

UNIVERSIDADE ESTADUAL DE MARINGÁ  
CENTRO DE CIÊNCIAS EXATAS  
DEPARTAMENTO DE MATEMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM MATEMÁTICA  
(Mestrado)

ANDERSON ERVINO SCHWERTNER

O Método de Levenberg-Marquardt para problemas de  
Otimização de Menor Valor Ordenado

Maringá-PR

2019



Anderson Ervino Schwertner

# O Método de Levenberg-Marquardt para problemas de Otimização de Menor Valor Ordenado

Dissertação apresentada ao Programa de Pós-Graduação em Matemática do Departamento de Matemática, Centro de Ciências Exatas da Universidade Estadual de Maringá, como requisito para obtenção do título de Mestre em Matemática.

Área de concentração: Matemática Aplicada.

Orientador: Dr. Emerson Vitor Castelani

Coorientador: Dr. Francisco Nogueira Calmon Sobral

Maringá

2019

Dados Internacionais de Catalogação na Publicação (CIP)  
(Biblioteca Setorial BSE-DMA-UEM, Maringá, PR, Brasil)

S415m Schwertner, Anderson Ervino  
O método de Levenberg-Marquardt para problemas de  
otimização de menor valor ordenado / Anderson Ervino  
Schwertner. -- Maringá, 2019.  
xx 100 f. : il., tabs., color.

Orientador: Prof. Dr. Emerson Vitor Castelani.  
Coorientador: Prof. Dr. Francisco Nogueira Calmon  
Sobral.

Dissertação (mestrado) - Universidade Estadual de  
Maringá, Centro de Ciências Exatas, Programa de Pós-  
Graduação em Matemática - Área de Concentração:  
Matemática Aplicada, 2019.

1. Método de Levenberg-Marquardt. 2. Otimização  
de menor valor ordenado. 3. Quadrados mínimos não  
linear. 4. Levenberg-Marquardt Method. 5. Low order-  
value optimization. 6. Non-linear least squares. I.  
Castelani, Emerson Vitor, orient. II. Sobral,  
Francisco Nogueira Calmon, orient. III. Universidade  
Estadual de Maringá. Centro de Ciências Exatas.  
Programa de Pós-Graduação em Matemática - Área de  
Concentração: Matemática Aplicada. IV. Título.

CDD 22.ed. 519.6

Edilson Damasio CRB9-1.123

ANDERSON ERVINO SCHWERTNER

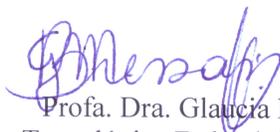
**O MÉTODO DE LEVENBERG-MARQUARDT PARA PROBLEMAS DE  
OTIMIZAÇÃO DE MENOR VALOR ORDENADO**

Dissertação apresentada ao Programa de Pós-Graduação em Matemática do Departamento de Matemática, Centro de Ciências Exatas da Universidade Estadual de Maringá, como parte dos requisitos necessários para a obtenção do título de Mestre em Matemática tendo a Comissão Julgadora composta pelos membros:

COMISSÃO JULGADORA:



Prof. Dr. Emerson Vitor Castelani  
DMA/Universidade Estadual de Maringá (Presidente)



Profª. Dra. Gláucia Maria Bressan  
Universidade Tecnológica Federal do Paraná – Cornélio Procópio



Prof. Dr. Emerson Luiz do Monte Carmelo  
Universidade Estadual de Maringá

Aprovada em 25 de fevereiro de 2019.

Local de defesa: Sala 107, Bloco F67, campus da Universidade Estadual de Maringá.



Dedico esta dissertação à minha família, em especial aos meus pais, Traudi e Otmar, os quais me fizeram entender, ainda pequeno, que o maior bem que se possa possuir é o conhecimento.



---

---

# AGRADECIMENTOS

---

Agradeço primeiramente à Deus por me guiar e auxiliar na busca de meus sonhos, pelo seu cuidado e amor por mim e minha família.

À minha amada esposa, pela paciência e carinho, por estar sempre ao meu lado e me incentivar.

Aos professores Emerson Vitor Castelani e Francisco Nogueira Calmon Sobral, por aceitaram me orientar, pela dedicação, companheirismo e apoio essenciais para a realização deste trabalho.

Aos demais professores, funcionários e colegas do Programa de Pós-Graduação em Matemática, e todos aqueles que contribuíram para que este percurso pudesse ser concluído.

À CAPES, pelo apoio financeiro (Código de Financiamento 001).



“É somente nas misteriosas equações do amor  
que qualquer lógica ou razão pode ser encontrada.”

John F. Nash Jr.



---

---

# RESUMO

---

Neste trabalho apresentamos uma breve revisão de alguns conceitos importantes acerca do problema de Quadrados Mínimos, do Método de Levenberg-Marquardt e do problema de Otimização de Menor Valor Ordenado (LOVO). Posteriormente, abordamos uma nova estratégia para solução de problemas LOVO irrestritos com convergência para pontos fracamente críticos, através de uma variação do método clássico de Levenberg-Marquardt. São apresentados resultados relevantes acerca da convergência do método. Além das contribuições já citadas, são sugeridos dois novos parâmetros de *damping* e uma estratégia híbrida. Ao final da dissertação são apresentados dados sobre o desempenho do novo método, em comparação com uma variação do Método de Gauss-Newton para problemas LOVO, já conhecido na literatura.

**Palavras-chave:** Método de Levenberg-Marquardt, Otimização de Menor Valor Ordenado, Quadrados Mínimos Não Linear.



---

---

# ABSTRACT

---

In this work, we present a brief review of some important concepts about Least Squares problems, Levenberg-Marquardt method and the Low Order-Value Optimization problem (LOVO). Subsequently, we develop a new strategy for solving unconstrained LOVO problems with convergence to weakly critical points, through a variation of the classical Levenberg-Marquardt method. Relevant results are presented on the convergence of the method. In addition, two new damping parameters are suggested, also a hybrid strategy. At the end of the dissertation, the new method is compared to a variation of the Gauss-Newton Method for LOVO problems, already known in the literature.

**Keywords:** Levenberg-Marquardt Method, Low Order-Value Optimization, Non-linear Least Squares.



---

---

## LISTA DE FIGURAS

---

2.1	Polinômios de Maclaurin de graus 1, 3, 5 e 7 da função seno. . . . .	13
3.1	Solução do problema de quadrados mínimos linear, caso onde $b \in Im(A)$ . . .	28
3.2	Solução do problema de quadrados mínimos linear, caso onde $b \notin Im(A)$ . . .	29
5.1	Gráfico da função $f_{min}$ para $m = 3$ funções reais arbitrárias. . . . .	50

---

---

## LISTA DE TABELAS

---

4.1	Parâmetros de <i>damping</i> presentes na literatura. . . . .	42
4.2	Parâmetros de <i>damping</i> propostos. . . . .	42
4.3	Parâmetros de <i>damping</i> híbridos. . . . .	43
7.1	Parâmetros de <i>damping</i> empregados nos testes numéricos. . . . .	69
7.2	Modelos utilizados para os testes. . . . .	70
7.3	Convergência dos métodos de Gauss-Newton (GN) e Levenberg-Marquardt (LM) com <i>damping</i> PD7, para o teste Exponencial com 1000 pontos. . . . .	73
7.4	Desempenho dos algoritmos para o modelo Polinomial de Grau 1 para 100 pontos, com ponto inicial $x_0 = (0.0, 0.0)$ . . . . .	76
7.5	Desempenho dos algoritmos para o modelo Polinomial de Grau 1 para 1000 pontos, com ponto inicial $x_0 = (0.0, 0.0)$ . . . . .	77
7.6	Desempenho dos algoritmos para o modelo Polinomial de Grau 1 para 5000 pontos, com ponto inicial $x_0 = (0.0, 0.0)$ . . . . .	78
7.7	Desempenho dos algoritmos para o modelo Polinomial de Grau 3 para 100 pontos, com ponto inicial $x_0 = (0.0, 0.0, 0.0, 0.0)$ . . . . .	79
7.8	Desempenho dos algoritmos para o modelo Polinomial de Grau 3 para 1000 pontos, com ponto inicial $x_0 = (0.0, 0.0, 0.0, 0.0)$ . . . . .	80
7.9	Desempenho dos algoritmos para o modelo Polinomial de Grau 3 para 5000 pontos, com ponto inicial $x_0 = (0.0, 0.0, 0.0, 0.0)$ . . . . .	81
7.10	Desempenho dos algoritmos para o modelo Exponencial para 100 pontos, com ponto inicial $x_0 = (0.0, 0.0, 0.0, 0.0)$ . . . . .	82

7.11	Desempenho dos algoritmos para o modelo Exponencial para 1000 pontos, com ponto inicial $x_0 = (0.0, 0.0, 0.0, 0.0)$ . . . . .	83
7.12	Desempenho dos algoritmos para o modelo Exponencial para 5000 pontos, com ponto inicial $x_0 = (0.0, 0.0, 0.0, 0.0)$ . . . . .	84
7.13	Desempenho dos algoritmos para o modelo Senoidal (Tipo 1) para 100 pontos, com ponto inicial $x_0 = (1.0, 1.0, 1.0, 1.0)$ . . . . .	85
7.14	Desempenho dos algoritmos para o modelo Senoidal (Tipo 1) para 1000 pontos, com ponto inicial $x_0 = (1.0, 1.0, 1.0, 1.0)$ . . . . .	86
7.15	Desempenho dos algoritmos para o modelo Senoidal (Tipo 1) para 5000 pontos, com ponto inicial $x_0 = (1.0, 1.0, 1.0, 1.0)$ . . . . .	87
7.16	Desempenho dos algoritmos para o modelo Senoidal (Tipo 2) para 100 pontos, com ponto inicial $x_0 = (5.0, 5.0, 5.0, 5.0, 5.0)$ . . . . .	88
7.17	Desempenho dos algoritmos para o modelo Senoidal (Tipo 2) para 1000 pontos, com ponto inicial $x_0 = (5.0, 5.0, 5.0, 5.0, 5.0)$ . . . . .	89
7.18	Desempenho dos algoritmos para o modelo Senoidal (Tipo 2) para 5000 pontos, com ponto inicial $x_0 = (5.0, 5.0, 5.0, 5.0, 5.0)$ . . . . .	90
7.19	Desempenho dos algoritmos para o modelo Logístico para 100 pontos, com ponto inicial $x_0 = (0.0, 0.0, 0.0)$ . . . . .	91
7.20	Desempenho dos algoritmos para o modelo Logístico para 1000 pontos, com ponto inicial $x_0 = (0.0, 0.0, 0.0)$ . . . . .	92
7.21	Desempenho dos algoritmos para o modelo Logístico para 5000 pontos, com ponto inicial $x_0 = (0.0, 0.0, 0.0)$ . . . . .	93
7.22	Desempenho dos algoritmos para o modelo Logístico para 5000 pontos, com ponto inicial $x_0 = (1.0, 0.0, 0.0)$ . . . . .	94
8.1	Performance dos métodos empregados nos testes numéricos. . . . .	97

---

---

# SUMÁRIO

---

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Preliminares</b>	<b>3</b>
2.1	Uma breve revisão de Álgebra Linear . . . . .	3
2.1.1	A desigualdade geral de Cauchy-Bunyakovskii-Schwarz . . . . .	4
2.1.2	Normas Matriciais . . . . .	4
2.1.3	Decomposição de Matrizes . . . . .	7
2.1.4	Positividade de uma Matriz Simétrica . . . . .	9
2.2	Uma breve revisão de Otimização . . . . .	10
2.2.1	Minimizadores locais e globais . . . . .	11
2.2.2	Condições de Otimalidade para Problemas de Minimização Irrestrita	11
2.2.3	O Método do Gradiente . . . . .	18
2.2.4	O Método de Newton . . . . .	20
2.2.5	O Método de Newton-Raphson Multidimensional . . . . .	22
<b>3</b>	<b>Uma introdução ao problema de quadrados mínimos</b>	<b>25</b>
3.1	O problema de quadrados mínimos . . . . .	25
3.1.1	O caso linear . . . . .	27
3.1.2	O caso não linear . . . . .	30
3.1.3	Métodos clássicos para o problema de quadrados mínimos não linear .	30
<b>4</b>	<b>O Método de Levenberg-Marquardt</b>	<b>34</b>

---

4.1	O Método de Levenberg-Marquardt para problemas de quadrados mínimos não lineares . . . . .	35
4.2	O parâmetro de <i>damping</i> . . . . .	38
4.2.1	Estratégias para a escolha do parâmetro de <i>damping</i> . . . . .	39
4.3	Algoritmo de Levenberg-Marquardt com Busca Linear . . . . .	43
<b>5</b>	<b>O Problema de Otimização de Menor Valor Ordenado</b>	<b>49</b>
5.1	O problema LOVO . . . . .	49
5.1.1	Uma generalização do problema de quadrados mínimos não linear . . . . .	51
5.2	Algoritmo LOVO com convergência para pontos fracamente críticos . . . . .	51
5.3	O Método de Gauss-Newton para problemas LOVO . . . . .	54
<b>6</b>	<b>O Método de Levenberg-Marquardt para problemas LOVO</b>	<b>56</b>
6.1	O Método de Levenberg-Marquardt para problemas LOVO . . . . .	57
6.1.1	Algoritmo de Levenberg-Marquardt para problemas LOVO e alguns resultados . . . . .	58
<b>7</b>	<b>Experimentos Numéricos</b>	<b>68</b>
7.1	Funções testadas e resultados . . . . .	70
7.1.1	Análise dos Resultados . . . . .	70
7.2	Tabelas . . . . .	75
<b>8</b>	<b>Considerações Finais e Trabalhos Futuros</b>	<b>95</b>
	<b>Bibliografia</b>	<b>98</b>



---

# INTRODUÇÃO

---

Sejam  $F_i, i = 1, \dots, r$ , funções reais definidas em  $\Omega \subset \mathbb{R}^n$  e  $p$  um inteiro tal que  $p \in \{1, \dots, r\}$ . O problema de Otimização de Menor Valor Ordenado (LOVO) consiste em minimizar a soma das funções, tomados os  $p$  menores valores. Tal problema foi proposto inicialmente por Andreani et al [3], os quais também desenvolveram parte substancial da teoria clássica e proveram algumas aplicações práticas, como alinhamento de proteínas, estimação robusta de parâmetros e a busca por padrões ocultos em um determinado conjunto de dados.

Sejam  $y_1, y_2, \dots, y_r \in \mathbb{R}$  dados obtidos através da observação de determinado fenômeno,  $y = \Phi(x, t)$  uma função de ajuste e  $F_i = (\Phi(x, t_i) - y_i)^2$  o erro quadrático obtido na observação  $i$ , ao adotarmos o parâmetro  $x \in \Omega$ . Assim, caso  $p = r$ , o problema LOVO coincide com o problema clássico de quadrados mínimos não-linear. Agora, suponha que  $r - p$  observações provêm de erros de observação e que desejamos eliminar sua influência na determinação da função de ajuste. Dessa forma, este subproblema LOVO pode ser visto como uma generalização do método de quadrados mínimos não-linear, capaz de eliminar a influência de pontos que discrepam dos demais.

Ao generalizar o problema de quadrados mínimos não linear, a Otimização de Menor Valor Ordenado torna-se uma ferramenta valiosa em diversas áreas, podendo ser empregada em inúmeros problemas práticos advindos das Ciências, Economia, Tecnologia, entre outras.

No Capítulo 2 apresentamos alguns resultados básicos de acerca de Álgebra Linear e Otimização, focando principalmente em conceitos relacionados a Matrizes e Minimização Irrestrita, bem como um algoritmo de descida com busca linear, baseado em [10, 16].

