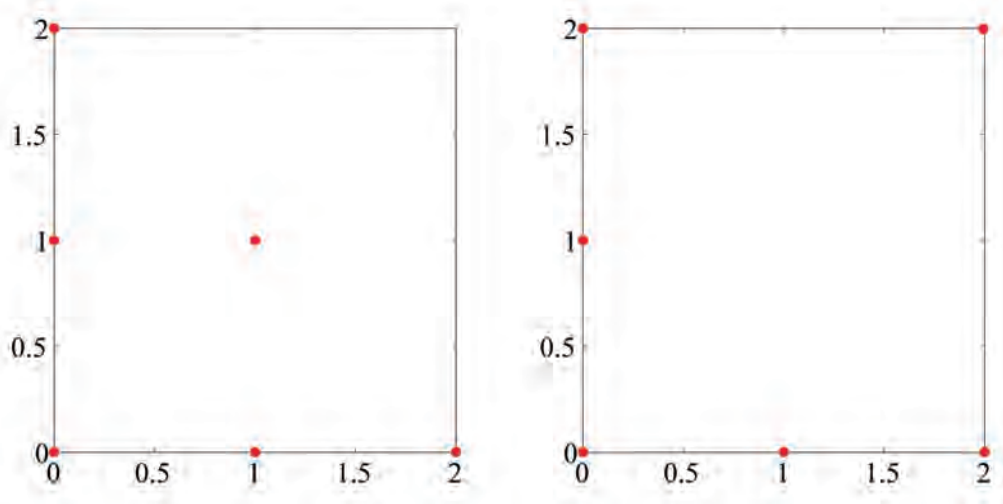


Figura 3.6: Aplicação do Algoritmo 3.3.

Vimos que o posicionamento de um conjunto pode ser expresso através do número de condição da matriz $M(\bar{\phi}, \hat{Y})$, onde \hat{Y} é uma versão transladada e com mudança de escala de Y . Agora vamos mostrar um algoritmo equivalente ao Algoritmo 3.2, o qual considera o número de condição mencionado.

O Algoritmo 3.2 confirma se um conjunto Y é posicionado em B e o modifica ou completa, se necessário, para formar um conjunto posicionado. O algoritmo seguinte apresenta o mesmo resultado aplicando eliminação gaussiana com pivoteamento por linhas da matriz $M(\bar{\phi}, Y)$. O i -ésimo pivô da eliminação gaussiana é selecionado da i -ésima coluna da matriz $M(\bar{\phi}, Y)$ modificada depois da fatoração das primeiras $i - 1$ colunas. Os elementos da i -ésima coluna de $M(\bar{\phi}, Y)$ modificada são valores de certos polinômios avaliados no conjunto Y . Vamos chamar esses polinômios de Polinômios Pivôs e os denotaremos por u_i , $i = 1, 2, \dots, p$. Veremos que os polinômios pivô estão relacionados com os Polinômios de Lagrange.

Algoritmo 3.4. [5, Algoritmo 6.4] *Completa um conjunto não posicionado via fatoração LU*

Dados:

- Uma aproximação inicial para a base de polinômios pivôs $u_i = \phi_i$, $i = 1, \dots, p$.
- $p_{ini} + 1$ o número de pontos no conjunto Y inicial.

PARA $i = 0, \dots, p$

PASSO 1 - SELEÇÃO

Encontre $j_i = \arg \max_{i \leq j \leq p_{ini}} |u_i(y^j)|$.

SE $|u_i(y^{j^i})| > 0$ E $i \leq p_{ini}$,

Troque os pontos y^i e y^{j^i} de posição no conjunto Y .

SENÃO

Calcule $y^i \in \arg \max_{x \in B} |u_i(x)|$.

PASSO 2 - ELIMINAÇÃO GAUSSIANA

PARA $j = i + 1, \dots, p$, FAÇA

$$u_j(x) = u_j(x) - \frac{u_j(y^i)}{u_i(y^i)} u_i(x).$$

Uma diferença entre os Algoritmos 3.2 e 3.4 é que o último não normaliza os polinômios pivôs. Após o término do algoritmo temos um conjunto de polinômios pivôs $u = \{u_0, u_1, \dots, u_p\}$ e a matriz correspondente $M(u, Y)$ é matriz triangular superior da fatoração LU de $M(\phi, Y)$. Apesar dessa diferença os Algoritmos 3.2 e 3.4 produzem conjuntos idênticos de pontos. É o que mostramos no próximo lema.

Lema 3.21. [5, Lema 6.4] *Dado o mesmo conjunto inicial Y , possivelmente não posicionado, na mesma região B , e a mesma base inicial ϕ . Então os Algoritmos 3.2 e 3.4 produzem idênticos conjuntos posicionados Y .*

Demonstração. Temos que na primeira iteração, quando $i = 0$, que ambos algoritmos selecionam o mesmo ponto para y^0 . Vamos supor como hipótese de indução que as primeiras k iterações (a partir da iteração 0) de ambos os algoritmos resultam em um mesmo conjunto de pontos $\{y^0, \dots, y^{k-1}\}$. Na k -ésima iteração do Algoritmo 3.4 o k -ésimo polinômio pivô apresenta as seguintes propriedades: é combinação linear dos primeiros $k + 1$ elementos da base ϕ e seu valor é zero nos primeiros k pontos y^0, \dots, y^{k-1} de Y . O mesmo é verdade para o k -ésimo polinômio de Lagrange ℓ_k no início da k -ésima iteração do Algoritmo 3.2. Como pela hipótese de indução os primeiros k pontos produzidos pelos dois algoritmos são os mesmos, isso significa que no início da k -ésima iteração os dois algoritmos consideram dois polinômios que (i) estão no mesmo subespaço $(k + 1)$ -dimensional de polinômios e (ii) o valor é zero nos k pontos de um subconjunto posicionado. Então, os dois polinômios devem coincidir a menos de uma constante. Disto segue que quando $i = k$, o ponto selecionado pelos Algoritmos 3.2 e 3.4 produzem o mesmo ponto y^k . Por indução segue o lema. \square

Agora vamos considerar algoritmos possíveis para melhorar o posicionamento de um conjunto posicionado Y usando polinômios pivô. Vamos aplicar uma mudança no Algoritmo 3.4 onde usamos um limitante χ para aceitar o próximo pivô durante o passo de seleção.

Algoritmo 3.5. [5, Algoritmo 6.5] *Melhora o posicionamento via fatoração LU*

Dados:

- Uma aproximação inicial para a base de polinômios pivô $u_i = \phi_i$, $i = 1, \dots, p$.
- Uma constante $\chi > 0$ para seleção do pivô.

PARA $i = 0, \dots, p$

PASSO 1 - SELEÇÃO

Se possível, escolha $j_i \in \{i, \dots, |Y| - 1\}$ tal que $|u_i(y^{j_i})| \geq \chi$.

SE j_i é encontrado,

Troque os pontos y^i e y^{j_i} de posição no conjunto Y .

SENÃO

Calcule $y^i \in \arg \max_{x \in B} |u_i(x)|$.

Pare se $|u_i(y^{j_i})| < \chi$, pois o limitante do pivô é muito grande.

PASSO 2 - ELIMINAÇÃO GAUSSIANA

PARA $j = i + 1, \dots, p$, FAÇA

$$u_j(x) = u_j(x) - \frac{u_j(y^i)}{u_i(y^i)} u_i(x).$$

O resultado do algoritmo modificado é um conjunto Y posicionado tal que quando a eliminação gaussiana é aplicada em $M(\hat{\phi}, Y)$ todos os pivôs possuem valor absoluto não menores que χ . O limitante χ é usado para conseguir conjuntos melhores posicionados sem ter que descartar muitos pontos de Y . Se χ é muito pequeno, muitos pontos serão aceitos no conjunto de interpolação, o que implica que o conjunto pode não ser bem posicionado. Por outro lado, se o valor de χ é muito grande, é necessário trocar quase todos os pontos de Y para satisfazer a condição de limitação. E ainda pode acontecer casos em que é impossível encontrar um ponto que obtenha um valor de pivô grande o suficiente. Felizmente, conseguimos limitantes superiores para χ para garantir que o algoritmo é bem sucedido.