No Capítulo 3 introduzimos o problema de quadrados mínimos linear e o caso não linear, bem como alguns conceitos necessários para sua compreensão, incluindo exemplos práticos.

No Capítulo 4 apresentamos o Método de Levenberg-Marquardt, muito utilizado no contexto de problemas de quadrados mínimos não-lineares, e alguns resultados clássicos [13, 15]. Desenvolvemos também a construção de boas estimativas para parâmetros de *damping*, baseados em [5, 7]. Além disso, são propostos dois novos parâmetros e uma estratégia híbrida. Ao final do capítulo há uma contribuição original: desenvolvemos uma variação do Método de Levenberg-Marquardt com Busca Linear, assim como hipóteses e resultados de convergência para tal método.

No Capítulo 5 introduzimos o problema LOVO, o subproblema de quadrados mínimos não-linear generalizado, algumas definições e resultados importantes, assim como um algoritmo para o problema irrestrito, com convergência para pontos fracamente críticos, apresentado em [3]. Também é apresentado um algoritmo desenvolvido em [2] para problemas LOVO, baseado no Método de Gauss-Newton.

A principal contribuição reside no Capítulo 6, onde propomos uma nova abordagem ao problema LOVO irrestrito, através da adaptação do método de Levenberg-Marquardt com Busca Linear. Além do algoritmo, apresentamos hipóteses e resultados de convergência para pontos fracamente críticos.

Por fim, no Capítulo 7 são apresentados dados relativos a performance do novo método, em comparação com o algoritmo baseado em Gauss-Newton, ao serem aplicados em problemas LOVO criados a partir de problemas de quadrados mínimos lineares e não lineares. Considerando o algoritmo proposto, são avaliados os desempenhos de diversos parâmetros de *damping* distintos presentes na literatura, bem como para dois novos parâmetros sugeridos neste trabalho.

---

# PRELIMINARES

---

Neste capítulo apresentamos definições e resultados básicos importantes para a compreensão e o desenvolvimento deste trabalho.

Na Seção 2.1, são abordados conceitos referentes à Álgebra Linear, em especial, a desigualdade geral de Cauchy-Bunyakovskii-Schwarz, normas matriciais, positividade de uma matriz, além de abordarmos brevemente algumas decomposições clássicas de matrizes, tais como LU, LU com Pivotação Parcial, LDV,  $LDL^T$  e Cholesky. Os principais resultados desta seção podem ser encontrados em [17] e [22].

Já a Seção 2.2 contempla diversos resultados de Otimização, principalmente, conceitos ligados à minimização irrestrita. Também são apresentados um algoritmo de descida com busca linear e as definições de série e polinômio de Taylor. Por fim, é retomado o Método do Gradiente e o Método de Newton-Raphson. Para esta seção, nossas principais referências são [10] e [16].

## 2.1 Uma breve revisão de Álgebra Linear

A análise e solução de sistemas de equações algébricas é um problema fundamental que permeia todas as ciências matemáticas, encontrando respaldo em diversas aplicações, desde simples questões de economia doméstica à árduos e volumosos problemas de Mineração de Dados (*Data Mining*).

Apesar do Papiro de Rhind (cerca de 1.650 a.C.) já descrever diversos problemas envolvendo equações lineares segundo Milies [18, p.3] e Santos [20, p.7], Meyer [17, p.1] afirma que o primeiro relato conhecido de análise de equações simultâneas se encontra no livro chinês

*Chiu-chang Suan-chu*, escrito por volta do ano 200 a.C. Neste livro, encontra-se um método de resolução de sistemas lineares o qual consistia da manipulação de varas de bambu coloridas sob regras pré-determinadas. Posteriormente, tal método difundiu-se no Japão e chegou à Europa, onde se tornou conhecido como Eliminação Gaussiana, graças ao matemático alemão Johann Carl Friedrich Gauss (1777-1855) [17, p.1-2].

Vale ainda destacar que, apesar dos “traços” encontrados do desenvolvimento da Álgebra Linear em escritos babilônicos e chineses desde a Antiguidade, esta área da matemática só atingiu a devida consistência teórica no século XIX, com o trabalho dos matemáticos ingleses Arthur Cayley (1821-1895) e James Joseph Sylvester (1814-1897) [20, p.4].

### 2.1.1 A desigualdade geral de Cauchy-Bunyakovskii-Schwarz

O teorema apresentado a seguir é um importante resultado sobre espaços vetoriais com produto interno, pois relaciona o produto interno de dois vetores com o produto da norma induzida desses vetores. Lembre que, se  $\mathbb{V}$  é espaço vetorial sobre  $\mathbb{R}$  ou  $\mathbb{C}$  com produto interno  $\langle x, y \rangle$ ,  $x, y \in \mathbb{V}$ , então  $\|x\| = \sqrt{\langle x, x \rangle}$  define uma norma sobre  $\mathbb{V}$ .

**Teorema 2.1.** (*Desigualdade geral de Cauchy-Bunyakovskii-Schwarz*) *Se  $\mathbb{V}$  é um espaço vetorial com produto interno, então:*

$$|\langle x, y \rangle| \leq \|x\| \|y\|,$$

para todo  $x, y \in \mathbb{V}$ . Note que a igualdade se verifica se e somente se  $y = \alpha x$ , onde  $\alpha = \langle x, y \rangle / \|x\|^2$ .

**Demonstração:** Ver [17, Teorema 5.3.4]. □

### 2.1.2 Normas Matriciais

Em Álgebra Linear, um dos principais tópicos a serem abordados é a chamada Norma Matricial, a qual constitui uma métrica para matrizes.

**Definição 2.2.** [17, Definição 5.2.3] *Seja  $\mathcal{M}(\mathbb{C})$  o conjunto de todas as matrizes complexas*

de ordens finitas. Uma *norma matricial* é uma função  $\|\cdot\| : \mathcal{M}(\mathbb{C}) \rightarrow \mathbb{R}_+$  que satisfaz as seguintes propriedades:

- i)  $\|A\| \geq 0$  e  $\|A\| = 0 \iff A = 0$ ;
- ii)  $\|\alpha A\| = |\alpha| \|A\|$ , para todo escalar  $\alpha$ ;
- iii)  $\|A + B\| \leq \|A\| + \|B\|$ , para matrizes de mesma dimensão;
- iv)  $\|AB\| \leq \|A\| \|B\|$ , para matrizes compatíveis com a multiplicação.

Lembre que, dada uma norma vetorial definida em  $\mathbb{C}^p$ , onde  $p = m \cdot n$ , podemos induzir uma norma matricial no espaço vetorial das matrizes complexas de dimensão  $m \times n$   $\mathbb{C}^{m \times n}$ , definindo

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

para  $A \in \mathbb{C}^{m \times n}$  e  $x \in \mathbb{C}^n$ .

Tal conceito nos será útil para definirmos a *2-Norma*, a norma matricial induzida pela norma vetorial euclidiana.

**Definição 2.3.** [6, Definição 1.39] Seja  $A \in \mathbb{C}^{m \times n}$ . Chamamos de *matriz transposta conjugada* de  $A$ , e denotamos por  $A^*$ , à matriz  $[\overline{A}]^T$ . Isto é,  $A^* \in \mathbb{C}^{n \times m}$  cujos elementos são dados por  $[A^*]_{ij} = \overline{[A]_{ji}}$ .

**Definição 2.4.** [17, Definição 5.2.7] Considere o espaço vetorial  $\mathbb{C}^{m \times n}$ . Chamamos de *2-Norma matricial*, à norma matricial induzida pela norma euclidiana, dada por

$$\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2 = \sqrt{\lambda_{max}},$$

onde  $\lambda_{max}$  é o maior valor de  $\lambda$  tal que  $A^*A - \lambda I$  é singular, isto é, o maior autovalor de  $A^*A$ .

**Teorema 2.5.** *Seja  $A \in \mathbb{C}^{n \times n}$  uma matriz não singular. Então,*

$$\|A^{-1}\|_2 = \frac{1}{\min_{\|x\|_2=1} \|Ax\|} = \sqrt{\lambda_{min}},$$

onde  $\lambda_{min}$  é o menor autovalor da matriz  $A^*A$ .

**Demonstração:** Ver [17, Teorema 5.2.8]. □

Os últimos dois resultados desta seção estabelecem relações importantes entre a  $2$ -Norma de uma matriz e seus autovalores.

**Teorema 2.6.** *Seja  $A \in \mathbb{R}^{n \times n}$  uma matriz não singular com autovalores  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$ . Então:*

$$i) \sigma_1 = \max_{\|x\|_2=1} \|Ax\|_2 = \|A\|_2;$$

$$ii) \sigma_n = \min_{\|x\|_2=1} \|Ax\|_2 = \frac{1}{\|A^{-1}\|_2}.$$

**Demonstração:** Ver [17, Teoremas 5.12.4 e 5.12.5]. □

**Definição 2.7.** [6, Definição 1.41] Seja  $A \in \mathbb{C}^{n \times n}$ . Dizemos que  $A$  é *hermitiana* se  $A = A^*$ , ou equivalentemente, se  $[A]_{ij} = \overline{[A]_{ji}}$ .

**Teorema 2.8.** (*Teorema de Courant-Fischer*) *Seja  $A \in \mathbb{C}^{n \times n}$  uma matriz hermitiana, com autovalores  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$ . Considere  $\mathcal{V}$  um subespaço  $k$ -dimensional sobre  $\mathbb{C}$  arbitrário. Os autovalores de  $A$  são dados por:*

$$\sigma_i = \max_{\dim \mathcal{V}=i} \min_{x \in \mathcal{V}, \|x\|_2=1} x^* Ax, e \tag{2.1}$$

$$\sigma_i = \min_{\dim \mathcal{V}=n-i+1} \max_{x \in \mathcal{V}, \|x\|_2=1} x^* Ax. \tag{2.2}$$

Desse modo, quando  $i = n$  na expressão (2.1) e quando  $i = 1$  em (2.2),  $\mathcal{V} = \mathbb{C}^n$ , estes casos se reduzem as equações:

$$\sigma_1 = \max_{\|x\|_2=1} x^* Ax,$$

$$\sigma_n = \min_{\|x\|_2=1} x^* Ax.$$

**Demonstração:** Ver [17, Teorema 7.5.5]. □

### 2.1.3 Decomposição de Matrizes

Agora, abordaremos algumas decomposições clássicas de matrizes quadradas. Considere  $A \in \mathbb{R}^{n \times n}$ . Caso possamos escrever

$$A = LU,$$

onde  $L \in \mathbb{R}^{n \times n}$  é uma matriz triangular inferior com diagonal unitária e  $U \in \mathbb{R}^{n \times n}$  é triangular superior, dizemos que  $A$  admite decomposição LU.

**Teorema 2.9.** *(Decomposição LU) Seja  $A \in \mathbb{R}^{n \times n}$  uma matriz cujas submatrizes principais são todas não singulares. Existe uma matriz triangular inferior unitária  $L \in \mathbb{R}^{n \times n}$  e uma matriz triangular superior  $U \in \mathbb{R}^{n \times n}$  tais que  $A = LU$ , e esta decomposição é única.*

**Demonstração:** Ver [22, Teorema 1.7.19]. □

Conhecida a decomposição LU da matriz  $A$ , a solução do sistema linear  $Ax = b$ , torna-se relativamente simples, bastando resolver os sistemas triangulares:

$$Ly = b,$$

$$Ux = y.$$

Segundo Meyer [17, p.146], a vantagem desta abordagem consiste no fato de que uma vez computada a decomposição LU de  $A$ , qualquer outro sistema linear do tipo  $Ax = \bar{b}$ , pode ser resolvido utilizando apenas  $n^2$  multiplicações/divisões e  $n^2 - n$  adições/subtrações.

Contudo, pode ocorrer que  $A \in \mathbb{R}^{n \times n}$  não admita decomposição LU, como é o caso de matrizes não singulares nas quais ocorrem pivôs nulos ao aplicar o processo de Eliminação Gaussiana. Neste caso, se  $A$  é não singular, existe uma matriz de permutação  $P$  tal que  $PA = LU$ , tal fatoração é chamada de decomposição LU com Pivotação Parcial.

**Teorema 2.10.** *(Decomposição LU com Pivotação Parcial) Seja  $A \in \mathbb{R}^{n \times n}$  uma matriz não singular. Existem matrizes de permutação  $P_1, P_2 \in \mathbb{R}^{n \times n}$  tais que  $P_1AP_2$  admite decomposição LU, isto é,  $P_1AP_2 = LU$ . Note que apenas  $P_1$  ou  $P_2$  é necessário.*

**Demonstração:** Ver [8, Teorema 2.5] □

Dada a decomposição  $LU = PA$ , resolver o sistema linear  $Ax = b$  é relativamente simples, basta fazer  $\bar{b} = Pb$  e resolver os sistemas triangulares:

$$\begin{aligned} Ly &= \bar{b}, \\ Ux &= y. \end{aligned}$$

Consideremos agora o caso em que  $A \in \mathbb{R}^{n \times n}$  é matriz singular e possui decomposição LU. Note que, por definição, a matriz  $L$  possui diagonal principal unitária. Entretanto, a matriz  $U$  não possui necessariamente diagonal principal unitária, o que dá uma certa assimetria a esta decomposição. Dada a matriz triangular superior  $U$ , podemos decompô-la em  $U = DV$ , em que  $D \in \mathbb{R}^{n \times n}$  é matriz diagonal e  $V \in \mathbb{R}^{n \times n}$  é matriz triangular superior com diagonal unitária:

$$U = \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & u_{nn} \end{bmatrix} = \begin{bmatrix} u_{11} & 0 & \dots & 0 \\ 0 & u_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & u_{nn} \end{bmatrix} \begin{bmatrix} 1 & \frac{u_{12}}{u_{11}} & \dots & \frac{u_{1n}}{u_{11}} \\ 0 & 1 & \dots & \frac{u_{2n}}{u_{22}} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} = DV.$$

Tal decomposição é chamada de decomposição LDV.

**Teorema 2.11.** (*Decomposição LDV*) *Seja  $A \in \mathbb{R}^{n \times n}$  uma matriz cujas submatrizes principais são todas não singulares. Existem uma matriz triangular inferior unitária  $L \in \mathbb{R}^{n \times n}$ , uma matriz diagonal  $D \in \mathbb{R}^{n \times n}$  e uma matriz triangular superior unitária  $V \in \mathbb{R}^{n \times n}$  tais que  $A = LDV$ , e esta decomposição é única.*

**Demonstração:** Ver [22, Teorema 1.7.28]. □

Caso  $A$  seja simétrica e admita decomposição LDV, obtemos a chamada decomposição  $LDL^T$ :

**Teorema 2.12.** (*Decomposição  $LDL^T$* ) *Seja  $A \in \mathbb{R}^{n \times n}$  uma matriz simétrica cujas submatrizes principais são todas não singulares. Então  $A$  pode ser escrita como o produto  $A = LDL^T$ , onde  $L \in \mathbb{R}^{n \times n}$  é uma matriz triangular inferior unitária,  $D \in \mathbb{R}^{n \times n}$  é uma matriz diagonal e tal decomposição é única.*

**Demonstração:** Ver [22, Teorema 1.7.30]. □

Antes de apresentarmos a última decomposição de matrizes, necessitamos introduzir o conceito de matriz definida positiva. Tal assunto será aprofundado na subseção 2.1.4.

**Definição 2.13.** [17, p.559] Uma matriz simétrica  $A \in \mathbb{R}^{n \times n}$  é dita *definida positiva* se  $x^T A x > 0$  para todo vetor  $x \in \mathbb{R}^n$  não nulo.

**Teorema 2.14.** (*Decomposição de Cholesky*) Seja  $A \in \mathbb{R}^{n \times n}$  uma matriz definida positiva. Então  $A$  pode ser decomposta de maneira única como o produto  $A = R^T R$ , onde  $R \in \mathbb{R}^{n \times n}$  é uma matriz triangular superior, cujas entradas da diagonal principal são todas positivas. A matriz  $R$  é chamada de *Fator de Cholesky* de  $A$ .