Como podemos expressar o $(k + 1)$ -ésimo polinômio pivô u_k como $(v^k)^\top \hat{\phi}$, onde $v^k = (v_0, \dots, v_{k-1}, 1, 0, \dots, 0)$, temos que $\|v^k\|_\infty \geq 1$. Pelo Lema 3.11, existe $\sigma > 0$ tal que se o limitante χ do pivô é escolhido entre 0 e σ , então o Algoritmo 3.5 é bem sucedido. Pelos Lemas 3.12 e 3.13 conseguimos limitantes para σ . Temos que quando $d = 1$, $\sigma = 1$ e, quando $d = 2$, temos $\sigma = 1/4$.

Por outro lado, se queremos usar a abordagem por pivoteamento mas atualizar apenas um ponto por vez, usamos o algoritmo descrito a seguir. Essencialmente, é o Algoritmo 3.5 sem limitante para o pivô. Quando assumimos que o conjunto inicial Y é posicionado

e possui $(p + 1)$ pontos, o algoritmo vai rodar até a última iteração, onde otimiza o último polinômio pivô e atualiza o último ponto em Y .

Algoritmo 3.6. [5, Algoritmo 6.6] *Incrementa posicionamento via fatoração LU*

Dados:

- Uma aproximação inicial para a base de polinômios pivô $u_i = \phi_i$, $i = 1, \dots, p$.
- Um conjunto inicial Y posicionado contendo $p + 1$ pontos.

PARA $i = 0, \dots, p - 1$

PASSO 1 - SELEÇÃO

Encontre $j_i = \arg \max_{i \leq j \leq p} |u_i(y^j)|$.

Troque os pontos y^i e y^{j_i} de posição no conjunto Y .

PASSO 2 - ELIMINAÇÃO GAUSSIANA

PARA $j = i + 1, \dots, p$, FAÇA

$$u_j(x) = u_j(x) - \frac{u_j(y^i)}{u_i(y^i)} u_i(x).$$

Passo de Melhoramento

$$y_{\text{new}}^p \in \arg \max_{x \in B} |u_p(x)|.$$

Este algoritmo tenta identificar o melhor ponto candidato para a atualização. Como a fatoração LU produz pivôs por linhas (ou seja, por pontos), os pontos que resultam em um pivô ruim são levados para as últimas linhas. Assim, ao substituir o último ponto, esperamos obter melhoria no posicionamento.

Para identificar o próximo ponto a ser substituído, o Algoritmo 3.6 deve ser repetido desde o início. Se após a aplicação do Algoritmo 3.6 o último ponto já é o ponto ótimo para o correspondente polinômio pivô, então nenhuma melhora adicional poderá ser obtida por esta estratégia. Não há nenhuma garantia de que a sequência de pontos gerada pelo Algoritmo 3.3 é idêntica à sequência de pontos gerada pela estratégia que consiste em aplicar sequencialmente o Algoritmo 3.6. No entanto, quando essas sequências são idênticas, ou seja, se o ponto que produz o maior valor absoluto dos polinômios de Lagrange é o ponto que produz o menor pivô são os mesmos em cada iteração, então os conjuntos de interpolação encontrados são idênticos.

Capítulo 4

Experimentos Numéricos

Neste capítulo vamos apresentar alguns testes computacionais feitos com uma adaptação dos algoritmos apresentados no Capítulo 2. Os testes aos quais o método foi submetido constituem uma coletânea de problemas organizada por Moré, Garbow e Hillstom [13]. Trata-se de um conjunto de 35 problemas diferenciáveis de minimização irrestrita, onde as funções objetivo são soma de quadrados. Em alguns dos problemas é possível escolher a quantidade de variáveis, dessa maneira fizemos estas escolhas de tal modo a conseguirmos 50 problemas.

4.1 Algoritmo Implementado

Os Algoritmos 2.1 e 2.3, descritos no Capítulo 2, requerem classes de modelos plenamente lineares e plenamente quadráticos, respectivamente. Utilizamos na implementação do algoritmo interpolação polinomial completa, ou seja, usamos $p = \frac{(n+1)(n+2)}{2}$ pontos de interpolação para construir os modelos. Neste caso, como vimos no Capítulo 3, se o conjunto é Λ -posicionado, os modelos gerados são plenamente lineares e plenamente quadráticos como desejamos. O algoritmo implementado é descrito abaixo.

Algoritmo 4.1. *Região de Confiança sem Derivadas com Interpolação Polinomial*

Dados:

- $x_0 \in \mathbb{R}^n$, $\bar{\Delta} > 0$, $\Delta_0 \in (0, \bar{\Delta}]$.

- $0 \leq \eta_0 \leq \eta_1 < 1$, com $\eta_1 \neq 0$, $0 < \gamma < 1 < \gamma_{inc}$, $\varepsilon_c > 0$, $\beta > \mu > 0$ e $\omega \in (0, 1)$.

PASSO 0 - MODELO INICIAL

Use o Algoritmo 3.2 para completar um conjunto Y_0 posicionado.

Calcule $m_0(x) = \sum_{i=0}^p f(y^i) \ell_i(x)$.

Calcule $g_0(x) = \nabla m_0(x)$.

Calcule $H_0(x) = \nabla^2 m_0(x)$.

$k = 0$.

PASSO 1 - CRITICALIDADE

Calcule $\sigma_k = \max(\|g_k\|, -\tau_k)$, onde τ_k é o menor autovalor de H_k .

SE $\sigma_k \leq \varepsilon_c$ E $\Delta_k > \mu\sigma_k$,

$i = 0$,

$m_k^{(0)} = m_k$.

REPITA

$i = i + 1$.

Aplique o Algoritmo 3.3 e melhore o posicionamento do conjunto Y_k .

Calcule $m_k^{(i)} = \sum_{i=0}^p f(y^i) \ell_i(x)$.

Calcule $g_k^{(i)} = \nabla m_k^{(i)}(x)$.

Calcule $H_k^{(i)} = \nabla^2 m_k^{(i)}(x)$.

Denote o novo modelo como \tilde{m}_k .

$\tilde{\Delta}_k = \omega^{i-1} \Delta_k$.

Calcule $\sigma_k^{(i)} = \max(\|g_k^{(i)}\|, -\tau_k^{(i)})$, onde $\tau_k^{(i)}$ é o menor autovalor de $H_k^{(i)}$.

ATÉ $\tilde{\Delta}_k \leq \mu\sigma_k^{(i)}$.

$\Delta_k = \min\{\max\{\tilde{\Delta}_k, \beta\sigma_k^{(i)}\}, \Delta_k\}$.

$m_k = \tilde{m}_k$.

SENÃO

Mantenha o modelo m_k .

Mantenha o raio Δ_k .

PASSO 2 - CÁLCULO DO PASSO

Calcule p_k que obtenha uma redução suficiente no modelo m_k , tal que $x_k + p_k \in B(x_k, \Delta_k)$.

PASSO 3 - ACEITAÇÃO DO PASSO

Calcule $\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(x_k) - m_k(x_k + p_k)}$.

SE $\rho_k \geq \eta_0$.

$x_{k+1} = x_k + p_k$.

Inclua x_{k+1} no conjunto de amostra e obtenha um novo modelo m_{k+1} .

SENÃO

$x_{k+1} = x_k$.

$m_{k+1} = m_k$.

PASSO 4 - MELHORAMENTO DO MODELO

SE $\rho_k < \eta_1$

Use o Algoritmo 3.3 para melhorar o modelo m_k .

Defina m_{k+1} como m_k (possivelmente melhorado).

PASSO 5 - ATUALIZAÇÃO DO RAIÃO

SE $\rho_k \geq \eta_1$

$$\Delta_{k+1} = \min\{\gamma_{inc}\Delta_k, \bar{\Delta}\}.$$

SENÃO

SE $\rho_k \geq \eta_0$

$$\Delta_{k+1} = \gamma\Delta_k.$$

SENÃO

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

$k = k + 1$.

Escolhemos utilizar os Algoritmos 3.2 e 3.3 para encontrar e melhorar os conjuntos posicionados para interpolação polinomial pois em ambos ao final temos o conjunto posicionado e a respectiva base de Lagrange. Para o subproblema dado no Passo 2 do Algoritmo 4.1 utilizamos o método Dogleg, caso a Hessiana do modelo seja definida positiva, ou Steihaug, caso contrário. Para maiores informações ver [14] e [19].

4.2 Resultados Computacionais

Para a implementação do Algoritmo 4.1 foi utilizado o software MATLAB[®] em sua versão R2010a em um computador portátil com processador Intel[®] Core[™] i5-430M Processor com 3 MB de memória cache, com velocidade do clock de 2.26 GHz e com 4 GB memória RAM, com sistema operacional Windows[®] 7 Professional.

Abaixo apresentamos os resultados obtidos com várias configurações dos parâmetros utilizados. Quanto à precisão do Algoritmo 3.3 para maximizar os polinômios de Lagrange em $B(x_k, \Delta_k)$ e encontrar um conjunto interpolador melhor posicionado, utilizamos um conjunto de amostra contendo t pontos, onde t depende da quantidade de variáveis do problema, e para todo os testes utilizamos $t = 10n$. Em todos os testes adotamos como critério de parada uma precisão 10^{-6} para o raio Δ_k ou um limite máximo de 1000 iterações. Fixamos também os seguintes valores para os parâmetros: $\mu = 1$, $\beta = 0.8$, $\varepsilon_c = 10^{-4}$ e $\omega = 0.5$. Os demais parâmetros foram diferentes em cada bateria de testes.

4.2.1 Bateria de Testes

Teste 1

Utilizamos os seguintes parâmetros:

$$\eta_0 = 0.25, \quad \eta_1 = 0.75, \quad \gamma = 0.6, \quad \gamma_{inc} = 2, \quad \Delta_0 = 5, \quad e \quad \bar{\Delta} = 50.$$

Tabela 4.1: *Resultados Numéricos - Teste 1*

P	n	m	tempo	iter.	solução	P	n	m	tempo	iter.	solução
1	2	2	1,02	110	$1,66 \times 10^{-11}$	23	4	5	9,04	192	$3,32 \times 10^{-5}$
2	2	2	1,26	151	48,98425	24	2	4	0,68	40	$8,07 \times 10^{-7}$
3	2	2	6,84	1000	$1,31 \times 10^{-1}$	24	4	8	2,08	30	$9,55 \times 10^{-6}$
4	2	3	6,87	1000	$9,81 \times 10^{11}$	25	2	4	0,56	24	$4,62 \times 10^{-22}$
5	2	3	0,57	43	$4,33 \times 10^{-11}$	25	5	7	5,70	48	$1,55 \times 10^{-16}$
6	2	10	0,70	57	124,36218	26	2	2	0,47	26	$6,80 \times 10^{-10}$
7	3	3	1,99	85	$1,81 \times 10^{-17}$	26	5	5	3,54	29	$1,64 \times 10^{-8}$
8	3	15	2,26	68	$8,21 \times 10^{-3}$	27	2	2	0,32	2	$1,58 \times 10^{-30}$
9	3	15	0,97	20	$1,50 \times 10^{-8}$	27	5	5	0,79	2	$2,56 \times 10^{-27}$
10	3	16	1,03	40	$6,49 \times 10^6$	28	2	2	0,68	51	$4,39 \times 10^{-23}$
11	3	20	-	-	-	28	5	5	9,72	80	$3,42 \times 10^{-20}$
12	3	20	3,57	151	$1,26 \times 10^{-6}$	29	2	2	0,63	47	$8,81 \times 10^{-20}$
13	4	4	2,39	38	$6,98 \times 10^{-8}$	29	5	5	5,24	37	$1,87 \times 10^{-8}$
14	4	6	13,32	264	$7,67 \times 10^{-14}$	30	2	2	0,74	50	$1,98 \times 10^{-21}$
15	4	11	8,29	141	$3,08 \times 10^{-4}$	30	5	5	7,66	58	$1,32 \times 10^{-18}$
16	4	20	3,79	64	85822,20163	31	2	2	0,69	52	$2,79 \times 10^{-19}$
17	5	33	-	-	-	31	5	5	5,97	49	$1,75 \times 10^{-18}$
18	6	13	83,33	358	$5,37 \times 10^{-8}$	32	2	2	0,35	3	0
19	11	65	2579,49	812	$4,01 \times 10^{-2}$	32	5	5	0,90	2	$1,35 \times 10^{-28}$
20	6	31	262,31	1000	$2,29 \times 10^{-3}$	33	2	2	0,41	6	0,2
20	9	31	717,72	1000	$3,19 \times 10^{-5}$	33	5	5	118,06	1000	0,90909
21	4	4	7,79	142	$6,82 \times 10^{-9}$	34	3	3	0,78	21	2
21	6	6	46,68	201	$1,30 \times 10^{-14}$	34	5	5	17,85	147	2,42857
22	8	8	30,30	57	$3,96 \times 10^{-9}$	35	2	2	0,59	39	$1,23 \times 10^{-9}$
23	2	3	0,49	18	$1,07 \times 10^{-5}$	35	5	5	3,71	35	$1,77 \times 10^{-10}$

A Tabela 4.1 apresenta os resultados obtidos, mostrando o número do problema, o número de variáveis, o tempo em segundos para resolver o problema e a solução encontrada. Apresentamos também o fator m que pode ser escolhido em cada problema, que representa a quantidade de fatores na soma de quadrados dos problemas dados em [13].

Dos 50 problemas testados, o algoritmo proposto resolveu 42 deles satisfatoriamente, dentro do critério de parada para o raio Δ_k . Dentre os 8 demais testes, 3 chegaram próximos da solução. Para os problemas 3 e 4, o algoritmo não encontrou solução antes

de 1000 iterações. Para o problema 10, apesar de realizar menos iterações, não encontrou a solução. E para os problemas 20 e 33, apesar de alcançar as 1000 iterações, conseguiu chegar bem próximo da solução. Para os problemas 11 e 17, houve dificuldade com o raio inicial, que originou números muito grande ao gerar o modelo, ocasionando erro do tipo Inf no Matlab.

Teste 2

Para a segunda rodada de testes, mantivemos o mesmo critério de parada do teste anterior e utilizamos os seguintes parâmetros:

$$\eta_0 = 0.1, \quad \eta_1 = 0.5, \quad \gamma = 0.5, \quad \gamma_{inc} = 3, \quad \Delta_0 = 2, \quad e \quad \bar{\Delta} = 100.$$

Tabela 4.2: *Resultados numéricos - Teste 2*

P	n	m	tempo	iter.	solução	P	n	m	tempo	iter.	solução
1	2	2	1,02	97	$2,22 \times 10^{-17}$	23	4	5	19,81	391	$2,25 \times 10^{-5}$
2	2	2	1,02	112	48,98425	24	2	4	0,60	30	$8,07 \times 10^{-7}$
3	2	2	6,82	1000	$1,31 \times 10^{-1}$	24	4	8	1,95	31	$9,82 \times 10^{-6}$
4	2	3	6,51	1000	$9,84 \times 10^{11}$	25	2	4	0,54	28	$1,72 \times 10^{-22}$
5	2	3	0,78	70	$1,72 \times 10^{-17}$	25	5	7	8,34	68	$1,84 \times 10^{-16}$
6	2	10	0,73	63	124,36218	26	2	2	0,68	42	$9,12 \times 10^{-22}$
7	3	3	2,32	91	$2,17 \times 10^{-17}$	26	5	5	3,46	33	$3,51 \times 10^{-8}$
8	3	15	1,35	35	$8,21 \times 10^{-3}$	27	2	2	0,32	2	$1,53 \times 10^{-29}$
9	3	15	3,78	101	$1,13 \times 10^{-8}$	27	5	5	0,83	2	$7,87 \times 10^{-27}$
10	3	16	0,81	27	$6,61 \times 10^6$	28	2	2	0,65	44	$6,91 \times 10^{-22}$
11	3	20	24,89	1000	$3,07 \times 10^{-2}$	28	5	5	8,33	60	$1,31 \times 10^{-20}$
12	3	20	6,79	239	$3,30 \times 10^{-11}$	29	2	2	0,59	40	$9,57 \times 10^{-19}$
13	4	4	1,98	30	$5,53 \times 10^{-8}$	29	5	5	5,11	50	$2,02 \times 10^{-20}$
14	4	6	7,22	140	$8,60 \times 10^{-11}$	30	2	2	0,70	50	$1,27 \times 10^{-19}$
15	4	11	8,69	164	$3,08 \times 10^{-4}$	30	5	5	6,17	50	$1,53 \times 10^{-18}$
16	4	20	4,08	68	85822,20163	31	2	2	0,64	44	$1,87 \times 10^{-16}$
17	5	33	-	-	-	31	5	5	9,22	70	$2,06 \times 10^{-17}$
18	6	13	82,40	339	$1,97 \times 10^{-7}$	32	2	2	0,33	3	$2,47 \times 10^{-32}$
19	11	65	3068,50	1000	$4,01 \times 10^{-2}$	32	5	5	0,92	3	$4,93 \times 10^{-29}$
20	6	31	268,40	1000	$2,34 \times 10^{-3}$	33	2	2	0,42	7	0,2
20	9	31	703,34	1000	$4,19 \times 10^{-5}$	33	5	5	25,70	229	0,90909
21	4	4	8,16	163	$9,75 \times 10^{-10}$	34	3	3	0,69	11	2
21	6	6	50,88	225	$3,96 \times 10^{-9}$	34	5	5	117,75	1000	2,42857
22	8	8	49,49	65	$1,27 \times 10^{-9}$	35	2	2	0,63	39	$2,62 \times 10^{-10}$
23	2	3	0,45	16	$1,09 \times 10^{-5}$	35	5	5	4,28	41	$7,88 \times 10^{-10}$