**Demonstração:** Ver [22, Teorema 1.4.7]. □

## 2.1.4 Positividade de uma Matriz Simétrica

A definição a seguir caracteriza a chamada positividade de uma matriz simétrica.

**Definição 2.15.** [12, p.495-496] Seja  $A \in \mathbb{R}^{n \times n}$  uma matriz simétrica de ordem  $n$ . Dizemos que  $A$  é:

- i) *semidefinida positiva*, e denota-se por  $A \geq 0$ , se  $x^T A x \geq 0, \forall x \in \mathbb{R}^n$ .
- ii) *definida positiva*, e denota-se por  $A > 0$ , se  $x^T A x > 0, \forall x \in \mathbb{R}^n, x \neq 0$ .
- iii) *semidefinida negativa*, e denota-se por  $A \leq 0$ , se  $x^T A x \leq 0, \forall x \in \mathbb{R}^n$ .
- iv) *definida negativa*, e denota-se por  $A < 0$ , se  $x^T A x < 0, \forall x \in \mathbb{R}^n, x \neq 0$ .
- v) *indefinida*, quando existem  $x, y \in \mathbb{R}^n$  tais que  $x^T A x < 0 < y^T A y$ .

No caso específico das matrizes definidas positivas, podemos ainda caracterizá-las da seguinte maneira:

**Observação 2.16.** [17, p.559] Seja  $A \in \mathbb{R}^{n \times n}$  uma matriz simétrica de ordem  $n$ . As afirmações a seguir são equivalentes:

- i)  $A$  é definida positiva;
- ii)  $x^T A x > 0, \forall x \in \mathbb{R}^n, x \neq 0$ ;
- iii) Todos os autovalores de  $A$  são positivos;
- iv)  $A$  é semidefinida positiva e não singular.
- v)  $A = B^T B$ , para alguma matriz não singular  $B$ .

Apesar da fatoração  $A = B^T B$  não ser única, existe apenas uma única matriz triangular superior com entradas diagonais positivas  $R$  tal que  $A = R^T R$ , isto é, se  $A$  é matriz definida positiva, então  $A$  admite Decomposição de Cholesky e o Fator de Cholesky associado a  $A$  é a matriz  $R$ , conforme [17, p.559].

## 2.2 Uma breve revisão de Otimização

Apesar de muitas vezes não percebermos, interagimos com problemas de otimização constantemente. A busca por soluções ótimas para problemas simples do dia-a-dia, como a quantidade ideal de comida a ser preparada para o jantar de modo que não haja desperdício de alimentos, por exemplo, podem constituir problemas interessantes de otimização.

Reavaliar processos de extração, produção, transporte, acondicionamento e entrega, buscando reduzir quer seja o valor de custo ou o impacto ambiental da atividade, são apenas alguns dos problemas mais famosos com os quais a Otimização se depara.

De modo geral, a área de Otimização se ocupa do estudo de problemas matemáticos, reais ou teóricos, nos quais se busca maximizar ou minimizar uma função de várias variáveis, através da escolha metódica de valores dentro de um conjunto factível.

### 2.2.1 Minimizadores locais e globais

Considere o seguinte problema genérico de Otimização:

$$\min_{x \in \Omega} f(x), \quad (2.3)$$

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

**Definição 2.17.** [10, Definição 1.1] Um ponto  $x_* \in \Omega$  é dito *minimizador local* de (2.3) se existe  $\varepsilon > 0$  tal que  $f(x) \geq f(x_*)$  para todo  $x \in \Omega$  tal que  $\|x - x_*\| < \varepsilon$ . Caso  $f(x) > f(x_*)$  para todo  $x \in \Omega$  tal que  $\|x - x_*\| < \varepsilon$ , dizemos que  $x_*$  é *minimizador local estrito* de (2.3).

**Definição 2.18.** [10, Definição 1.2] Um ponto  $x_* \in \Omega$  é dito *minimizador global* de (2.3) se  $f(x) \geq f(x_*)$  para todo  $x \in \Omega$ . Caso  $f(x) > f(x_*)$  para todo  $x \in \Omega$ , dizemos que  $x_*$  é *minimizador global estrito* de (2.3).

O teorema a seguir garante de existência de minimizador global, sob hipóteses de compacidade de  $\Omega$  e continuidade de  $f$  em  $\Omega$ .

**Teorema 2.19.** (*Teorema de Bolzano-Weierstrass*) Se  $\Omega$  é compacto e  $f : \Omega \rightarrow \mathbb{R}$  é contínua, então existe  $x_*$  minimizador global de (2.3).

**Demonstração:** Ver [16, Teorema 1.3.1]. □

### 2.2.2 Condições de Otimalidade para Problemas de Minimização Irrestrita

Considere o problema de minimização irrestrita:

$$\min_{x \in \mathbb{R}^n} f(x), \quad (2.4)$$

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

Os próximos dois teoremas estabelecem as condições necessárias para minimizadores locais, isto é, condições que todos os minimizadores locais devem satisfazer. Contudo, vale lembrar dois importantes resultados para funções de uma variável:

**Lema 2.20.** *Se  $f : \mathbb{R} \rightarrow \mathbb{R}$  é função de classe  $C^1$  e  $x_*$  é minimizador local de  $f$ , então  $f'(x_*) = 0$ .*

**Demonstração:** Ver [11, Seção 9.7, Teorema 1]. □

**Lema 2.21.** *Se  $f : \mathbb{R} \rightarrow \mathbb{R}$  é função de classe  $C^2$  e  $x_*$  é minimizador local de  $f$ , então  $f'(x_*) = 0$  e  $f'(x_*) \geq 0$ .*

**Demonstração:** Ver [11, Seção 9.7, Teorema 2]. □

**Teorema 2.22.** *(Condição Necessária de 1ª Ordem) Seja  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  uma função de classe  $C^1$ . Se  $x_*$  é minimizador local de  $f$ , então  $\nabla f(x_*) = 0$ .*

**Demonstração:** Seja  $d \in \mathbb{R}^n$  uma direção arbitrária e defina  $\phi : \mathbb{R} \rightarrow \mathbb{R}$  como  $\phi(t) = f(x_* + td)$ . Dado que  $x_*$  é minimizador local de  $f$ , temos que  $t = 0$  é minimizador local de  $\phi$  e, portanto, pelo Lema 2.20,  $\phi'(0) = 0$ . Derivando  $\phi$ , temos que  $\phi'(t) = \nabla f(x_* + td)^T d$ . Calculando a expressão anterior em  $t = 0$ , obtemos  $\phi'(0) = \nabla f(x_*)^T d = 0$ . Da arbitrariedade na escolha de  $d \in \mathbb{R}^n$ , podemos concluir que  $\nabla f(x_*) = 0$  □

**Teorema 2.23.** *(Condição Necessária de 2ª Ordem) Seja  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  uma função de classe  $C^2$ . Se  $x_*$  é minimizador local de  $f$ , então  $\nabla f(x_*) = 0$  e  $\nabla^2 f(x_*) \geq 0$ .*

**Demonstração:** A primeira parte da demonstração segue de imediato do Teorema 2.22. Como na demonstração anterior, seja  $d \in \mathbb{R}^n$  uma direção arbitrária e defina  $\phi : \mathbb{R} \rightarrow \mathbb{R}$  como  $\phi(t) = f(x_* + td)$ . Derivando  $\phi$  duas vezes, obtemos  $\phi''(t) = d^T \nabla^2 f(x_* + td) d$ . Assim, pelo Lema 2.21,  $\phi''(0) = d^T \nabla^2 f(x_*) d \geq 0$ . Portanto, pela arbitrariedade na escolha de  $d \in \mathbb{R}^n$ , temos que  $\nabla^2 f(x_*) \geq 0$  □

Antes de enunciarmos o próximo teorema de minimização irrestrita, iremos resgatar o conceito de Fórmula de Taylor.

**Teorema 2.24.** (*Fórmula de Taylor infinitesimal*) Dado um intervalo  $I \subset \mathbb{R}$ , seja  $f : I \rightarrow \mathbb{R}$  uma função  $n$  vezes derivável no ponto  $a \in I$ . A função  $r : J \rightarrow \mathbb{R}$ , definida no intervalo  $J = \{h \in \mathbb{R}; a + h \in I\}$  pela igualdade

$$f(a + h) = f(a) + f'(a)h + \frac{f''(a)}{2}h^2 + \dots + \frac{f^{(n)}(a)}{n!}h^n + r(h),$$

satisfaz  $\lim_{h \rightarrow 0} \frac{r(h)}{h^n} = 0$ . Reciprocamente, se  $p(h)$  é um polinômio de grau  $\leq n$  tal que  $r(h) = f(a + h) - p(h)$  cumpra  $\lim_{h \rightarrow 0} \frac{r(h)}{h^n} = 0$ , então  $p(h)$  é o Polinômio de Taylor de ordem  $n$  de  $f$  no ponto  $a$ , isto é,

$$p(h) = \sum_{i=0}^n \frac{f^{(i)}(a)}{i!} h^i.$$

**Demonstração:** Ver [14, Seção 9.1, Teorema 1]. □

No caso específico em que  $a = 0$ , o Polinômio de Taylor de ordem  $n$  de  $f$  no ponto  $a$  é chamado de Polinômio de Maclaurin de ordem  $n$  de  $f$ .

**Exemplo 2.25.** Na Figura 2.1 estão representados os polinômios de Maclaurin de ordem 1, 3, 5 e 7 da função  $f(x) = \text{sen}(x)$ .

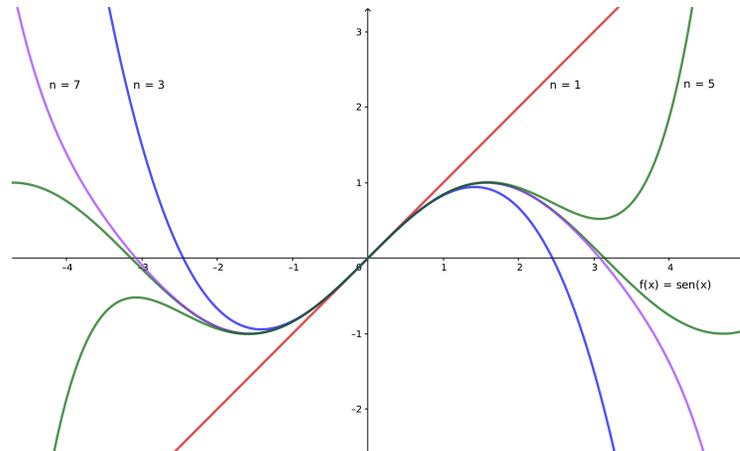


Figura 2.1: Polinômios de Maclaurin de graus 1, 3, 5 e 7 da função seno.

**Teorema 2.26.** (*Condição Suficiente de Otimalidade*) Seja  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  é função de classe  $C^2$ . Se  $x_* \in \mathbb{R}^n$  é tal que  $\nabla f(x_*) = 0$  e  $\nabla^2 f(x_*) > 0$ , então  $x_*$  é minimizador local estrito de  $f$ .

**Demonstração:** Considere o conjunto  $\Theta = \{s \in \mathbb{R}^n; \|s\|_2 = 1\}$  e defina a função  $\gamma : \Theta \rightarrow \mathbb{R}$  dada por  $\gamma(s) = s^T \nabla^2 f(x_*) s$ . É fácil ver que  $\gamma$  é uma função contínua e, dado que  $\Theta$  é um conjunto compacto, o Teorema 2.19 garante que  $\gamma$  admite valor máximo e mínimo em  $\Theta$ .

Seja  $a \in \mathbb{R}$  o valor mínimo de  $\gamma$  em  $\Theta$ . Por hipótese,  $\nabla^2 f(x_*) > 0$ , logo  $\gamma(s) = s^T \nabla^2 f(x_*) s \geq a > 0$ . Seja  $d \in \mathbb{R}^n$  uma direção arbitrária. Note que  $\frac{d}{\|d\|_2} \in \Theta$ , logo

$$d^T \nabla^2 f(x_*) d \geq a \|d\|_2^2. \quad (2.5)$$

Considere a expansão em Série de Taylor da função  $f$  em torno de  $x_*$ , dada por

$$f(x_* + d) = f(x_*) + \nabla f(x_*)^T d + \frac{1}{2} d^T \nabla^2 f(x_*)^T d + r(d), \quad (2.6)$$

onde  $\lim_{\|d\|_2 \rightarrow 0} \frac{r(d)}{\|d\|_2^2} = 0$ . Por hipótese,  $\nabla f(x_*) = 0$ . Logo, por (2.6),

$$\begin{aligned} \frac{f(x_* + d) - f(x_*)}{\|d\|_2^2} &= \frac{1}{2} \frac{d^T}{\|d\|_2} \nabla^2 f(x_*) \frac{d}{\|d\|_2} + \frac{r(d)}{\|d\|_2^2}, \\ &\geq \frac{a}{2} + \frac{r(d)}{\|d\|_2^2}. \end{aligned} \quad (2.7)$$

Uma vez que  $\lim_{\|d\|_2 \rightarrow 0} \frac{r(d)}{\|d\|_2^2} = 0$  e  $-\frac{a}{4} < 0$ , todo  $d \in \mathbb{R}^n$  com  $\|d\|_2$  suficientemente pequeno satisfaz  $\frac{r(d)}{\|d\|_2^2} > -\frac{a}{4}$ .

Deste modo, para  $d \in \mathbb{R}^n$  com  $\|d\|_2$  suficientemente pequeno, (2.7) garante que

$$\frac{f(x_* + d) - f(x_*)}{\|d\|_2^2} > \frac{a}{2} - \frac{a}{4} > \frac{a}{4}.$$

Logo,

$$f(x_* + d) - f(x_*) > \frac{a}{4} \|d\|_2^2 > 0,$$

e assim  $f(x_* + d) > f(x_*)$ . Portanto,  $x_*$  é minimizador local estrito de  $f$ .  $\square$

Considerando ainda o problema de minimização irrestrita (2.4), seja  $y \in \mathbb{R}^n$  tal que  $\nabla f(y) \neq 0$ . O Teorema 2.22 afirma que  $y$  não é minimizador local de  $f$  em  $\mathbb{R}^n$ , logo, pela contrapositiva, em toda vizinhança de  $y$  existe um ponto, digamos  $z \in \mathbb{R}^n$ , tal que  $f(z) < f(y)$ . Nos interessa, portanto, caracterizar as possíveis direções a partir de  $y$  nas quais é possível encontrar  $z$ , isto é, direções nas quais é possível diminuir o valor da função.

**Definição 2.27.** [16, p.100] Sejam  $x, d \in \mathbb{R}^n$ . Dizemos que  $d$  é *direção de descida* a partir de  $x$  se existe  $\varepsilon > 0$  tal que, para todo  $t \in (0, \varepsilon]$ , temos que  $f(x + td) < f(x)$ .

**Lema 2.28.** Se  $\nabla f(x)^T d < 0$ , então  $d$  é *direção de descida* a partir de  $x$ .

**Demonstração:** Ver [16, Lema 6.1.1]. □

O teorema que será apresentado a seguir estabelece a chamada Condição de Decréscimo Suficiente, ou Condição de Armijo. Essencialmente, tal condição exige que o decréscimo no valor da função seja de certa forma proporcional ao tamanho do passo dado [10, p.25].

**Teorema 2.29.** (*Condição de Armijo*) Sejam  $\alpha \in (0, 1)$  uma constante e  $x, d \in \mathbb{R}^n$ , satisfazendo  $\nabla f(x) \neq 0$  e  $\nabla f(x)^T d < 0$ . Então existe  $\epsilon = \epsilon(\alpha)$ , tal que  $f(x+td) \leq f(x) + \alpha t \nabla f(x)^T d$  para todo  $t \in (0, \epsilon]$ .