A Tabela 4.2 exibe os resultados para a bateria de testes 2. Observe que dos 50 problemas propostos, o algoritmo resolveu 41 dentro do critério de parada para o raio Δ_k e, dos 9 restantes chegou próximo à solução em 4 deles. No entanto, não encontrou

a solução nos demais. Novamente para os problemas 3 e 4 o algoritmo não encontrou solução antes de 1000 iterações, apesar de reduzir o valor da função. No problema 10, apesar de o algoritmo terminar antes de 1000 iterações, não encontrou a solução. Para os problemas 11, 19, 20, e 34, apesar de alcançar 1000 iterações, o algoritmo conseguiu chegar próximo das soluções. Novamente no problema 17 houve dificuldade ocasionada pelo raio inicial o que gerou erro do tipo Inf no Matlab.

Teste 3

Utilizamos os seguintes parâmetros:

$$\eta_0 = 0.05, \quad \eta_1 = 0.25, \quad \gamma = 0.3, \quad \gamma_{inc} = 2, \quad \Delta_0 = 1, \quad e \quad \bar{\Delta} = 80.$$

Tabela 4.3: *Resultados Numéricos - Teste 3*

P	n	m	tempo	iter.	solução	P	n	m	tempo	iter.	solução
1	2	2	1,08	114	$1,02 \times 10^{-15}$	23	4	5	11,19	206	$2,25 \times 10^{-5}$
2	2	2	0,95	107	48,98425	24	2	4	0,53	22	$8,07 \times 10^{-7}$
3	2	2	6,76	1000	$1,29 \times 10^{-1}$	24	4	8	1,37	16	$1,16 \times 10^{-5}$
4	2	3	6,70	1000	$9,81 \times 10^{11}$	25	2	4	0,58	25	$1,31 \times 10^{-19}$
5	2	3	0,57	33	$1,47 \times 10^{-10}$	25	5	7	5,52	48	$6,51 \times 10^{-19}$
6	2	10	0,49	31	124,36218	26	2	2	0,53	29	$9,38 \times 10^{-22}$
7	3	3	1,86	77	$1,17 \times 10^{-16}$	26	5	5	2,89	22	$3,39 \times 10^{-10}$
8	3	15	2,08	66	$8,21 \times 10^{-3}$	27	2	2	0,34	3	$6,41 \times 10^{-16}$
9	3	15	0,63	10	$1,14 \times 10^{-8}$	27	5	5	0,89	3	$1,52 \times 10^{-27}$
10	3	16	0,74	20	$6,46 \times 10^6$	28	2	2	0,55	25	$5,14 \times 10^{-23}$
11	3	20	26,27	1000	$2,88 \times 10^{-2}$	28	5	5	8,98	65	$3,22 \times 10^{-20}$
12	3	20	4,00	151	$3,59 \times 10^{-8}$	29	2	2	0,45	16	$1,24 \times 10^{-18}$
13	4	4	1,48	24	$5,58 \times 10^{-10}$	29	5	5	3,19	23	$3,56 \times 10^{-21}$
14	4	6	9,16	193	$2,28 \times 10^{-15}$	30	2	2	0,49	26	$2,79 \times 10^{-18}$
15	4	11	7,03	120	$3,08 \times 10^{-4}$	30	5	5	3,39	26	$1,93 \times 10^{-18}$
16	4	20	2,84	47	85822,20163	31	2	2	0,52	29	$2,41 \times 10^{-17}$
17	5	33	94,41	721	$7,71 \times 10^{-5}$	31	5	5	4,21	31	$8,41 \times 10^{-18}$
18	6	13	74,02	302	$2,17 \times 10^{-8}$	32	2	2	0,38	4	0
19	11	65	2751,33	871	$4,01 \times 10^{-2}$	32	5	5	1,30	5	$2,62 \times 10^{-29}$
20	6	31	267,20	995	$2,29 \times 10^{-3}$	33	2	2	0,40	5	0,2
20	9	31	463,90	656	$3,28 \times 10^{-5}$	33	5	5	108,26	926	0,90909
21	4	4	7,06	153	$2,34 \times 10^{-10}$	34	3	3	0,43	3	2
21	6	6	42,52	189	$7,92 \times 10^{-12}$	34	5	5	30,64	244	2,42857
22	8	8	47,49	68	$2,33 \times 10^{-10}$	35	2	2	0,48	19	$7,50 \times 10^{-11}$
23	2	3	0,55	23	$8,36 \times 10^{-6}$	35	5	5	2,80	24	$4,95 \times 10^{-10}$

A Tabela 4.3 exhibe os resultados para a bateria de testes 3. Entre os 50 problemas,

o algoritmo resolveu 46 dentro do critério de parada para o raio Δ_k . Dos 4 restantes, em 1 a solução foi satisfatória e 3 deles não foram resolvidos. O algoritmo novamente não encontrou solução antes de 1000 iterações para os problemas 3 e 4. No problema 10, apesar de o algoritmo terminar antes de 1000 iterações, não encontrou a solução. Para o problema 11 o algoritmo chegou próximo da solução mas atingiu o máximo de iterações.

Teste 4

Utilizamos os seguintes parâmetros:

$$\eta_0 = 0.05, \quad \eta_1 = 0.1, \quad \gamma = 0.3, \quad \gamma_{inc} = 2, \quad \Delta_0 = 0.5, \quad \text{e} \quad \bar{\Delta} = 50.$$

Tabela 4.4: *Resultados Numéricos - Teste 4*

P	n	m	tempo	iter.	solução	P	n	m	tempo	iter.	solução
1	2	2	0,92	93	$7,58 \times 10^{-15}$	23	4	5	12,79	243	$2,25 \times 10^{-5}$
2	2	2	0,91	100	48,98425	24	2	4	0,55	23	$8,07 \times 10^{-7}$
3	2	2	5,24	1000	$1,31 \times 10^{-1}$	24	4	8	1,61	22	$9,89 \times 10^{-6}$
4	2	3	6,15	1000	$9,81 \times 10^{11}$	25	2	4	0,55	23	$3,67 \times 10^{-20}$
5	2	3	0,70	49	$8,89 \times 10^{-18}$	25	5	7	5,54	45	$1,74 \times 10^{-13}$
6	2	10	0,57	35	124,36218	26	2	2	0,52	25	$2,42 \times 10^{-19}$
7	3	3	1,81	84	$1,88 \times 10^{-18}$	26	5	5	2,96	26	$6,31 \times 10^{-9}$
8	3	15	1,60	44	$8,21 \times 10^{-3}$	27	2	2	0,37	5	$2,46 \times 10^{-20}$
9	3	15	3,31	102	$1,13 \times 10^{-8}$	27	5	5	1,07	4	$3,03 \times 10^{-28}$
10	3	16	2,51	115	$5,83 \times 10^6$	28	2	2	0,50	19	$8,31 \times 10^{-19}$
11	3	20	25,19	1000	$2,92 \times 10^{-2}$	28	5	5	6,79	49	$1,12 \times 10^{-20}$
12	3	20	5,29	213	$1,32 \times 10^{-8}$	29	2	2	0,46	19	$1,78 \times 10^{-19}$
13	4	4	2,65	39	$1,76 \times 10^{-8}$	29	5	5	3,42	22	$2,23 \times 10^{-21}$
14	4	6	7,38	149	$2,98 \times 10^{-11}$	30	2	2	0,51	28	$2,11 \times 10^{-19}$
15	4	11	8,62	152	$3,08 \times 10^{-4}$	30	5	5	3,98	27	$5,98 \times 10^{-17}$
16	4	20	2,45	41	85822,20163	31	2	2	0,52	27	$9,75 \times 10^{-17}$
17	5	33	92,00	684	$7,68 \times 10^{-5}$	31	5	5	4,13	34	$4,63 \times 10^{-16}$
18	6	13	75,55	321	$2,94 \times 10^{-7}$	32	2	2	0,40	5	$4,93 \times 10^{-32}$
19	11	65	2868,61	901	$4,01 \times 10^{-2}$	32	5	5	1,30	5	$1,41 \times 10^{-29}$
20	6	31	265,00	1000	$2,29 \times 10^{-3}$	33	2	2	0,42	5	0,2
20	9	31	704,95	1000	$3,55 \times 10^{-5}$	33	5	5	69,43	583	0,90909
21	4	4	7,55	124	$7,34 \times 10^{-9}$	34	3	3	0,49	5	2
21	6	6	44,58	197	$8,89 \times 10^{-13}$	34	5	5	105,74	894	2,42857
22	8	8	46,33	71	$1,39 \times 10^{-11}$	35	2	2	0,50	20	$6,66 \times 10^{-11}$
23	2	3	0,52	22	$2,38 \times 10^{-5}$	35	5	5	3,34	32	$5,38 \times 10^{-10}$

A Tabela 4.4 exhibe os resultados para a bateria de testes 4. Entre os 50 problemas, o algoritmo resolveu 44 dentro do critério de parada para o raio Δ_k . Dos 6 restantes, em

3 a solução foi satisfatória. O algoritmo novamente não encontrou solução antes de 1000 iterações para os problemas 3 e 4 e, para o problema 10, apesar de terminar antes das 1000 iterações, não encontrou a solução. Para os problemas 11 e 20 o algoritmo encontrou valores próximo das soluções.

Teste 5

Utilizamos os seguintes parâmetros:

$$\eta_0 = 0.5, \quad \eta_1 = 0.9, \quad \gamma = 0.5, \quad \gamma_{inc} = 3, \quad \Delta_0 = 0.5, \quad e \quad \bar{\Delta} = 50.$$

Tabela 4.5: *Resultados Numéricos - Teste 5*

P	n	m	tempo	iter.	solução	P	n	m	tempo	iter.	solução
1	2	2	1,28	145	$8,38 \times 10^{-15}$	23	4	5	8,11	140	$2,26 \times 10^{-5}$
2	2	2	1,54	190	48,98425	24	2	4	0,58	23	$8,07 \times 10^{-7}$
3	2	2	6,33	1000	$1,31 \times 10^{-1}$	24	4	8	2,09	36	$9,41 \times 10^{-6}$
4	2	3	6,68	1000	$9,83 \times 10^{11}$	25	2	4	0,58	27	$9,63 \times 10^{-22}$
5	2	3	0,60	43	$3,03 \times 10^{-11}$	25	5	7	5,58	40	$8,46 \times 10^{-17}$
6	2	10	0,71	58	124,36218	26	2	2	0,58	28	$3,11 \times 10^{-15}$
7	3	3	1,45	53	$6,92 \times 10^{-17}$	26	5	5	3,02	24	$9,21 \times 10^{-9}$
8	3	15	1,91	57	$8,21 \times 10^{-3}$	27	2	2	0,38	4	$5,05 \times 10^{-29}$
9	3	15	0,70	9	$1,22 \times 10^{-8}$	27	5	5	0,88	3	$1,13 \times 10^{-27}$
10	3	16	0,68	19	$6,69 \times 10^6$	28	2	2	0,68	41	$3,65 \times 10^{-20}$
11	3	20	24,91	1000	$3,13 \times 10^{-2}$	28	5	5	4,57	31	$1,98 \times 10^{-15}$
12	3	20	5,24	205	$2,04 \times 10^{-6}$	29	2	2	0,59	29	$5,27 \times 10^{-19}$
13	4	4	2,85	50	$9,73 \times 10^{-8}$	29	5	5	6,47	49	$5,96 \times 10^{-23}$
14	4	6	15,47	312	$5,25 \times 10^{-16}$	30	2	2	0,58	38	$3,02 \times 10^{-17}$
15	4	11	10,30	183	$3,08 \times 10^{-4}$	30	5	5	4,49	32	$3,37 \times 10^{-17}$
16	4	20	3,79	71	85822,20163	31	2	2	0,65	43	$1,17 \times 10^{-19}$
17	5	33	130,78	976	$7,67 \times 10^{-5}$	31	5	5	6,10	46	$3,95 \times 10^{-19}$
18	6	13	106,42	447	$5,38 \times 10^{-6}$	32	2	2	0,39	4	$1,97 \times 10^{-31}$
19	11	65	3181,14	1000	$4,02 \times 10^{-2}$	32	5	5	1,16	4	$1,23 \times 10^{-29}$
20	6	31	264,61	1000	$2,38 \times 10^{-3}$	33	2	2	0,43	6	0,2
20	9	31	709,95	1000	$3,58 \times 10^{-5}$	33	5	5	122,64	1000	0,90909
21	4	4	10,93	198	$1,95 \times 10^{-9}$	34	3	3	0,50	4	2
21	6	6	61,48	268	$1,89 \times 10^{-9}$	34	5	5	92,57	780	2,42857
22	8	8	77,15	108	$7,66 \times 10^{-8}$	35	2	2	0,61	36	$3,38 \times 10^{-8}$
23	2	3	0,57	28	$1,04 \times 10^{-5}$	35	5	5	3,83	36	$4,17 \times 10^{-10}$

A Tabela 4.5 exibe os resultados para a bateria de testes 5. Entre os 50 problemas, o algoritmo resolveu 42 dentro do critério de parada para o raio Δ_k . Dos 8 restantes, 3 deles não foram resolvidos. O algoritmo não encontrou solução para os problemas 3, 4 e 10. Para os problemas 11, 19, 20 e 33, o algoritmo com esta configuração chegou bem próximo das soluções.

4.2.2 Perfil de desempenho

Em 2002, Dolan e Moré introduziram em [8] o conceito de perfil de desempenho (*performance profile*), uma ferramenta para comparar o desempenho de N_s métodos de um conjunto S quando aplicado para resolver N_p problemas de um conjunto P , usando algumas medidas como o número de iterações ou o número de avaliações de função.

Em [20] Sakamori e Gomes-Ruggiero utilizam este conceito e fazem uma explicação, em português, sobre o perfil de desempenho.

Para comparar diferentes métodos, precisamos calcular uma taxa de desempenho $r_{s,p}$ em relação a uma medida \hat{m} , que pode ser quantificada como a quantidade da medida \hat{m} que o método s usou para resolver o problema p , tal quantidade chamamos de $m_{s,p}$. E calculamos a taxa de desempenho para cada problema p e método s como sendo $r_{s,p} = m_{s,p} / \min\{m_{s,p} \mid s \in S\}$, se o problema p é resolvido pelo método s e, caso não seja resolvido, $r_{s,p} = r_M$ onde r_M , é um parâmetro fixado suficientemente grande.

Em seguida, para cada $s \in S$, construímos a função de distribuição acumulada $\rho_s : \mathbb{R}^n \rightarrow [0, 1]$, para a taxa de desempenho tal que $\rho_s(\alpha) = (1/N_p) \text{card}\{p \in P \mid r_{s,p} \leq \alpha\}$. Tal função representa o desempenho do método s . Para analisá-lo, os valores $\rho_s(1)$ e $\bar{\alpha}$, tal que $\rho_s(\bar{\alpha}) = 1$, fornecem informações importantes. Seja \bar{s} o método que dá o valor máximo para $\rho_s(1)$. Este método pode resolver o maior número de problemas usando o menor número da medida \hat{m} . A eficiência do método s em termos do número de problemas que foram resolvidos é avaliada pelo menor valor de α , denotado por $\bar{\alpha}_s$, tal que $\rho_s(\bar{\alpha}_s) = 1$, se existe tal valor para $\alpha < r_M$. Portanto, o melhor método em termos de eficiência será o método \hat{s} para o qual $\bar{\alpha}_{\hat{s}} = \min\{\bar{\alpha}_s, \forall s \in S\}$.

Apresentamos na Figura 4.1 o perfil de desempenho do Algoritmo 4.1, resultado da bateria de testes para calibrar os parâmetros que apresentam o melhor desempenho. A figura mostra o perfil de desempenho referente ao número de avaliações de função, que estão apresentados na Tabela 4.6.

Pelo número de avaliações da função o conjunto de parâmetros do Teste 3 é que apresentou melhor desempenho, resolvendo 49% dos problemas com o menor número de avaliações. Em segundo, o algoritmo com a configuração do Teste 4 resolveu 30% com o menor número de avaliações; o algoritmo com os parâmetros escolhidos nos Teste 1, 2 e 5 resolveu 13%, 9% e 7% dos problemas com o menor número de iterações, respectivamente.