**Demonstração:** Por hipótese, temos que  $\nabla f(x)^T d < 0$ , logo  $\nabla f(x)^T d \neq 0$ . Portanto

$$\nabla f(x)^T d = \lim_{t \rightarrow 0} \frac{f(x + td) - f(x)}{t}.$$

Logo,

$$\lim_{t \rightarrow 0} \frac{f(x + td) - f(x)}{t \nabla f(x)^T d} = 1.$$

Deste modo, existe  $\epsilon > 0$  tal que para todo  $t \in (0, \epsilon]$ ,

$$\frac{f(x + td) - f(x)}{t \nabla f(x)^T d} \geq \alpha,$$

isto é,  $f(x + td) \leq f(x) + \alpha t \nabla f(x)^T d$ , para todo  $t \in (0, \epsilon]$ . □

A seguir, apresentamos um algoritmo básico de descida com Backtracking. Este algoritmo foi baseado no Algoritmo 6.1.5, proposto por [16]. Um algoritmo semelhante pode ser encontrado em [10, Algoritmo 4.2].

---

**Algoritmo 1:** ALGORITMO DE DESCIDA COM BACKTRACKING
 

---

**Entrada:**  $f(x)$ ,  $\nabla f(x)$ , uma aproximação inicial  $x_0 \in \mathbb{R}^n$ , constantes  $\alpha \in (0, 1)$ ,  $\beta > 0$  e  $\theta \in (0, 1)$ .

**Saída:**  $x_* \in \mathbb{R}^n$ , minimizador de  $f(x)$ .

```

1 início
2    $k \leftarrow 0$  ;
3   se  $\nabla f(x_k) = 0$  então
4     Pare.
5   senão
6     Escolha  $d_k \in \mathbb{R}^n$  tal que:
7       
$$\|d_k\|_2 \geq \beta \|\nabla f(x_k)\|_2 \tag{2.8}$$

8       
$$\nabla f(x_k)^T d_k \leq -\theta \|\nabla f(x_k)\|_2 \|d_k\|_2 \tag{2.9}$$

9       Faça  $t = 1$  ;
10      enquanto  $f(x_k + td_k) > f(x_k) + \alpha t \nabla f(x_k)^T d_k$  faça
11        Escolha novo  $t \in [0.1t, 0.9t]$ .
12      fim
13      Faça:
14          
$$x_{k+1} \leftarrow x_k + td_k$$

15          
$$k \leftarrow k + 1$$

16    fim
17 fim
18 retorna  $x_k$ 

```

---

As expressões (2.8) e (2.9) são conhecidas como Condição Beta e Condição do Ângulo, respectivamente. Tais condições devem ser satisfeitas pela direção de descida escolhida, de modo a inibir o mau comportamento do algoritmo. Mais especificamente:

- A Condição Beta impede que as direções de descida tomadas sejam desnecessariamente pequenas ([16], p.102), exigindo que  $\|d_k\|_2$  seja maior que determinada fração do gradiente da função aplicada no iterando  $x_k$ .

- A Condição do Ângulo impede que a direção  $d_k$  seja “quase” ortogonal à  $\nabla f(x_k)$  ([10], p.26). Deste modo, as direções admissíveis formam um espécie de cone agudo com eixo na semi-reta gerada por  $-\nabla f(x_k)$  [16, p.103].

O teorema a seguir, bem como sua demonstração, já são conhecidas, conforme [16, Teorema 6.1.6]. A fim de notação, utilizamos o símbolo  $\subset_\infty$  para indicar *subconjunto infinito*.

**Teorema 2.30.** *Se  $x_*$  é ponto limite de uma sequência gerada pelo Algoritmo 1, então  $\nabla f(x_*) = 0$ .*

**Demonstração:** Para cada  $k \in \mathbb{N}$ , denotamos  $s_k = x_{k+1} - x_k = td_k$  para todo  $k \in \mathbb{N}$ . Seja  $K_1 \subset_\infty \mathbb{N}$  tal que  $\lim_{k \in K_1} x_k = x_*$ . Deste modo, temos dois casos:

- $\lim_{k \in K_1} \|s_k\|_2 = 0$ ;
- Existem um subconjunto infinito  $K_2 \subset_\infty K_1$  e  $\epsilon > 0$  tais que  $\|s_k\|_2 > \epsilon$  para todo  $k \in K_2$ .

Suponhamos que valha i). Então, dividimos a prova em dois casos:

- Existe  $K_3 \subset_\infty K_1$  tal que  $s_k = d_k$ , isto é,  $t = 1$ .

Pela validade da Condição Beta,

$$\|\nabla f(x_*)\|_2 = \lim_{k \in K_3} \|\nabla f(x_k)\|_2 \leq \lim_{k \in K_3} \frac{\|d_k\|_2}{\beta} = \lim_{k \in K_3} \frac{\|s_k\|_2}{\beta} = 0.$$

Portanto,  $\|\nabla f(x_*)\|_2 = 0$  e assim  $\nabla f(x_*) = 0$ .

- Para todo  $k \in K_1$ ,  $k \geq k_0$ , temos  $t < 1$ .

Neste caso, para todo  $k \in K_1$ ,  $k > k_0$  existe  $\bar{s}_k$  um múltiplo de  $s_k$  tal que  $\|\bar{s}_k\|_2 \leq 10\|s_k\|_2$  e  $f(x_k + \bar{s}_k) > f(x_k) + \alpha \nabla f(x_k)^T \bar{s}_k$ . Desta maneira, é fácil ver que  $\lim_{k \in K_1} \|\bar{s}_k\|_2 = 0$  e que a sequência  $\{\bar{s}_k\}_{K_1}$  satisfaz a Condição do Ângulo,

$$\nabla f(x_k)^T \bar{s}_k \leq -\theta \|\nabla f(x_k)\|_2 \|\bar{s}_k\|_2, \quad (2.10)$$

para todo  $k \in K_1$ ,  $k \geq k_0$ .

Seja  $v$  um ponto de acumulação de  $\frac{\bar{s}_k}{\|\bar{s}_k\|_2}$ . Então existe  $K_4 \subset_\infty K_1$  tal que  $\lim_{k \in K_4} \frac{\bar{s}_k}{\|\bar{s}_k\|_2} = v$ . Portanto, por (2.10),

$$\nabla f(x_*)^T v = \lim_{k \in K_4} \nabla f(x_k)^T v = \lim_{k \in K_4} \nabla f(x_k)^T \frac{\bar{s}_k}{\|\bar{s}_k\|_2} \leq -\theta \lim_{k \in K_1} \|\nabla f(x_k)\|_2. \quad (2.11)$$

Assim, para todo  $k \in K_4$ ,  $f(x_k + \bar{s}_k) - f(x_k) = \nabla f(x_k + \xi_k \bar{s}_k)^T \bar{s}_k$ ,  $\xi_k \in (0, 1)$ . Portanto, como a Condição de Armijo falha em  $\bar{s}_k$ ,  $\nabla f(x_k + \xi_k \bar{s}_k) > \alpha \nabla f(x_k)^T \bar{s}_k$ , para todo  $k \in K_4$ . Isto é, para todo  $k \in K_4$ ,  $\nabla f(x_*)^T v \geq \alpha \nabla f(x_k)^T v$ . Logo,  $\nabla f(x_*)^T v \geq 0$  e por (2.11) segue  $\nabla f(x_*)^T v = 0$ . Se  $\nabla f(x_*) \neq 0$ , por (2.11), para  $j \in K_4$  e  $k$  suficientemente grande,

$$0 = \nabla f(x_*)^T v \leq -\theta \|\nabla f(x_k)\|_2 < 0.$$

Portanto,  $\nabla f(x_*) = 0$ .

Suponhamos agora a validade de ii). Pela validade da Condição de Armijo,

$$\begin{aligned} f(x_k + s_k) &\leq f(x_k) + \alpha \nabla f(x_k)^T s_k, \\ &\leq f(x_k) - \alpha \theta \|\nabla f(x_k)\|_2 \|s_k\|_2, \\ &\leq f(x_k) - \alpha \theta \epsilon \|\nabla f(x_k)\|_2, \end{aligned}$$

para todo  $k \in K_2$ . Portanto,  $f(x_{k+1}) - f(x_k) \leq -\alpha \theta \epsilon \|\nabla f(x_k)\|_2$ , ou seja,

$$\frac{f(x_k) - f(x_{k+1})}{\alpha \theta \epsilon} \geq \|\nabla f(x_k)\|_2.$$

Passando o limite para  $k \in K_2$  e pela continuidade de  $f$ ,  $\lim_{k \in K_2} \|\nabla f(x_k)\|_2 = 0$ .

Portanto  $\nabla f(x_*) = 0$ . □

### 2.2.3 O Método do Gradiente

Seja  $f : \mathbb{R}^n \rightarrow R$  uma função de classe  $C^1$ . Considere o problema irrestrito:

$$\min_{x \in \mathbb{R}^n} f(x). \quad (2.12)$$

Um dos métodos mais difundidos para resolução deste tipo de problema é o método clássico do Gradiente, também conhecido como método de Cauchy. No contexto do Algoritmo 1, este método consiste na escolha de  $d_k$  como sendo a direção  $-\nabla f(x_k)$  a partir de  $x_k$ , para cada iteração  $k$ .

A escolha desta direção justifica-se pelo fato de que a direção oposta ao gradiente de  $f$  é a direção de maior decréscimo da função. Note ainda que  $d_k = -\nabla f(x_k)$  satisfaz naturalmente as condições (2.8) e (2.9). Portanto, a convergência do método é garantida pelo Teorema (2.30).

A seguir, apresentamos um algoritmo para o método de Cauchy baseado em [10, Algoritmo 6.1], com a Condição de Armijo.

---

**Algoritmo 2:** ALGORITMO DE CAUCHY

---

**Entrada:**  $f(x)$ ,  $\nabla f(x)$ ,  $\alpha \in (0, 1)$  e uma aproximação inicial  $x_0 \in \mathbb{R}^n$ .

**Saída:**  $x_* \in \mathbb{R}^n$ , minimizador de  $f(x)$ .

```

1 início
2    $k \leftarrow 0$  ;
3   se  $\nabla f(x_k) = 0$  então
4     Pare.
5   senão
6     Faça  $d_k = -\nabla f(x_k)$ ;
7     Faça  $t = 1$ ;
8     enquanto  $f(x_k + td_k) > f(x_k) + \alpha t \nabla f(x_k)^T d_k$  faça
9       Escolha novo  $t \in [0.1t, 0.9t]$ .
10    fim
11    Faça  $x_{k+1} \leftarrow x_k + td_k$ ;
12    Faça  $k \leftarrow k + 1$ ;
13  fim
14 fim
15 retorna  $x_k$ 

```

---

## 2.2.4 O Método de Newton

Seja  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  uma função de classe  $C^2$ . Considere o problema de minimização irrestrito

$$\min_{x \in \mathbb{R}^n} f(x). \quad (2.13)$$

Tomando o Polinômio de Taylor de segunda ordem de  $f$  ao redor de  $x_k$ , temos uma aproximação para o valor de  $f(x)$ , numa vizinhança deste ponto,

$$f(x) \approx f(x_k) + \nabla f(x_k)^T(x - x_k) + \frac{1}{2}(x - x_k)^T \nabla^2 f(x_k)(x - x_k), \quad (2.14)$$

onde  $\nabla f(x)$  e  $\nabla^2 f(x)$  são o gradiente e a hessiana de  $f$ , respectivamente.

Assim, nosso objetivo será minimizar (2.14), modelo local de  $f(x)$ . Tomando  $d = x - x_k$ , podemos resolver o problema (2.13) através da solução do problema aproximado

$$\min_{d \in \mathbb{R}^n} f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2 f(x_k) d. \quad (2.15)$$

Pelo Teorema 2.22, temos que a direção de minimização satisfaz

$$\frac{\partial}{\partial d} \left( f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2 f(x_k) d \right) = 0.$$

Logo,  $\nabla f(x_k) + \nabla^2 f(x_k) d = 0$ . Portanto,

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

Deste modo, desde que  $\nabla^2 f(x_k)$  seja não singular, a direção de minimização a partir de  $x_k$  é dada por

$$d_k = -(\nabla^2 f(x_k))^{-1} \nabla f(x_k). \quad (2.16)$$

Assim, podemos caracterizar o Método de Newton pela iteração

$$x_{k+1} = x_k - (\nabla^2 f(x_k))^{-1} \nabla f(x_k). \quad (2.17)$$

O teorema a seguir estabelece a convergência do método.

**Teorema 2.31.** *Seja  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  duas vezes diferenciável e  $x_* \in \mathbb{R}^n$  tais que:*

- i)  $\nabla^2 f(x)$  é contínua numa vizinhança aberta  $\Omega \subset \mathbb{R}^n$  de  $x_*$ ;*

ii)  $\nabla f(x_*) = 0$ ;

iii)  $\nabla^2 f(x_*)$  é definida positiva;

iv) existe  $L > 0$  tal que  $\|\nabla^2 f(x) - \nabla^2 f(x_*)\| \leq L\|x - x_*\|$ , para todo  $x \in \Omega$ .

Considerando a iteração dada em (2.17), temos que:

i) se o ponto inicial  $x_0$  está suficientemente próximo de  $x_*$ , a sequência de iterações converge para  $x_*$ ;

ii) a ordem de convergência é quadrática;

iii) a sequência da norma dos gradientes  $\{\|\nabla f(x_k)\|\}$  converge quadraticamente para zero.

**Demonstração:** Ver [19], Teorema 3.5. □

Na sequência, apresentamos um algoritmo para o Método de Newton baseado em [10, Algoritmo 6.2], com Armijo.

---

**Algoritmo 3:** ALGORITMO DE NEWTON

---

**Entrada:**  $f(x)$ ,  $\nabla f(x)$ ,  $\nabla^2 f(x)$ ,  $\alpha \in (0, 1)$  e uma aproximação inicial  $x_0 \in \mathbb{R}^n$ .**Saída:**  $x_* \in \mathbb{R}^n$ , minimizador de  $f(x)$ .

```

1 início
2    $k \leftarrow 0$  ;
3   se  $\nabla f(x_k) = 0$  então
4     Pare.
5   senão
6     Faça  $d_k = -(\nabla^2 f(x_k))^{-1} \nabla f(x_k)$ ;
7     enquanto  $f(x_k + td_k) > f(x_k) + \alpha t \nabla f(x_k)^T d_k$  faça
8       Escolha novo  $t \in [0.1t, 0.9t]$ .
9     fim
10    Faça  $x_{k+1} \leftarrow x_k + td_k$ ;
11    Faça  $k \leftarrow k + 1$ ;
12  fim
13 fim
14 retorna  $x_k$ 

```

---

### 2.2.5 O Método de Newton-Raphson Multidimensional

Seja  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  uma função de classe  $C^2$  e considere o problema de encontrar um minimizador de  $f$ . Pelo Teorema (2.22), qualquer minimizador  $x_* \in \mathbb{R}^n$  de  $f$  satisfaz a condição  $\nabla f(x) = 0$ .

Definindo  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$  como  $F(x) = \nabla f(x)$ , nosso problema inicial é equivalente a resolver o sistema de equações  $F(x) = 0$ , o qual pode ser ou não linear e cuja solução nem sempre poderá ser obtida de maneira direta.