Para 90% dos problemas a configuração do Teste 3 não utilizou mais que 1,2 vezes o menor número de avaliações da função. A configuração do Teste 4 resolve 90% dos

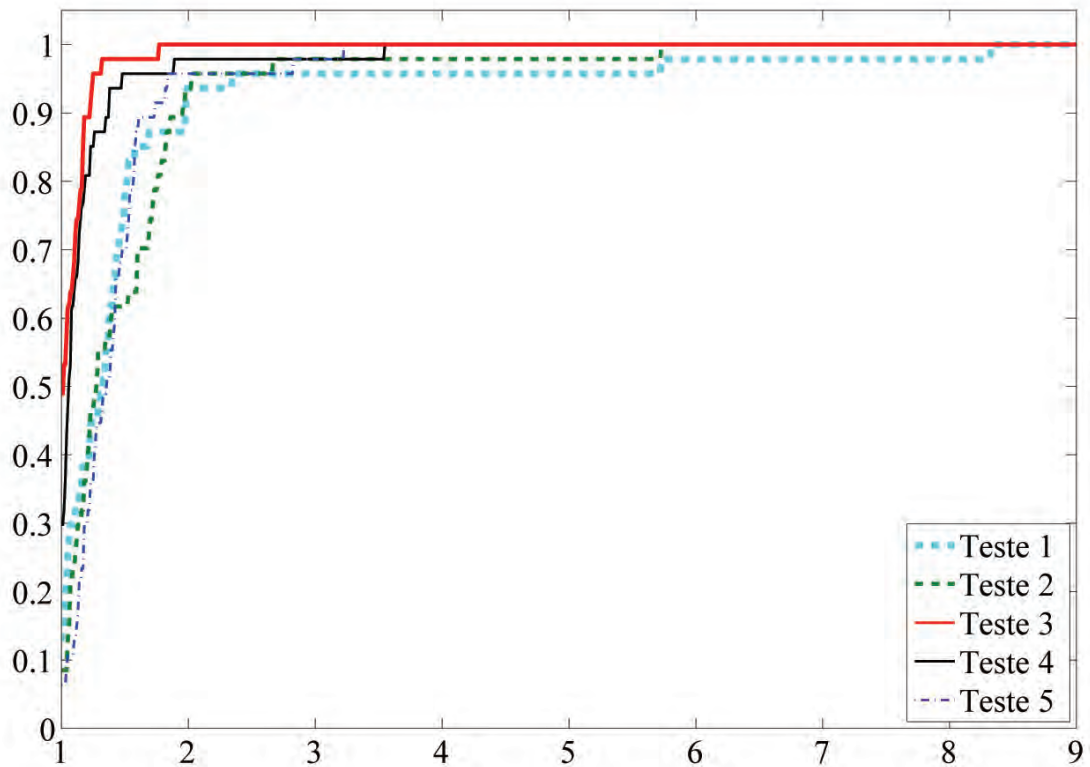


Figura 4.1: Perfil de Desempenho - Número de Avaliações da função.

problemas usando não mais que 1,4 vezes o menor número de avaliações. Enquanto as configurações do Teste 5 resolve 90% dos problemas levando 1,75 vezes o menor número de avaliações. Já os Testes 1 e 2 resolvem 90% dos problemas levando 1,95 vezes o menor número de avaliações da função.

Podemos considerar que o algoritmo com a configuração de parâmetros dada pelo Teste 3 é o mais eficiente entre as configurações testadas. Permitindo gastar até 1,8 vezes o menor número de avaliações da função a configuração de parâmetros dada pelo Teste 3 resolve todos dos problemas que foram resolvidos por alguma configuração, sendo que com os parâmetros do Teste 4 o algoritmo resolve 96%; e com os parâmetros do Teste 5 resolve 92% dos problemas. As configurações dos Teste 1 e 2 resolvem 87% e 83% dos problemas com até 1,8 vezes o menor número de avaliações da função.

A configuração de parâmetros do Teste 3 apresentou melhor desempenho em relação ao número de avaliações da função. Por esse motivo decidimos investigar a relação da precisão do Algoritmo 3.3 para maximizar os polinômios de Lagrange em $B(x_k, \Delta_k)$ com esta configuração de parâmetros. Para isto, fixamos os parâmetros com os valores do Teste 3 e alteramos apenas a quantidade t de pontos aleatórios do conjunto de amostra. Os novos testes, são apresentados abaixo.

Tabela 4.6: *Número de Avaliações da Função Objetivo*

P	n	T1	T2	T3	T4	T5	P	n	T1	T2	T3	T4	T5
1	2	884	816	728	750	1038	24	4	1349	977	673	575	677
2	2	1046	950	742	822	1286	25	2	354	318	288	248	354
5	2	390	416	322	260	400	25	5	1993	1827	1681	1522	2060
6	2	406	374	206	214	374	26	2	528	412	346	464	480
7	3	758	950	944	900	1158	26	5	1108	1248	901	924	1087
8	3	1090	1078	932	980	1170	27	2	208	208	196	204	250
9	3	922	858	812	870	954	27	5	718	741	720	699	982
11	3	-	12010	12010	12010	12010	28	2	616	580	358	312	498
12	3	1854	2310	1350	2550	3812	28	5	1154	1428	922	782	1035
13	4	754	869	718	818	1058	29	2	458	552	366	314	498
14	4	5046	4079	3758	4619	5473	29	5	1598	2120	1247	1071	1523
15	4	2641	4294	2660	2541	3121	30	2	472	536	336	314	498
16	4	1069	831	746	763	848	30	5	1541	1809	1014	1146	1401
17	5	-	-	17754	17478	23021	31	2	500	558	404	366	414
18	6	10444	11490	10352	12156	15694	31	5	2288	2261	1154	1198	1823
19	11	68296	80078	69198	73414	80078	32	2	228	222	230	238	250
20	6	30028	30028	28806	30028	30028	32	5	718	762	764	787	785
20	9	57055	57055	46663	57055	57055	33	2	228	228	218	222	226
21	4	2986	3521	3252	3456	3446	33	5	44000	27963	31616	38525	43979
21	6	7570	6548	6820	7508	10220	34	3	462	682	368	422	336
22	8	5170	5948	4662	6856	8550	34	5	12403	33097	16282	43858	39979
23	2	282	294	210	226	322	35	2	568	584	378	338	428
23	4	5089	6861	6209	5741	6388	35	5	1448	1799	1988	1552	1127
24	2	436	402	328	350	366							

A Tabela 4.7 apresenta a comparação entre o número de avaliações da função dos Testes 3, 6 e 7. Usamos $t = 20n$ em T6 e $t = 30n$ em T7 apenas nos problemas em que a configuração Teste 3 apresentou resultados satisfatórios, ou seja, exceto os problemas 3, 4 e 10. Mesmo aumentando o número de pontos da amostra, o algoritmo não resolveu estes problemas.

A Tabela 4.8 apresenta a comparação do tempo computacional, dado em segundos, entre os Testes 3, 6 e 7.

Usando o número de avaliações da função como comparação entre os Testes 3, 6 e 7 podemos notar que quando $t = 30n$ o algoritmo apresenta ligeira vantagem, como podemos ver no perfil de desempenho apresentado pela Figura 4.2. No entanto, como mais pontos estão sendo utilizados, cada iteração é computacionalmente mais cara. Por esse motivo resolvemos fazer também a comparação em relação ao tempo computacional. A Figura 4.3 mostra a comparação entre os tempos dos Testes 3, 6 e 7.