Desta maneira, dado  $x_k \in \mathbb{R}^n$ , podemos considerar a aproximação dada pelo polinômio de Taylor de grau 1 da função  $F$  em torno do ponto  $x_k$  e resolver o sistema linear

$$F(x_k) + J(x_k)(x - x_k) = 0,$$

onde  $J(x_k)$  denota a matriz jacobiana de  $F$  avaliada em  $x_k$ . Note que  $J(x_k) \in \mathbb{R}^{n \times n}$ . Caso

$J(x_k)$  seja não singular, o sistema pode ser resolvido fazendo

$$x_{k+1} = x_k - (J(x_k))^{-1}F(x_k), \quad (2.18)$$

obtendo assim uma nova aproximação para a raiz da função  $F$ . A expressão (2.18) corresponde a uma iteração do método de Newton-Raphson Multidimensional.

Retornando ao nosso problema inicial, podemos reescrever (2.18) como

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

e portanto, obtivemos a mesma iteração dada pelo Método de Newton (2.17). Dada esta equivalência entre os métodos, a convergência de Newton-Raphson segue de imediato do Teorema 2.31.

O Método de Newton-Raphson também é chamado de Método de Newton para zeros de funções. Segundo Martínez e Santos [16, p.74] e Santos Fonseca [21, p.1], o método foi proposto inicialmente em 1669 pelo matemático britânico Isaac Newton (1643-1727) para encontrar raízes de funções polinomiais. Posteriormente, em 1690, foi estendido para funções reais quaisquer por Joseph Raphson (c. 1648-1715). Sua generalização para sistemas de equações não lineares surgiu apenas em 1740, com o trabalho do matemático Thomas Simpson (1710-1761).

A seguir, apresentaremos um algoritmo básico para este método. Este algoritmo é uma adaptação de [5, Algoritmo 3], [16, p.74-75] e [21, p.38-39].

---

**Algoritmo 4:** ALGORITMO DE NEWTON-RAPHSON MULTIDIMENSIONAL

---

**Entrada:**  $F(x)$ ,  $J(x)$  e uma aproximação inicial  $x_0 \in \mathbb{R}^n$ .

**Saída:**  $x_* \in \mathbb{R}^n$ , raiz de  $F(x)$ .

```
1 início
2    $k \leftarrow 0$  ;
3   se  $F(x_k) = 0$  então
4     Pare.
5   senão
6     Faça  $d_k = -(J(x_k))^{-1}F(x_k)$ ;
7     Faça  $x_{k+1} \leftarrow x_k + d_k$ ;
8     Faça  $k \leftarrow k + 1$ ;
9   fim
10 fim
11 retorna  $x_k$ 
```

---

---

# UMA INTRODUÇÃO AO PROBLEMA DE QUADRADOS MÍNIMOS

---

Neste capítulo, abordamos conceitos fundamentais acerca do problema de quadrados mínimos.

Na Seção 3.1, tratamos de alguns aspectos teóricos e uma interpretação geométrica do problema de quadrados mínimos linear. Na Seção 3.2, enfatizamos o estudo do caso não linear e suas particularidades. Já na Seção 3.3, são apresentados os métodos de Newton e Gauss-Newton para o problema de quadrados mínimos não linear. As principais referências deste capítulo são [1], [10], [16] e [19].

## 3.1 O problema de quadrados mínimos

O problema de quadrados mínimos é um caso particular de problema otimização, onde desejamos minimizar a função objetivo

$$f(x) = \frac{1}{2} \sum_{i=1}^m r_i^2(x), \quad (3.1)$$

onde cada  $r_i : \mathbb{R}^n \rightarrow \mathbb{R}$  é uma função suave, denominada residual. Neste capítulo assumimos que  $m \geq n$ .

Segundo Nocedal e Wright [19, p.247], problemas de quadrados mínimos surgem em diversas áreas de aplicações, dentre elas, Química, Física, Economia, entre outras; e podem ser, de fato, uma das maiores fontes de problemas de minimização irrestrita. Tais problemas surgem da necessidade de mensurar a discrepância entre os dados obtidos ex-

perimentalmente e o modelo teórico que descreve o sistema, um problema recorrente em atividades que envolvam modelagem. Neste sentido, ao minimizar a função (3.1), obtemos os parâmetros do modelo que melhor se adequam aos dados.

De modo a facilitar a compreensão do problema, considere o vetor residual  $R : \mathbb{R}^n \rightarrow \mathbb{R}^m$ , cujas entradas correspondem a cada um dos residuais  $r_i$ , considerados em (3.1),

$$R(x) = \begin{bmatrix} r_1(x) \\ r_2(x) \\ \vdots \\ r_m(x) \end{bmatrix}.$$

Deste modo, podemos reescrever (3.1) como

$$f(x) = \frac{1}{2} \|R(x)\|_2^2. \quad (3.2)$$

A matriz jacobiana  $J(x)$  de  $f$ , composta pelas derivadas parciais de primeira ordem dos residuais  $r_i$ ,  $i = 1, \dots, m$ , é dada por

$$J(x) = \left[ \frac{\partial r_i}{\partial x_j} \right]_{\substack{i=1, \dots, m \\ j=1, \dots, n}} = \begin{bmatrix} \nabla r_1(x)^T \\ \nabla r_2(x)^T \\ \vdots \\ \nabla r_m(x)^T \end{bmatrix}.$$

Assim, as derivadas primeira e segunda de  $f$  podem ser escritas em função de  $J(x)$ . De fato,

$$\nabla f(x) = \sum_{i=1}^m r_i(x) \nabla r_i(x) = J(x)^T R(x), \quad (3.3)$$

$$\begin{aligned} \nabla^2 f(x) &= \sum_{i=1}^m \nabla r_i(x) \nabla r_i(x)^T + \sum_{i=1}^m r_i(x) \nabla^2 r_i(x), \\ &= J(x)^T J(x) + S(x), \end{aligned} \quad (3.4)$$

onde  $S(x) = \sum_{i=1}^m r_i(x) \nabla^2 r_i(x)$ .

Frequentemente, obter as derivadas parciais dos residuais e a matriz jacobiana da função (3.2) é relativamente simples, obtendo facilmente o gradiente de  $f$  (3.3) e o termo  $J(x)^T J(x)$  de sua hessiana (3.4), sem a necessidade de calcularmos as segundas derivadas

de cada função  $r_i$ . Entretanto, o cálculo de  $\nabla^2 r_i(x)$ , necessário ao termo  $S(x)$  de (3.4), pode ser algo muito dispendioso, principalmente em problemas de grande porte. Felizmente, muitas vezes o termo  $J(x)^T J(x)$  é mais importante que  $S(x)$  em (3.4), uma vez que, ou os residuais têm comportamento semelhante a funções afim próximo a solução (isto é, quando  $\nabla^2 r_i$  são relativamente pequenos) ou porque os resíduos são muito pequenos (isto é,  $r_i(x)$  são relativamente pequenos) [19, p.246-247].

### 3.1.1 O caso linear

O problema de minimizar (3.1), ou sua forma equivalente (3.2), muitas vezes de difícil solução, pode se beneficiar das características inerentes aos seus residuais.

Um caso particular e bem delimitado, quando as funções  $r_i, i = 1, \dots, m$ , são todas lineares, é dito *problema de quadrados mínimos linear*. Nesta situação, a função  $R : \mathbb{R}^m \rightarrow \mathbb{R}^n$  pode ser escrita como  $R(x) = Ax - b$ , onde  $A \in \mathbb{R}^{m \times n}$  e  $b \in \mathbb{R}^m$ . Deste modo, obtemos o seguinte problema de minimização irrestrita

$$\min_{x \in \mathbb{R}^m} \|Ax - b\|_2. \quad (3.5)$$

Antes de prosseguirmos, considere as seguintes definições:

**Definição 3.1.** [17, p.170-171] Seja  $A \in \mathbb{R}^{m \times n}$  uma matriz real de dimensão  $m \times n$ . O *espaço imagem* ou *espaço coluna* de  $A$  é o subespaço vetorial  $Im(A) \subset \mathbb{R}^m$ , gerado pela imagem da função  $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$  dada por  $g(x) = Ax$ , isto é,  $Im(A) = \{Ax : x \in \mathbb{R}^n\} = span\{A_1, A_2, \dots, A_n\}$ , onde  $A_1, A_2, \dots, A_n$  são as colunas da matriz  $A$ .

**Definição 3.2.** [17, p.174] Seja  $A \in \mathbb{R}^{m \times n}$  uma matriz real de dimensão  $m \times n$ . O *núcleo* ou *espaço nulo* de  $A$  é o subespaço vetorial  $Nuc(A) \subset \mathbb{R}^n$  dado por  $Nuc(A) = \{x \in \mathbb{R}^n : Ax = 0\}$ . Em outras palavras, o núcleo de  $A$  é o conjunto de todas as soluções do sistema homogêneo  $Ax = 0$ .

Nas condições de (3.5), analisemos o sistema linear

$$Ax = b, \quad (3.6)$$

onde  $x \in \mathbb{R}^n$ .

O sistema linear (3.6) admite solução se e somente se  $b \in \text{Im}(A)$ , isto é,  $b$  deve ser escrito como combinação linear das colunas da matriz  $A$ . A Figura 3.1 nos fornece uma interpretação geométrica deste conceito no espaço  $\mathbb{R}^3$ .

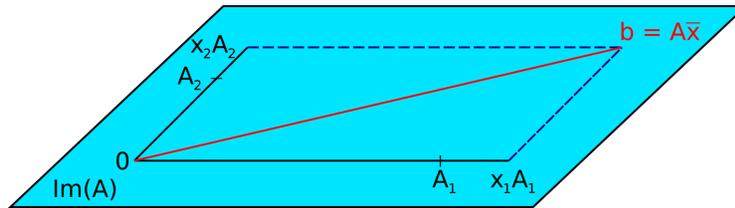


Figura 3.1: Solução do problema de quadrados mínimos linear, caso onde  $b \in \text{Im}(A)$ .

Assim, se  $b \in \text{Im}(A)$  e  $A$  é não singular, a solução é dada por  $x = A^{-1}b$ .

Caso  $b \notin \text{Im}(A)$ , nosso objetivo consiste em determinar  $\bar{x} \in \mathbb{R}^m$  tal que a distância entre  $A\bar{x}$  e  $b$  seja mínima.

Os resultados de Álgebra Linear enunciados a seguir estabelecem a existência da solução do problema (3.5).

**Lema 3.3.** *Um vetor  $x \in \mathbb{R}^m$  é solução do problema de quadrados mínimos (3.5) se e somente se satisfaz as chamadas equações normais, descritas abaixo:*

$$A^T Ax = A^T b. \quad (3.7)$$

**Demonstração:** Ver [1, Lema 7.2.1]. □

**Teorema 3.4.** *Para qualquer matriz  $A \in \mathbb{R}^{m \times n}$ , existe ao menos uma solução para as equações normais (3.7). Além disso, a solução é única se e somente se  $\text{Nuc}(A) = 0$ .*

**Demonstração:** Ver [1, Teorema 7.2.1]. □

Note que  $\text{Nuc}(A) = 0$  se e somente se  $A$  é matriz não singular.

O teorema a seguir estabelece uma relação entre o sistema linear (3.6) e a solução do problema (3.5):

**Teorema 3.5.** (*Teorema do Ponto mais Próximo*) Sejam  $\mathbb{V}$  um espaço vetorial com produto interno,  $\mathbb{M}$  um subespaço de  $\mathbb{V}$  e  $b \in \mathbb{V}$ . O único vetor em  $\mathbb{M}$  que está mais próximo de  $b$  é  $p = P_{\mathbb{M}}b$ , a projeção ortogonal de  $b$  sobre  $\mathbb{M}$ .

**Demonstração:** Ver [17, Teorema 5.13.13]. □

Aplicando o Teorema 3.5,  $\mathbb{M} = \text{Im}(A)$  e  $\mathbb{V} = \mathbb{R}^m$ , a solução  $\bar{x}$  de (3.5) é tal que  $A\bar{x}$  é a projeção ortogonal de  $b$  sobre  $\text{Im}(A)$ .

A Figura 3.2 ilustra o resultado anterior para o caso em que  $b \notin \text{Im}(A)$  e  $\mathbb{V} = \mathbb{R}^3$ .

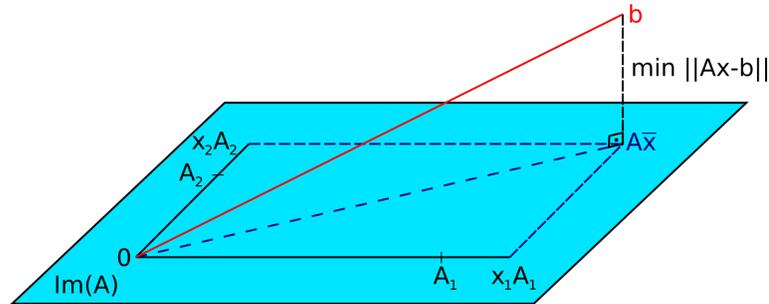


Figura 3.2: Solução do problema de quadrados mínimos linear, caso onde  $b \notin \text{Im}(A)$ .

Note que o problema de minimização irrestrita (3.5) é equivalente ao problema

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|Ax - b\|_2^2. \quad (3.8)$$

A Condição Necessária de 1ª Ordem (Teorema 2.22), afirma que:

$$\begin{aligned} \nabla \left( \frac{1}{2} \|Ax - b\|_2^2 \right) &= 0, \\ \nabla \left( \frac{1}{2} x^T A^T A x - x^T A^T b + \frac{1}{2} b^T b \right) &= 0, \\ A^T A x - A^T b &= 0, \\ A^T A x &= A^T b, \end{aligned}$$

e assim obtivemos as mesmas equações normais indicadas em (3.7). Portanto, pelo Teorema 3.4, tal problema sempre admite solução.

Conforme demonstrado por Benatti [5, p.51], independentemente do posto da matriz  $A$ , o problema (3.8) também satisfaz a Condição Necessária de 2ª Ordem (Teorema 2.23), uma vez que

$$\nabla^2 \left( \frac{1}{2} \|Ax - b\|_2^2 \right) = \nabla(A^T Ax - A^T b) = A^T A \geq 0,$$

e assim, a função objetivo é convexa. Portanto, todo ponto que satisfaz as equações normais (3.7) é minimizador do problema.

### 3.1.2 O caso não linear

Neste seção abordamos brevemente o problema de quadrados mínimos não linear, um caso muito mais amplo que o anterior. De fato, basta que qualquer uma das funções coordenadas do vetor residual  $R(x)$  não seja linear para que o problema se torne não linear.

Como discutido anteriormente, os problemas de quadrados mínimos surgem frequentemente da necessidade em se avaliar a discrepância entre os dados coletados e a previsão dada pelo modelo teórico do fenômeno. Não raro, tais modelos não são lineares. Deste modo, estamos interessados agora no problema (3.2), onde

$$R(x) = 0 \tag{3.9}$$

é um sistema não linear.

Note que tal sistema pode ou não ter solução. Caso exista solução  $x_* \in \mathbb{R}^n$  de (3.9), então há uma equivalência entre os problemas (3.9) e (3.2). Caso contrário, necessitaremos de métodos iterativos para resolver (3.2).

### 3.1.3 Métodos clássicos para o problema de quadrados mínimos não linear

Considere o problema de quadrados mínimos não linear:

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} \|R(x)\|_2^2. \tag{3.10}$$

Relembre que, por (3.3) e (3.4), podemos escrever

$$\begin{aligned}\nabla f(x) &= J(x)^T R(x), \\ \nabla^2 f(x) &= J(x)^T J(x) + S(x),\end{aligned}$$

onde  $J(x)$  é a matriz jacobiana de  $f$  e  $S(x)$  é um termo que depende das funções coordenadas de  $R(x)$  e de suas derivadas segundas.