Como esperado, o Teste 3 apresentou menor tempo para resolver a maioria dos problemas, sendo que resolveu 83% dos problemas com o menor tempo. O Teste 6 resolveu 17% dos problemas com o menor tempo e o Teste 7 não resolveu nenhum problema com

Tabela 4.7: *Resultados Numéricos - Número de Iterações T3, T6 e T7*

P	n	m	T3	T6	T7	P	n	m	T3	T6	T7
1	2	2	728	748	810	24	4	8	673	574	610
2	2	2	742	774	774	25	2	4	288	358	384
5	2	3	322	402	378	25	5	7	1681	1836	1533
6	2	10	206	286	350	26	2	2	346	334	348
7	3	3	944	972	958	26	5	5	901	924	1018
8	3	15	932	846	908	27	2	2	196	212	196
9	3	15	812	366	330	27	5	5	720	720	720
11	3	20	12010	12010	12010	28	2	2	358	310	316
12	3	20	1350	1422	1156	28	5	5	922	922	832
13	4	4	718	821	825	29	2	2	366	366	300
14	4	6	3758	4418	4485	29	5	5	1247	1048	1113
15	4	11	2660	2478	2746	30	2	2	336	330	360
16	4	20	746	950	899	30	5	5	1014	1125	1129
17	5	33	17754	14580	15569	31	2	2	404	382	370
18	6	13	10352	10302	9946	31	5	5	1154	1269	1131
19	11	65	69198	62958	61560	32	2	2	230	210	210
20	6	31	28806	27446	27024	32	5	5	764	764	764
20	9	31	46663	43190	40664	33	2	2	218	270	250
21	4	4	3252	3041	2797	33	5	5	31616	43979	40603
21	6	6	6820	5756	6210	34	3	3	368	368	368
22	8	8	4662	5112	4448	34	5	5	16282	43979	30835
23	2	3	210	254	270	35	2	2	378	360	366
23	4	5	6209	5430	4348	35	5	5	1988	957	991
24	2	4	328	336	314						

o menor tempo. O Teste 3 consegue resolver todos os problemas gastando não mais que o dobro melhor tempo.

Acreditamos que, quanto ao tempo computacional, a melhor configuração para o algoritmo proposto entre os testes realizados é a escolha dos parâmetros do Teste 3

$$\eta_0 = 0.05, \quad \eta_1 = 0.25, \quad \gamma = 0.3, \quad \gamma_{inc} = 2, \quad \Delta_0 = 1, \quad e \quad \bar{\Delta} = 80.$$

Quanto à quantidade de pontos no conjunto de amostra, um menor número apresenta ganho no tempo computacional enquanto o número de avaliações da função não apresenta grandes mudanças ao aumentar a precisão do conjunto de amostra. A quantidade mínima de pontos no conjunto de amostra deve ser maior que a quantidade de pontos necessária na interpolação polinomial $p = (n + 1)(n + 2)/2$. Temos ainda que, se este número é muito pequeno a quantidade de iterações aumenta consideravelmente e, conseqüentemente, o tempo computacional. Se este número é muito grande, o tempo computacional é inviável e não obtemos considerável qualidade nos modelos. Tais fatores devem ser considerados, além do tipo de problema a ser resolvido. Quando a avaliação da função é cara, um

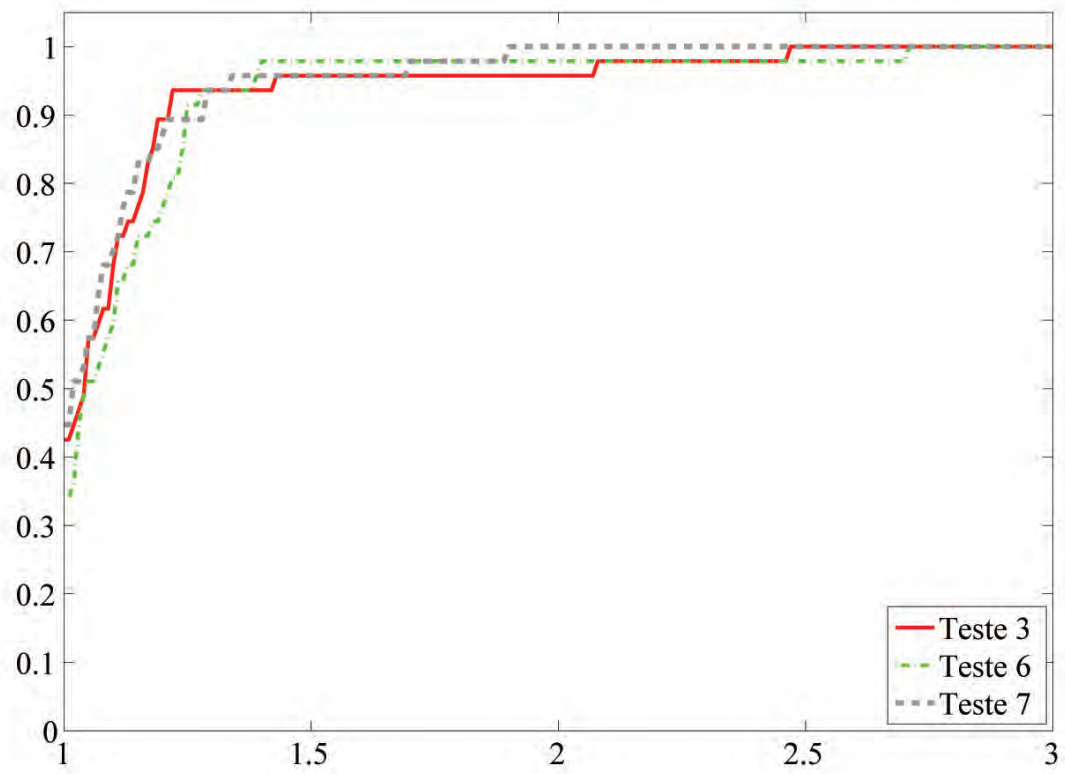


Figura 4.2: Perfil de Desempenho - Avaliações da função.

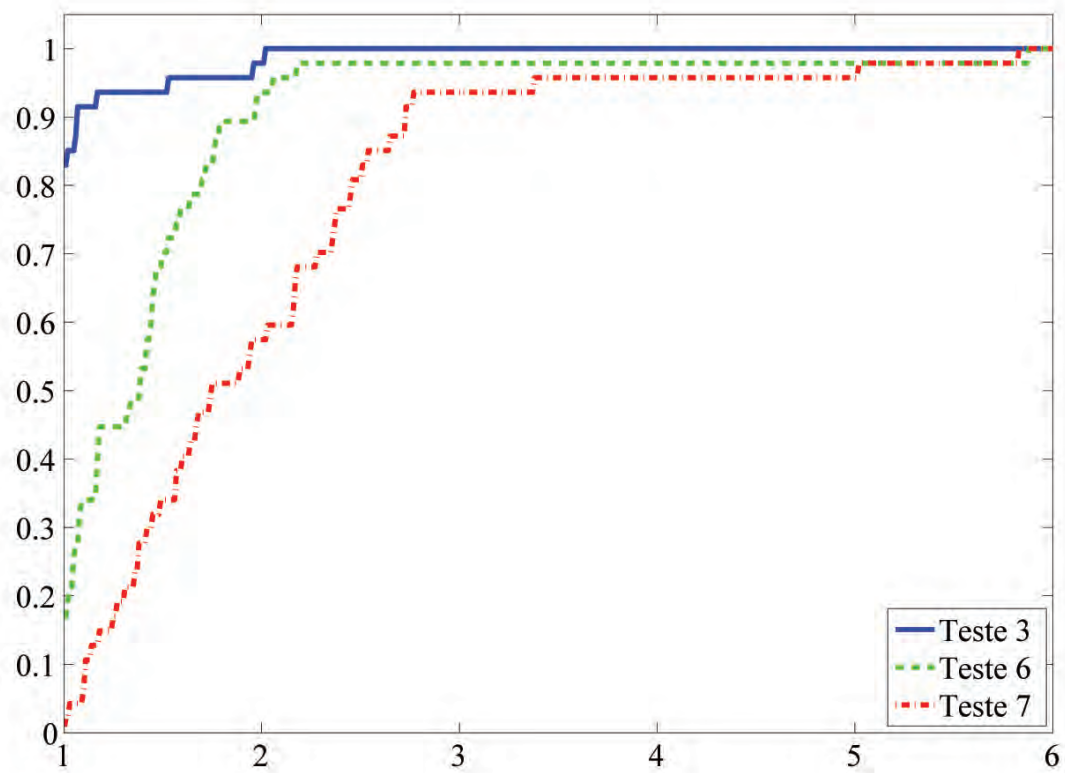


Figura 4.3: Perfil de Desempenho - Tempo Computacional.

Tabela 4.8: *Resultados Numéricos - Tempo Computacional - T3, T6 e T7*

P	n	m	T3	T6	T7		P	n	m	T3	T6	T7
1	2	2	1,07	1,09	1,59		24	4	8	1,36	1,95	4,61
2	2	2	0,94	1,36	1,65		25	2	4	0,57	0,54	0,60
5	2	3	0,56	0,75	0,88		25	5	7	5,52	8,27	11,87
6	2	10	0,49	0,57	0,82		26	2	2	0,52	0,55	0,59
7	3	3	1,86	2,70	3,60		26	5	5	2,89	5,66	7,65
8	3	15	2,08	1,36	3,45		27	2	2	0,33	0,34	0,36
9	3	15	0,63	0,83	1,03		27	5	5	0,88	1,50	2,10
11	3	20	26,26	37,25	45,69		28	2	2	0,55	0,52	0,57
12	3	20	3,99	4,34	6,70		28	5	5	8,98	4,58	5,70
13	4	4	1,48	2,92	3,63		29	2	2	0,44	0,47	0,56
14	4	6	9,16	19,89	24,97		29	5	5	3,18	5,06	6,18
15	4	11	7,02	9,76	19,13		30	2	2	0,48	0,57	0,70
16	4	20	2,84	4,02	6,14		30	5	5	3,39	6,06	8,49
17	5	33	94,41	46,83	74,55		31	2	2	0,52	0,60	0,71
18	6	13	74,01	102,44	204,51		31	5	5	4,20	7,40	9,59
19	11	65	2751,33	2591,85	3556,97		32	2	2	0,38	0,37	0,37
20	6	31	267,20	312,81	540,60		32	5	5	1,29	1,90	2,44
20	9	31	463,89	724,52	2324,86		33	2	2	0,39	0,42	0,46
21	4	4	7,05	11,54	17,28		33	5	5	100,26	185,64	256,26
21	6	6	42,51	64,81	91,90		34	3	3	0,43	0,37	0,37
22	8	8	47,46	83,76	111,59		34	5	5	30,64	180,03	178,43
23	2	3	0,55	0,59	0,75		35	2	2	0,47	0,50	0,67
23	4	5	11,18	16,25	17,55		35	5	5	2,79	5,74	6,09
24	2	4	0,53	0,61	0,69							

menor número de iterações é desejável, e assim escolhemos mais pontos no conjunto de amostra. Quando avaliar a função não é computacionalmente custoso, podemos escolher menos pontos no conjunto de amostra e ganhar tempo computacional.

Existem na literatura algumas implementações de métodos de região de confiança baseados em interpolação polinomial que não usam derivadas da função objetivo para resolver problemas irrestritos. Conn, Scheinberg e Vicente [5, Apêndice 1] fazem uma relação de softwares disponíveis na internet, tais como CONDOR, DFO, WEDGE, UOBYQA e NEWUOA. Como o foco principal deste trabalho foi a análise de convergência e o estudo da geometria dos conjuntos de interpolação não tivemos qualquer pretensão em realizar testes computacionais comparativos com estes softwares profissionais e robustos.

Conclusões

Nosso interesse neste trabalho foi estudar um método para otimização irrestrita que não faz uso de derivadas. Utilizamos para isso um método de região de confiança com modelos baseados em interpolação polinomial quadrática completa. Primeiramente estudamos os métodos de região de confiança no sentido clássico, ainda com o uso de derivadas. Em seguida estudamos as adaptações propostas por Conn, Scheinberg e Vicente [5] para um método de região de confiança sem derivadas e as condições necessárias a serem satisfeitas pelos modelos da função objetivo a serem considerados em cada iteração do algoritmo.

A partir disso, dedicamos parte significativa dos estudos na geometria do conjunto de interpolação; nas condições que os modelos devem satisfazer, como ser plenamente linear ou plenamente quadrático, e na construção de modelos baseados em interpolação. Sob certos aspectos, tais modelos têm comportamento local semelhante às aproximações de Taylor da função. Mostramos que conjuntos bem posicionados geram modelos com boa precisão, além disso discutimos como medir e melhorar o posicionamento de um conjunto de pontos de interpolação. Provamos resultados de convergência global para um método de região de confiança sem derivadas baseado em interpolação polinomial. Os resultados teóricos obtidos, foram apresentados em forma de poster no evento CWB 2010 - II Congresso de Matemática e suas Aplicações, realizado em dezembro de 2010 em Curitiba.

Para validarmos o método, apresentamos testes computacionais que mostraram algumas limitações, principalmente ao que tange o número de variáveis do problema, uma vez que quando trabalhamos com interpolação polinomial completa, a quantidade de variáveis cresce bastante com o número de variáveis da função objetivo. Resolvemos 50 problemas da coletânea de Moré, Garbow e Hillstom [13] em dimensão pequena, ou seja, inferior a 11. Os problemas foram resolvidos através do algoritmo proposto neste trabalho com diferentes escolhas de parâmetros. Dos problemas testados, apenas 3 deles não foram resolvidos com qualquer uma das escolhas de parâmetro.

Acreditamos, por fim, que a atualização do modelo no algoritmo possa ser melhorada

em trabalhos futuros, onde também poderemos otimizar os parâmetros para melhores resultados.

Referências Bibliográficas

- [1] P. G. Ciarlet e P. A. Raviart, *General Lagrange and Hermite interpolation in \mathbb{R}^n with applications to finite element methods*, Archive for Rational Mechanics and Analysis 46 (1972), 177–199.
- [2] A. R. Conn, N. I. M. Gould e Ph. L. Toint, *Trust-region methods*, MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2000.
- [3] A. R. Conn, K. Scheinberg e L. N. Vicente, *Geometry of interpolation sets in derivative free optimization*, Mathematical Programming 111 (2008), 141–172.
- [4] A. R. Conn, K. Scheinberg e L. N. Vicente, *Global convergence of general derivative-free trust-region algorithms to first and second order critical points*, SIAM Journal on Optimization 20 (2009), 387–415.
- [5] A. R. Conn, K. Scheinberg e L. N. Vicente, *Introduction to derivative-free optimization*, MPS-SIAM Series on Optimization, Philadelphia, 2009.
- [6] J. E. Dennis, Jr. e R. B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*, Prentice-Hall, Englewood Cliffs, New Jersey, 1983, Republicado por SIAM: Classics in Applied Mathematics, Philadelphia, 1996.
- [7] M. A. Diniz-Ehrhardt, J. M. Martínez e L. G. Pedroso, *Derivative-free methods for nonlinear programming with general lower-level constraints*, Computational & Applied Mathematics (2011), to appear.
- [8] E. D. Dolan e J. J. Moré, *Benchmarking optimization software with performance profiles*, Mathematical Programming 91 (2002), 201–213.
- [9] A. Izmailov e M. Solodov, *Otimização: Condições de otimalidade, elementos de análise convexa e dualidade*, vol. 1, IMPA, Rio de Janeiro, 2005.
- [10] A. Izmailov e M. Solodov, *Otimização: Métodos computacionais*, vol. 2, IMPA, Rio de Janeiro, 2007.
- [11] S. J. Leon, *Álgebra Linear com aplicações*, LTC, Rio de Janeiro, 1999.

- [12] V. L. da R. Lopes, M. A. Diniz-Ehrhardt e L. G. Pedroso, *Métodos sem derivadas para minimização irrestrita*, CWB2010 - II Congresso de Matemática e suas Aplicações, Curitiba, 2010.
- [13] J. J. Moré, B. S. Garbow e K. E. Hillstrome, *Testing unconstrained optimization software*, ACM Transactions on Mathematical Software 7 (1981), no. 1, 17–41.
- [14] J. Nocedal e S. J. Wright, *Numerical optimization*, Springer Series in Operations Research, Springer-Verlag, 1999.
- [15] L. G. Pedroso, *Programação não linear sem derivadas*, Tese de Doutorado, Instituto de Matemática, Estatística e Computação Científica, UNICAMP, 2009.
- [16] M. J. D. Powell, *The NEWUOA software for unconstrained optimization without derivatives*, Large-Scale Nonlinear Optimization, Springer, New York, 2006, 255–297.
- [17] M. J. D. Powell, *Developments of NEWUOA for minimization without derivatives*, IMA Journal of Numerical Analysis (2008), Vol 28, 649–664.
- [18] M. J. D. Powell, *The BOBYQA algorithm for bound constrained optimization without derivatives*, Department of Applied Mathematics and Theoretical Physics, Cambridge, England, technical report NA2009/06, 2009.
- [19] A. A. Ribeiro e E. W. Karas, *Um curso de otimização*, Universidade Federal do Paraná - UFPR, Curitiba, 2010, Disponível em: <http://people.ufpr.br/~ademir.ribeiro/ensino/livro/livro.pdf>. Acessado em: 08 ago 2010.
- [20] M. Sakamori e M. A. Gomes-Ruggiero, *Variações do método de máxima descida em otimização irrestrita*, TEMA - Tendências em Matemática Aplicada e Computacional 6 (2005), no. 2, 285–294.