### O Método de Newton

Considerando o Método de Newton discutido na subseção 2.2.4, a direção de Newton  $d_k^N$ , direção de minimização a partir de  $x_k$ , para o problema (3.10) satisfaz

$$(J(x_k)^T J(x_k) + S(x_k))d_k^N = -J(x_k)^T R(x_k). \quad (3.11)$$

Assim como discutido na abordagem inicial do Método de Newton, a direção  $d_k^N$  é de descida, e está unicamente determinada contanto que a hessiana  $\nabla^2 f(x)$  é definida positiva. Caso contrário, o sistema (3.11) admite infinitas soluções, porém elas não satisfazem, necessariamente, as condições dadas pela Definição 2.27 ou pelo Lema 2.28, as quais caracterizam as direções de descida a partir de  $x_k$ . De fato, caso  $\nabla^2 f(x_k)$  seja semidefinida positiva, a direção  $d_k^N$  pode ser ortogonal ao gradiente  $\nabla f(x_k)$ , e teríamos

$$\nabla f(x_k)^T d_k^N = -\nabla f(x_k)^T \nabla^2 f(x_k) \nabla f(x_k) = 0. \quad (3.12)$$

Conforme analisado no início da Seção 3.1, a determinação do termo  $S(x)$  pode ser algo muito dispendioso, principalmente em problemas de grande porte. Diante deste obstáculo, foram desenvolvidas diversas técnicas para aproximação desta matriz, gerando os chamados Métodos Quase-Newton. Basicamente, tais métodos podem ser definidos pela iteração

$$\begin{aligned}H_k d_k^{QN} &= \nabla f(x_k), \\ x_{k+1} &= x_k + d_k^{QN},\end{aligned} \quad (3.13)$$

onde  $H_k \in \mathbb{R}^{n \times n}$  geralmente tem boas propriedades, como simetria ( $H_k = H_k^T$ ), conforme [10, p.39] e [16, p.77].

Ainda neste intuito, podemos reduzir o custo computacional do problema (3.10) tomando uma aproximação linear para  $R(x)$ , derivando daí o Método de Gauss-Newton que será visto a seguir.

### O Método de Gauss-Newton

Novamente, considere o problema de minimização irrestrita dado em (3.10). Tomando a aproximação linear da função  $R(x)$ , dada pelo polinômio de Taylor de primeira ordem de  $R$  ao redor de  $x_k$ , temos que

$$R(x) \approx R(x_k) + J(x_k)d, \quad (3.14)$$

numa vizinhança de  $x_k$ , onde  $d = x - x_k$ .

Esperamos que, localmente,  $\|R(x)\|_2 \approx \|R(x_k) + J(x_k)d\|_2$ , e assim a solução do problema (3.10) pode ser aproximada através da solução de

$$\min_{d \in \mathbb{R}^n} \frac{1}{2} \|R(x_k) + J(x_k)d\|_2^2. \quad (3.15)$$

Pelo Teorema 2.22, a direção de minimização deve satisfazer

$$\frac{\partial}{\partial d} \left( \frac{1}{2} \|R(x_k) + J(x_k)d\|_2^2 \right) = 0.$$

Portanto,

$$\begin{aligned} \frac{\partial}{\partial d} \left( \frac{1}{2} (R(x_k)^T R(x_k) + 2d^T J(x_k)^T R(x_k) + d^T J(x_k)^T J(x_k)d) \right) &= 0, \\ J(x_k)^T R(x_k) + J(x_k)^T J(x_k)d &= 0, \\ J(x_k)^T J(x_k)d &= -J(x_k)^T R(x_k). \end{aligned} \quad (3.16)$$

A direção determinada por (3.16) é denominada de *direção de Gauss-Newton*  $d_k^{GN}$ .

O teorema a seguir estabelece um resultado de convergência para o Método de Gauss-Newton.

**Teorema 3.6.** *Seja  $x_0$  o ponto inicial de um algoritmo para o Método de Gauss-Newton. Suponha que o conjunto de nível  $\Gamma = \{x : f(x) \leq f(x_0)\}$  é limitado e que cada função*

residual  $r_i(x)$ ,  $i = 1, \dots, n$  é Lipschitz continuamente diferenciável em uma vizinhança  $\Omega$  de  $\Gamma$ . Considere ainda que existe uma constante  $\gamma > 0$  tal que  $\|J(x)z\|_2 \geq \gamma\|z\|_2$  para todo  $x \in \Omega$ . Se as iterações  $x_k$  são geradas pelo Método de Gauss-Newton com passo  $\alpha_k$  satisfazendo

$$i) f(x_k + \alpha_k d_k^{GN}) \leq f(x_k) + c_1 \alpha_k \nabla f(x_k)^T d_k^{GN},$$

$$ii) \nabla f(x_k + \alpha_k d_k^{GN})^T d_k^{GN} \geq c_2 \nabla f(x_k)^T d_k^{GN},$$

com  $0 < c_1 < c_2 < 1$ , então

$$\lim_{k \rightarrow \infty} J(x_k)^T R(x_k) = 0.$$

**Demonstração:** Ver [19, Teorema 10.1].

□

---

# O MÉTODO DE LEVENBERG-MARQUARDT

---

Dando continuidade ao estudo de métodos iterativos para problemas de quadrados mínimos não lineares, apresentamos neste capítulo conceitos elementares acerca do Método de Levenberg-Marquardt.

Este método de otimização foi publicado inicialmente em 1944 (ver [13]) pelo estatístico americano Kenneth Levenberg (1919-1973), enquanto ainda trabalhava no Frankford Arsenal, uma antiga fábrica de munições do Exército dos Estados Unidos da América. Posteriormente, o algoritmo foi redescoberto em 1963 (ver [15]) pelo também estatístico Donald Marquardt (1929-1997), o qual empregou o método para resolver problemas de ajuste de modelos químicos não lineares de dados laboratoriais da gigante multinacional DuPont [4, p.3].

Na Seção 4.1 são apresentados o Método de Levenberg-Marquardt, alguns resultados teóricos e um algoritmo clássico. Na seção seguinte, é desenvolvida uma estratégia de construção de parâmetros de *damping*, são apresentados alguns parâmetros conhecidos na literatura, assim como são propostos dois novos parâmetros de *damping* e uma estratégia híbrida. Por fim, na Seção 4.3 propomos uma variação do algoritmo de Levenberg-Marquardt com Busca Linear, assim como apresentamos resultados de convergência. As principais referências deste capítulo são [5], [7] [15] e [19].

## 4.1 O Método de Levenberg-Marquardt para problemas de quadrados mínimos não lineares

Inicialmente, seja  $R : \mathbb{R}^n \rightarrow \mathbb{R}^m$  uma função residual e considere o problema de quadrados mínimos não linear

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} \|R(x)\|_2^2. \quad (4.1)$$

Dado um ponto inicial  $x_0$ , o método de Levenberg-Marquardt aplicado ao problema (4.1) visa encontrar, a cada iteração  $k$ , uma direção  $d(\lambda_k) \in \mathbb{R}^n$  que seja solução do sistema linear

$$(J(x_k)^T J(x_k) + \lambda_k I) d(\lambda_k) = -J(x_k)^T R(x_k), \quad (4.2)$$

onde  $J(x) \in \mathbb{R}^{m \times n}$  é a matriz jacobiana de  $R(x)$  e  $\lambda \in \mathbb{R}_+$  é uma constante denominada parâmetro de Levenberg-Marquardt ou parâmetro de *damping*. A direção de Levenberg-Marquardt  $d(\lambda)$  também será denotada por  $d^{LM}$ .

Note que, diferentemente do que acontece no método de Gauss-Newton, o qual pode não estar bem definido caso a matriz  $J(x_k)^T J(x_k)$  não seja definida positiva, se  $\lambda_k > 0$ , a correção  $\lambda_k I$  aplicada na diagonal de  $J(x_k)^T J(x_k)$ , a torna não singular. Sendo assim, em cada iteração  $k$  o sistema linear (4.2) tem solução e esta é única. Consequentemente, o método está bem definido.

**Lema 4.1.** *Sejam  $A \in \mathbb{R}^{m \times n}$  e  $\lambda \in \mathbb{R}_+^*$ . A matriz  $A^T A + \lambda I$  é definida positiva.*

**Demonstração:** Como

$$v^T A^T A v = (Av)^T (Av) = \|Av\|_2^2 \geq 0$$

para todo vetor  $v \in \mathbb{R}^n$ ,  $A^T A$  é semidefinida positiva. Seja  $\lambda \in \mathbb{R}_+^*$  fixado. Assim, temos que

$$v^T (A^T A + \lambda I) v = v^T A^T A v + \lambda v^T v = \|Av\|_2^2 + \lambda \|v\|_2^2 > 0$$

para todo vetor não-nulo  $v \in \mathbb{R}^n$ . Portanto, a matriz  $A^T A + \lambda I$  é definida positiva. □

Uma outra forma de interpretarmos o método de Levenberg-Marquardt segue. Seja  $S(x) = \sum_{i=1}^m r_i(x) \nabla^2 r_i(x)$  e considere o modelo quadrático da função objetivo  $f(x)$ , em uma vizinhança do ponto  $x_k$ , aproximando sua matriz hessiana

$$\nabla^2 f(x) = \sum_{i=1}^m \nabla r_i(x) \nabla r_i(x)^T + \sum_{i=1}^m r_i(x) \nabla^2 r_i(x) = J(x)^T J(x) + S(x),$$

por

$$\nabla^2 f(x_k) \approx J(x_k)^T J(x_k) + \lambda_k I.$$

Obtendo-se assim o modelo

$$m_k(d) := \frac{1}{2} R(x_k)^T R(x_k) + J(x_k)^T R(x_k) d + \frac{1}{2} (J(x_k)^T J(x_k) + \lambda_k I). \quad (4.3)$$

Observe que, ao minimizar (4.3) obtemos a mesma direção  $d(\lambda_k)$  apontada em (4.2).

Vale salientar ainda que se  $\lambda_k = 0$ , a direção  $d^{LM}$  obtida pelo método de Levenberg-Marquardt coincide com a direção de Gauss-Newton  $d^{GN}$ , definida em (3.16). Neste caso, se  $J(x_k)^T J(x_k)$  é uma boa aproximação para a matriz hessiana, então o método apresentará taxa de convergência quadrática, partindo de um ponto suficientemente próximo da solução. Entretanto, se  $\lambda_k$  tender ao infinito, teremos que a direção  $d^{LM}$  tenderá à direção de máxima descida  $-\nabla f(x_k)$ , a qual denotaremos por  $-g_k$ , porém sua norma tenderá a zero, conforme estabelecido nos teoremas abaixo.

**Teorema 4.2.** *Seja  $d(\lambda)$  a solução de (4.2) para um dado valor  $\lambda$ . Então  $\|d(\lambda)\|_2^2$  é uma função contínua decrescente na variável  $\lambda$ , tal que  $\lim_{\lambda \rightarrow \infty} \|d(\lambda)\|_2^2 = 0$ .*

**Demonstração:** Ver [15, Teorema 2]. □

**Teorema 4.3.** *Seja  $\gamma(\lambda)$  o ângulo entre  $d(\lambda)$  (direção de Levenberg-Marquardt) e  $-g$  (direção de máxima descida). Então  $\gamma(\lambda)$  é uma função contínua monótona decrescente na variável  $\lambda$  tal que  $\lim_{\lambda \rightarrow \infty} \gamma(\lambda) = 0$ . Além disso,  $d(\lambda)$  rotaciona na direção  $-g$  quando  $\lambda \rightarrow \infty$ .*

**Demonstração:** Ver [15, Teorema 3]. □

Em seguida, apresentamos um algoritmo para o Método de Levenberg-Marquardt Clássico, baseado em [15, p.438], e um resultado de convergência para o método.

**Algoritmo 5:** LEVENBERG-MARQUARDT CLÁSSICO

**Entrada:**  $R(x)$ ,  $J(x)$ , uma aproximação inicial  $x_0 \in \mathbb{R}^n$ , um valor inicial  $\lambda_0 \in \mathbb{R}_+$ , parâmetros de atualização  $\kappa_1, \kappa_2 > 1$ , uma tolerância  $\varepsilon > 0$ .

**Saída:**  $x_* \in \mathbb{R}^n$ , ponto estacionário de  $f(x) = \frac{1}{2} \|R(x)\|_2^2$ .

1 **início**

2  $k \leftarrow 0$ ;

3 **enquanto**  $J(x_k)^T R(x_k) > \varepsilon$  **faça**

4     Obtenha  $d(\lambda_k)$  solução de:

$$(J(x_k)^T J(x_k) + \lambda_k I) d(\lambda_k) = -J(x_k)^T R(x_k);$$

5     Faça  $x_n = x_k + d(\lambda_k)$ ;

6     **se**  $R(x_n) < R(x_k)$  **então**

7          $x_{k+1} = x_n$ ;

8          $\lambda_{k+1} = \frac{\lambda_k}{\kappa_1}$ .

9     **senão**

10          $\lambda_{k+1} = \lambda_k \kappa_2$ .

11     **fim**

12     Faça:

$k \leftarrow k + 1$ .

13 **fim**

14 **fim**

15 **retorna**  $x_k$

**Lema 4.4.** *A direção de Levenberg-Marquardt  $d^{LM}$  é uma solução do subproblema de região de confiança*

$$\min_d \frac{1}{2} \|J(x)d + R(x)\|_2^2, \text{ sujeito a } \|d\|_2 \leq \Delta, \quad (4.4)$$

*se e somente se é factível e existe uma constante  $\lambda \geq 0$  tal que*

i)  $(J(x)^T J(x) + \lambda I) d^{LM} = -J(x)^T R(x)$ ,

ii)  $\lambda(\Delta - \|d^{LM}\|_2) = 0$ .

**Demonstração:** Ver [19, Lema 10.2]. □

**Teorema 4.5.** *Sejam  $x_0 \in \mathbb{R}^n$  o ponto inicial do Algoritmo 5,  $\eta \in (0, \frac{1}{4})$ ,  $\mu > 0$  e  $\gamma \geq 1$ . Suponha que o conjunto de nível  $\Gamma = \{x : f(x) \leq f(x_0)\}$  é limitado e que cada função residual  $r_i(x)$ ,  $i = 1, \dots, n$  é Lipschitz continuamente diferenciável em uma vizinhança  $\Omega$  de  $\Gamma$ . Considere o polinômio  $m_k$  dado em (4.3) e assuma que, para cada iteração  $k$ , a solução aproximada  $d_k$  do problema (4.4) satisfaz*

$$i) f(x_k) - f(x_k + d_k) > \eta(m_k(0) - m_k(d_k)),$$

$$ii) m_k(0) - m_k(d_k) \geq \mu \|J(x_k)^T R(x_k)\|_2 \min \left( \Delta_k, \frac{\|J(x_k)^T R(x_k)\|_2}{\|J(x_k)^T J(x_k)\|_2} \right),$$

$$iii) \|d_k\|_2 \leq \gamma \Delta_k.$$

Deste modo,

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

**Demonstração:** Ver [19, Teorema 10.3]. □

## 4.2 O parâmetro de *damping*

Conforme visto no início deste capítulo, a estratégia inovadora do Método de Levenberg-Marquardt consiste essencialmente na “correção” da singularidade da matriz  $J(x_k)^T J(x_k)$  através da adição de um valor  $\lambda_k > 0$ , chamado parâmetro de *damping*, em sua diagonal principal. Entretanto, caso a matriz  $J(x_k)^T J(x_k)$  seja definida positiva, basta resolver o sistema  $(J(x_k)^T J(x_k))d_k = -J(x_k)^T R(x_k)$ , obtendo-se a direção  $d^{GN}$  do Método de Gauss-Newton.

Computacionalmente, uma das maneiras mais eficientes para se determinar a positividade de uma matriz é aplicar o processo de Decomposição de Cholesky, o qual, como visto no Teorema 2.14, somente é aplicado para matrizes definidas positivas. Caso a decomposição falhe, deve-se encontrar  $\lambda_k > 0$  tal que  $J(x_k)^T J(x_k) + \lambda_k I$  seja definida positiva.

Pelos Teoremas (4.2) e (4.3), o parâmetro de *damping* tem influência direta na determinação da direção  $d(\lambda)$  obtida pelo método de Levenberg-Marquardt e, por consequência, na convergência do algoritmo.

Assim, é crucial determinar boas escolhas para  $\lambda$ , de modo que a cada iteração, o parâmetro  $\lambda_k$  escolhido seja suficientemente grande, reduzindo o valor da função objetivo e, simultaneamente, o passo tomado não seja muito pequeno, garantindo a convergência da sequência. Neste sentido, adotaremos as estratégias apresentadas por Benatti [5] e Davis e Whitting [7], as quais discutiremos a seguir.

### 4.2.1 Estratégias para a escolha do parâmetro de *damping*

Inicialmente, considere o caso em que a função  $R(x)$  é zero residual, isto é, existe  $x \in \mathbb{R}^n$  tal que  $R(x) = 0$ . Defina  $\lambda_k = \frac{1}{\omega_k}$ , para algum  $\omega_k > 0$ . De modo a simplificar a notação, o índice  $k$  será omitido no resto desta seção.

Defina a função

$$\phi(\omega) = f(x + d(\omega)). \quad (4.5)$$

satisfazendo

$$\left( J(x)^T J(x) + \frac{1}{\omega} D \right) d(\omega) = -J(x)^T R(x), \quad (4.6)$$

onde  $f(x) = \frac{1}{2} \|R(x)\|^2$  e  $D = \text{diag}(\alpha_1, \alpha_2, \dots, \alpha_n)$ , com  $\alpha_i > 0$ ,  $i = 1, \dots, n$ , constantes reais positivas quaisquer. Nosso objetivo será determinar  $\omega$ , raiz da função  $\phi(\omega)$ , satisfazendo (4.6).

Contudo, a equação (4.6) não está definida para  $\omega = 0$ . Deste modo, tomaremos a seguinte equação em substituição a anterior, a qual não apresenta a restrição  $\omega \neq 0$ .

$$\left( \omega (J(x)^T J(x)) + D \right) d(\omega) = -\omega (J(x)^T R(x)) \quad (4.7)$$

Por (4.6) e (4.7), conclui-se  $d(0) = 0$ . Note ainda que, de fato, a função (4.5) possui um zero, dado que  $R(x)$  é zero residual e, conseqüentemente,  $f(x)$  também o é. Nossa estratégia consistirá em aplicar o Método de Newton-Raphson, visto na Subseção 2.2.5, para determinar um zero da função  $\phi(\omega)$ .

Portanto, considere o polinômio de Maclaurin de grau 1 da função  $\phi(\omega)$ , apresentado na Subseção 2.2.2,

$$\psi(\omega) = \phi(0) + \phi'(0)\omega. \quad (4.8)$$

Derivando a expressão (4.5) e substituindo em  $\omega = 0$ , obtemos:

$$\begin{aligned} \phi'(\omega) &= \nabla f(x + d(\omega))^T d'(\omega), \\ \phi'(0) &= \nabla f(x + d(0))^T d'(0) = \nabla f(x)^T d'(0), \\ \phi'(0) &= (J(x)^T R(x)) d'(0). \end{aligned} \quad (4.9)$$

Por outro lado, derivando implicitamente (4.7) e substituindo em  $\omega = 0$ , temos:

$$\begin{aligned} (J(x)^T J(x)) d(\omega) + (\omega(J(x)^T J(x)) + D) d'(\omega) &= -(J(x)^T R(x)), \\ (\omega(J(x)^T J(x)) + D) d'(\omega) &= -J(x)^T R(x) - (J(x)^T J(x)) d(\omega), \\ D d'(0) &= -J(x)^T R(x) - (J(x)^T J(x)) d(0), \\ D d'(0) &= -J(x)^T R(x). \end{aligned} \quad (4.10)$$

Considere  $v = -d'(0)$ . Note que  $v$  depende explicitamente da matriz diagonal  $D$ , uma vez que  $d'(0)$  é solução do sistema (4.10).

Dado que as entradas diagonais  $[D]_{ii} = \alpha_i$ ,  $i = 1, \dots, n$ , de  $D$  são positivas; se  $(J(x)^T R(x))_i = 0$ , podemos tomar  $[v]_i = \frac{[J(x)^T R(x)]_i}{\alpha_i}$ . Assim, por (4.9) e (4.10), o zero da função  $\psi(\omega)$  satisfaz

$$\begin{aligned} \phi(0) + \phi'(0)\omega &= 0, \\ f(x) + \omega(J(x)^T R(x))^T d'(0) &= 0, \\ f(x) - \omega(J(x)^T R(x))^T v &= 0, \\ \omega(J(x)^T R(x))^T v &= f(x), \\ \omega &= \frac{f(x)}{(J(x)^T R(x))^T v}. \end{aligned}$$

Como  $\lambda = \frac{1}{\omega}$ , a equação acima equivale a

$$\lambda = \frac{(J(x)^T R(x))^T v}{f(x)}. \quad (4.11)$$

Considere agora o caso em que  $R(x)$  não é zero residual. De modo análogo ao anterior, considere a função  $\phi(\omega) = f(x + d(\omega))$ , com a restrição  $(J(x)^T J(x) + \frac{1}{\omega} D)d(\omega) = -J(x)^T R(x)$ . Contudo, ao invés do Método de Newton-Raphson, aplicaremos o Método de Cauchy apresentado na Subseção 2.2.3.

Deste modo, dado um certo  $\omega$ , uma nova aproximação para o minimizador de  $\phi$  é dado por

$$\begin{aligned}\omega^+ &= \omega - t\phi'(\omega), \\ &= \omega - t\nabla f(x + d(\omega))^T d'(\omega),\end{aligned}\tag{4.12}$$

onde  $t > 0$  é o tamanho do passo a ser dado na direção oposta ao gradiente de  $\phi$ . Substituindo  $\omega = 0$  em (4.12), temos

$$\omega^+ = -t\nabla f(x)^T d'(0) = t\nabla f(x)^T v = t(J(x)^T R(x))^T v.$$

Deste modo,

$$\lambda = \frac{1}{\omega^+} = \frac{1}{t(J(x)^T R(x))^T v}.\tag{4.13}$$

Assim como reitera Benatti ([5], p.62), apesar das expressões (4.11) e (4.13) parecem totalmente distintas, a expressão  $\lambda$  para o caso em que  $R(x)$  é função zero residual, é um caso particular da equação (4.13), obtida tomando o passo  $t$  como sendo

$$t = \frac{f(x)}{\left((J(x)^T R(x))^T v\right)^2}.$$

Deste modo, dada a liberdade na escolha de  $t$ , podemos utilizar (4.11) para ambos os casos, sendo  $R(x)$  função zero residual ou não.

Tendo por base a expressão (4.13), diversos autores propuseram novas formas de calcular o parâmetro de *damping*, ao explorar diferentes alternativas para a matriz de escalamento  $D_k$  e, por conseguinte, diferentes escolhas de  $v_k$ .

Na Tabela 4.1, indicamos alguns parâmetros de *damping* presentes na literatura, como [5, 9, 13, 23], por exemplo. Tais parâmetros serão futuramente empregados nos testes numéricos presentes no Capítulo 7.

Sigla	Autoria	Parâmetro de <i>damping</i>
PD1	Levenberg, ver [13]	$\lambda_k = \frac{\ J(x_k)^T R(x_k)\ _2^2}{f(x_k)}$
PD2	Levenberg modificado, ver [13], [5]	$\lambda_k = \ J(x_k)^T R(x_k)\ _2^2$
PD3	Levenberg modificado 2, ver [13], [5]	$\lambda_k = \ J(x_k)^T R(x_k)\ _2$
PD4	Yamashita e Fukushima, ver [23]	$\lambda_k = \ R(x_k)\ _2^2 = 2f(x)$
PD5	Fan e Yuan, ver [9]	$\lambda_k = \ R(x_k)\ _2 = \sqrt{2f(x)}$
PD6	Benatti, ver [5]	$\lambda_k = \frac{2\ J(x_k)^T R(x_k)\ _2}{3k}$

Tabela 4.1: Parâmetros de *damping* presentes na literatura.

Como nossa contribuição, também analisamos o desempenho numérico de nossas estimativas para o parâmetro de *damping*, apresentadas na Tabela 4.2:

Sigla	Autoria	Parâmetro de <i>damping</i>
PD7	Schwertner 01	$\lambda_k = \frac{2\sqrt{f(x_k)}}{3k}$
PD8	Schwertner 02	$\lambda_k = \frac{\ R(x_k)\ _2^2}{3k} = \frac{2f(x)}{3k}$

Tabela 4.2: Parâmetros de *damping* propostos.

Conforme visto no início deste capítulo, a cada iteração do método de Levenberg-Marquardt, o parâmetro de *damping*  $\lambda_k$  é um valor real positivo. Contudo, tal valor pode se aproximar arbitrariamente de zero, de modo que a matriz  $J(x_k)^T J(x_k) + \lambda_k I$ , apesar de definida positiva (Lema 4.1), pode apresentar mal condicionamento. Deste modo, propomos as seguintes estratégias híbridas baseadas em PD7 e PD8, conforme descrito na Tabela 4.3.

Sigla	Autoria	Parâmetro de <i>damping</i>
PD7H1	Schwertner 01, híbrido 01	$\lambda_k = \max \left\{ \frac{2\sqrt{f(x_k)}}{3k}, 10^{-2} \right\}$
PD7H2	Schwertner 01, híbrido 02	$\lambda_k = \max \left\{ \frac{2\sqrt{f(x_k)}}{3k}, 1 \right\}$
PD7H3	Schwertner 01, híbrido 03	$\lambda_k = \max \left\{ \frac{2\sqrt{f(x_k)}}{3k}, 10 \right\}$
PD8H1	Schwertner 02, híbrido 01	$\lambda_k = \max \left\{ \frac{2f(x)}{3k}, 10^{-2} \right\}$
PD8H2	Schwertner 02, híbrido 02	$\lambda_k = \max \left\{ \frac{2f(x)}{3k}, 1 \right\}$
PD8H3	Schwertner 02, híbrido 03	$\lambda_k = \max \left\{ \frac{2f(x)}{3k}, 10 \right\}$

Tabela 4.3: Parâmetros de *damping* híbridos.

### 4.3 Algoritmo de Levenberg-Marquardt com Busca Linear

Baseando-se na estratégia apresentada na seção anterior, podemos reestruturar o Algoritmo 5 (p.37), adequando-o a esta nova forma de atualização do parâmetro de *damping*. Além disso, podemos adotar a Condição de Armijo (Teorema 2.29), a qual assegura que haja um decréscimo monótono no valor da função objetivo, tornando-o de certa forma, proporcional ao tamanho do passo.

**Algoritmo 6: LEVENBERG-MARQUARDT COM BUSCA LINEAR**

**Entrada:**  $R(x)$ ,  $J(x)$ , uma aproximação inicial  $x_0 \in \mathbb{R}^n$ , uma constante  $\alpha \in (0, 1)$ ,  
uma tolerância  $\varepsilon > 0$ .

**Saída:**  $x_* \in \mathbb{R}^n$ , ponto estacionário de  $f(x) = \frac{1}{2} \|R(x)\|_2^2$ .

1 **início**

2  $k \leftarrow 0$ ;

3 **enquanto**  $J(x_k)^T R(x_k) > \varepsilon$  **faça**

4     Calcule  $\lambda_k$ ;

5     Obtenha  $d(\lambda_k)$  solução de:

$$(J(x_k)^T J(x_k) + \lambda_k I) d(\lambda_k) = -J(x_k)^T R(x_k); \quad (4.14)$$

6     Faça  $t = 1$ ;

7     **enquanto**  $f(x_k + td_k) > f(x_k) + \alpha t \nabla f(x_k)^T d_k$  **faça**

8         Escolha novo  $t \in [0.1t, 0.9t]$ .

9     **fim**

10     Faça:

$$x_{k+1} \leftarrow x_k + td(\lambda_k)$$

$$k \leftarrow k + 1$$

11 **fim**

12 **fim**

13 **retorna**  $x_k$

Para que possamos garantir a convergência do Algoritmo 6, precisamos considerar as seguintes hipóteses:

**Hipótese 4.6.** A escolha de  $\lambda$  é limitada inferiormente e superiormente por constantes não-negativas, isto é, existem constantes  $L_{min}, L_{max} \in \mathbb{R}_+$ , com  $L_{min} \leq L_{max}$ , tais que  $\lambda \in [L_{min}, L_{max}]$ .

**Hipótese 4.7.** Seja  $\sigma_1(x)$  o maior autovalor da matriz  $J(x)^T J(x)$ . Então, existe  $M \in \mathbb{R}$  tal que  $\sigma_1(x) < M$ , para todo  $x \in \mathbb{R}^n$ .

**Hipótese 4.8.** O ângulo  $\gamma_x(\lambda)$  entre a direção de Levenberg-Marquardt e a direção de máxima descida é limitado superiormente por um constante positiva  $\bar{\gamma}$  tal que  $0 \leq \gamma_x(\lambda) \leq \bar{\gamma} < \frac{\pi}{2}$ , para todo  $\lambda \in [L_{min}, L_{max}]$  e para todo  $x \in \mathbb{R}^n$ .

Os exemplos 4.9 e 4.10 mostram que a adoção das Hipóteses (4.6), (4.7) e (4.8) não resulta num conjunto vazio de problemas.

**Exemplo 4.9.** Considere o problema irrestrito de minimizar  $f(x) = \frac{1}{2}\|R(x)\|_2^2$ , onde

$$R(x) = \begin{bmatrix} 2x_1 + 1 & 0 \\ 0 & 2x_2 + 1 \end{bmatrix}.$$

Logo  $J(x) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$  e  $\nabla f(x) = J(x)^T R(x) = \begin{bmatrix} 4x_1 + 2 \\ 4x_2 + 2 \end{bmatrix}$ . Como  $J(x)^T J(x)$  é não singular, podemos tomar  $L_{min} = 0$  e  $L_{max} \in \mathbb{R}_+$  qualquer. O maior autovalor da matriz  $J(x)^T J(x)$  é 4, assim basta tomar  $M > 4$  qualquer. Agora,

$$\begin{aligned} (J(x)^T J(x) + \lambda I)d(\lambda) &= -J(x)^T R(x), \\ \begin{bmatrix} 4 + \lambda & 0 \\ 0 & 4 + \lambda \end{bmatrix} \begin{bmatrix} d_1(\lambda) \\ d_2(\lambda) \end{bmatrix} &= \begin{bmatrix} -4x_1 - 2 \\ -4x_2 - 2 \end{bmatrix}, \\ \begin{bmatrix} d_1(\lambda) \\ d_2(\lambda) \end{bmatrix} &= \begin{bmatrix} \frac{1}{4+\lambda} & 0 \\ 0 & \frac{1}{4+\lambda} \end{bmatrix} \begin{bmatrix} -4x_1 - 2 \\ -4x_2 - 2 \end{bmatrix}, \\ \begin{bmatrix} d_1(\lambda) \\ d_2(\lambda) \end{bmatrix} &= \begin{bmatrix} \frac{-4x_1 - 2}{4+\lambda} \\ \frac{-4x_2 - 2}{4+\lambda} \end{bmatrix}. \end{aligned}$$

Desse modo  $d(\lambda) = -\frac{1}{4+\lambda} J(x)^T R(x)$ . Assim,

$$\cos \gamma_x(\lambda) = \frac{-\nabla f(x)^T d(\lambda)}{\|\nabla f(x)\|_2 \|d(\lambda)\|_2} = \frac{\frac{1}{4+\lambda} \nabla f(x)^T \nabla f(x)}{\|\nabla f(x)\|_2 \|\frac{1}{4+\lambda} \nabla f(x)\|_2} = \frac{\frac{1}{4+\lambda} \|\nabla f(x)\|_2^2}{\frac{1}{4+\lambda} \|\nabla f(x)\|_2^2} = 1.$$

Portanto,  $\gamma_x(\lambda) = 0$ , para todo  $x \in \mathbb{R}^n$ . Consequentemente, basta tomar  $\bar{\gamma}$  tal que  $0 \leq \bar{\gamma} < \pi/2$ .

**Exemplo 4.10.** Considere o problema irrestrito de minimizar  $f(x) = \frac{1}{2}\|R(x)\|_2^2$ , com  $R(x) = Ax + b$ , onde  $A \in \mathbb{R}^{n \times n}$  e  $b \in \mathbb{R}^n$ . Neste caso,  $J(x) = A$  e

$$\nabla f(x) = J(x)^T R(x) = A^T[Ax + b] = A^T Ax + A^T b$$

Se  $A$  tem posto completo, isto é,  $\text{posto}(A) = n$ , podemos tomar  $L_{\min} = 0$  e  $L_{\max} \in \mathbb{R}_+$ . Caso contrário, basta tomar  $L_{\min} > 0$  e  $L_{\max} \geq L_{\min}$ . Seja  $\mu_{\max}$  o maior autovalor de  $A^T A$ . Assim, basta tomar  $M > \mu_{\max}$ . Como  $A^T A$  é normal e simétrica, existe uma matriz ortogonal  $Q$  tal que  $A^T A = QDQ^T$ , onde  $D$  é uma matriz diagonal com entradas diagonais positivas. Desta forma,

$$\begin{aligned} d(\lambda) &= -(J(x)^T J(x) + \lambda I)^{-1} J(x)^T R(x), \\ &= -(QDQ^T + \lambda I)^{-1} J(x)^T R(x), \\ &= -Q(D + \lambda I)^{-1} Q^T J(x)^T R(x), \\ &= -Q(D + \lambda I)^{-1} v, \end{aligned}$$

onde  $v = Q^T J(x)^T R(x)$ .

Considere as funções  $\mu_{\min}, \mu_{\max} : \mathcal{M}(\mathbb{R}) \rightarrow \mathbb{R}$  que associam cada matriz real ao seu menor e maior autovalores, respectivamente. Note ainda que

$$\|d(\lambda)\|_2 = \|(D + \lambda I)^{-1} J(x)^T R(x)\|_2 \text{ e}$$

$$\|v\|_2 = \|J(x)^T R(x)\|_2,$$

pois  $Q$  é uma matriz ortogonal. Assim, pela definição de  $\gamma_x(\lambda)$ , o Teorema de Courant-Fischer (Teorema 2.8), afirma que

$$\begin{aligned} \cos \gamma_x(\lambda) &= \frac{-d(\lambda)^T \nabla f(x)}{\|d(\lambda)\|_2 \|\nabla f(x)\|_2} = \frac{v^T (D + \lambda I)^{-1} Q^T J(x)^T R(x)}{\|d(\lambda)\|_2 \|J(x)^T R(x)\|_2} = \frac{v^T (D + \lambda I)^{-1} v}{\|d(\lambda)\|_2 \|v\|_2}, \\ &\geq \frac{\mu_{\min} \left( (D + \lambda I)^{-1} \right) \|v\|_2}{\|(D + \lambda I)^{-1} J(x)^T R(x)\|_2} \geq \frac{\mu_{\min} \left( (D + \lambda I)^{-1} \right) \|v\|_2}{\|(D + \lambda I)^{-1}\|_2 \|J(x)^T R(x)\|_2} \geq \frac{\mu_{\min} \left( (D + \lambda I)^{-1} \right)}{\|(D + \lambda I)^{-1}\|_2}, \\ &\geq \frac{\mu_{\min} \left( (D + \lambda I)^{-1} \right)}{\mu_{\max} \left( (D + \lambda I)^{-1} \right)}. \end{aligned}$$

(4.15)

Aplicando o Teorema 2.6,

$$\begin{aligned}\mu_{\min}\left((D + \lambda I)^{-1}\right) &= \frac{1}{\mu_{\max}\left((D + \lambda I)\right)} = \frac{1}{\mu_{\max}(D) + \lambda}, \\ \mu_{\max}\left((D + \lambda I)^{-1}\right) &= \frac{1}{\mu_{\min}\left((D + \lambda I)\right)} = \frac{1}{\mu_{\min}(D) + \lambda}.\end{aligned}\tag{4.16}$$

Substituindo (4.16) na desigualdade (4.15), obtemos

$$\cos \gamma_x(\lambda) \geq \frac{\mu_{\min}\left((D + \lambda I)^{-1}\right)}{\mu_{\max}\left((D + \lambda I)^{-1}\right)} = \frac{\mu_{\min}(D) + \lambda}{\mu_{\max}(D) + \lambda} > 0,$$

uma vez que todos os autovalores de  $D$  são positivos. Portanto,  $\gamma_x(\lambda) \neq \pi/2$ , para todo  $x \in \mathbb{R}^n$ , para todo  $\lambda \in [L_{\min}, L_{\max}]$ . Deste modo, existe  $\bar{\gamma}$  suficiente próximo de  $\pi/2$ , tal que  $0 \leq \gamma_x(\lambda) \leq \bar{\gamma} < \pi/2$ , para todo  $x \in \mathbb{R}^n$ , para todo  $\lambda \in [L_{\min}, L_{\max}]$ .

**Lema 4.11.** *Suponha que o Algoritmo (6) seja implementado de modo a satisfazer as Hipóteses (4.6), (4.7) e (4.8). Então existem constantes  $\beta > 0$  e  $\theta \in (0, 1)$  tais que:*

$$\|d(\lambda_k)\|_2 \geq \beta \|\nabla f(x_k)\|_2,$$

$$\nabla f(x_k)^T d(\lambda_k) \leq -\theta \|\nabla f(x_k)\|_2 \|d(\lambda_k)\|_2.$$

**Demonstração:** De fato, como  $\nabla f(x) = J(x)^T R(x)$ , por (4.14),

$$(J(x)^T J(x) + \lambda I)d(\lambda) = -\nabla f(x),$$

$$\|J(x)^T J(x) + \lambda I\|_2 \|d(\lambda)\|_2 \geq \|\nabla f(x)\|_2,$$

$$\begin{aligned}\|d(\lambda)\|_2 &\geq \frac{1}{\|J(x)^T J(x) + \lambda I\|_2} \|\nabla f(x)\|_2, \\ &\geq \frac{1}{\|J(x)^T J(x)\|_2 + \lambda} \|\nabla f(x)\|_2.\end{aligned}$$

Pela definição da  $2$ -Norma, temos

$$\|d(\lambda)\|_2 \geq \frac{1}{\sigma_1(x) + \lambda} \|\nabla f(x)\|_2.$$

As Hipóteses (4.6) e (4.7) garantem que

$$\begin{aligned}\|d(\lambda)\|_2 &\geq \frac{1}{M + \lambda} \|\nabla f(x)\|_2 \\ &\geq \frac{1}{M + L_{\max}} \|\nabla f(x)\|_2\end{aligned}$$

Portanto, basta tomar  $\beta = \frac{1}{M + L_{max}}$ .

Pela definição de  $\gamma_x(\lambda)$ , temos que

$$\cos \gamma_x(\lambda) = \frac{-\nabla f(x)^T d(\lambda)}{\|\nabla f(x)\|_2 \|d(\lambda)\|_2}.$$

Pelo Teorema 4.3 e pela Hipótese (4.6), vale que  $\gamma_x(\lambda) \leq \gamma_x(L_{min})$ , para todo  $\lambda \in [L_{min}, L_{max}]$  e todo  $x \in \mathbb{R}^n$ . Desse modo, como a função cosseno é contínua e decrescente no intervalo  $[0, \pi/2]$ , a Hipótese (4.8) garante

$$\cos \gamma_x(\lambda) \geq \cos \gamma_x(L_{min}) \geq \cos \bar{\gamma}, \forall \lambda \in [L_{min}, L_{max}] \text{ e } \forall x \in \mathbb{R}^n.$$

Portanto,

$$\frac{-\nabla f(x)^T d(\lambda)}{\|\nabla f(x)\|_2 \|d(\lambda)\|_2} \geq \cos \bar{\gamma} \iff \nabla f(x)^T d(\lambda) \leq -\cos \bar{\gamma} \|\nabla f(x)\|_2 \|d(\lambda)\|_2.$$

Note que  $\cos(\bar{\gamma}) \neq 0$ , pela Hipótese (4.8). Assim, tomando  $\theta = \cos(\bar{\gamma})$ , a condição do ângulo é satisfeita.  $\square$

**Teorema 4.12.** *Se  $x_*$  é ponto limite de uma sequência gerada pelo Algoritmo 6 sob as Hipóteses (4.6), (4.7) e (4.8), então  $\nabla f(x_*) = 0$ .*

**Demonstração:** Pelo Lema 4.11, existem constantes  $\beta > 0$  e  $\theta \in (0, 1)$  satisfazendo as condições (2.8) e (2.9) na escolha da direção  $d_k$  do Algoritmo 1. O Teorema 2.30, garante que se  $x_*$  é ponto limite de uma sequência gerada por este algoritmo, então  $\nabla f(x_*) = 0$ .  $\square$

---

# O PROBLEMA DE OTIMIZAÇÃO DE MENOR VALOR ORDENADO

---

Este capítulo encontra-se dividido em três seções. Na Seção 5.1 abordamos a definição do problema LOVO, sua fundamentação teórica, assim como o subproblema LOVO de quadrados mínimos generalizado. Na sequência, apresentamos um algoritmo LOVO com convergência para pontos fracamente críticos. Por fim, na Seção 5.3, é descrito o Método e Gauss-Newton para problemas LOVO. Caso o leitor tenha interesse, indicamos a leitura dos trabalhos [2], [3] e [24], os quais apresentam grande parte da teoria deste capítulo.

## 5.1 O problema LOVO

O problema de Otimização de Menor Valor Ordenado (LOVO) foi proposto inicialmente por Andreani et al [3], em 2005. Logo no ano seguinte, Yano defendeu sua tese de doutorado no tema (ver [24]). Nesta seção, apresentamos o problema LOVO, as definições básicas, bem como alguns resultados clássicos e um algoritmo para o problema LOVO irrestrito. A referência desta seção é [3].

Dadas  $r$  funções  $F_i : \Omega \rightarrow \mathbb{R}$ ,  $i = 1, \dots, r$ , onde  $\Omega \subset \mathbb{R}^n$  e  $p \in \{1, 2, \dots, r\}$  um inteiro, definimos a função de Menor Valor Ordenado  $S_p : \Omega \rightarrow \mathbb{R}$  por

$$S_p(x) = \sum_{j=1}^p F_{i_j(x)}(x),$$

onde  $\{i_1(x), \dots, i_r(x)\} = \{1, \dots, r\}$  e  $F_{i_1(x)} \leq \dots \leq F_{i_p(x)} \leq \dots \leq F_{i_r(x)}$ .

Definimos o problema de Otimização de Menor Valor Ordenado (LOVO) como

$$\min_{x \in \Omega} S_p(x). \quad (5.1)$$

Se as funções  $F_i$ ,  $i = 1, \dots, r$ , são contínuas, então a função  $S_p$  é contínua. Contudo, mesmo que estas funções sejam diferenciáveis, normalmente a função  $S_p$  é não-suave.

O problema LOVO pode ainda ser definido de outra maneira. Considere o conjunto  $\{1, 2, \dots, r\}$ . Tal conjunto contém exatamente  $m = \binom{r}{p} = \frac{r!}{p!(r-p)!}$  subconjuntos distintos  $C_1, C_2, \dots, C_m$  com cardinalidade  $p$ . Definimos então a família de funções

$$f_i(x) = \sum_{j \in C_i} F_j(x), \forall i = 1, \dots, m, \forall x \in \Omega$$

e a função

$$f_{min}(x) = \min\{f_1(x), f_2(x), \dots, f_m(x)\}.$$

Note que  $f_{min}(x) = S_p(x)$  para todo  $x \in \Omega$ . Assim, o problema LOVO consiste em

$$\min_{x \in \Omega} f_{min}(x). \quad (5.2)$$

A Figura 5.1 apresenta um gráfico da função  $f_{min}$  para o caso em que a família de funções  $f_i : \mathbb{R} \rightarrow \mathbb{R}$ ,  $i = 1, \dots, m$ , possui  $m = 3$  elementos. Note que os pontos de interseção entre os traços das funções  $f_1$  e  $f_2$ , assim como  $f_2$  e  $f_3$ , são pontos nos quais  $f_{min}$  não é diferenciável.

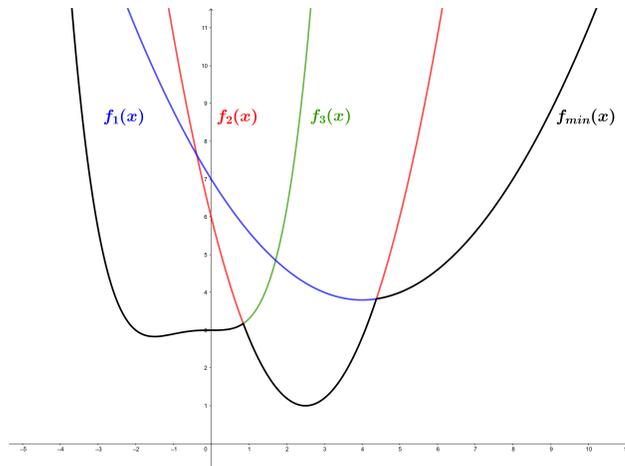


Figura 5.1: Gráfico da função  $f_{min}$  para  $m = 3$  funções reais arbitrárias.

### 5.1.1 Uma generalização do problema de quadrados mínimos não linear

Sejam  $y_1, y_2, \dots, y_r \in \mathbb{R}$  valores obtidos através da observação de determinado fenômeno e  $y = \Phi(x, t)$  uma função de ajuste. Denote por  $F_i$  o erro quadrático obtido pela função de ajuste na  $i$ -ésima observação, isto é,  $F_i = (\Phi(x, t_i) - y_i)^2$ . Portanto, a estimativa do parâmetro  $x$  através do método de quadrados mínimos não-linear, vem da solução do problema

$$\min_{x \in \Omega} \sum_{i=1}^r F_i(x). \quad (5.3)$$

Considere agora que  $r - p$  valores provêm de erros de observação e que desejamos eliminar sua influência na solução do problema (5.3) e, portanto, do ajuste. Os parâmetros do modelo através da solução do problema LOVO é traduzido por

$$\min_{x \in \Omega} S_p(x)$$

ou, equivalentemente,

$$\min_{x \in \Omega} f_{min}(x).$$

Desta forma, o subproblema LOVO apresentado anteriormente é uma generalização do método de quadrados mínimos não-linear, o qual possui a particularidade de ser capaz de eliminar a influência de “*outliers*”, isto é, de pontos que discrepam dos demais.

## 5.2 Algoritmo LOVO com convergência para pontos fracamente críticos

Antes de definirmos o algoritmo LOVO, precisamos fixar a notação de alguns termos.

**Definição 5.1.** [3, p.4] Para cada  $x \in \Omega$ , o conjunto  $I_{min}(x)$  é o conjunto de índices das funções  $f_i$  que minimizam a função  $f_{min}$  no ponto  $x$ , isto é,

$$I_{min}(x) = \{i \in 1, \dots, m \mid f_i(x) = f_{min}(x)\}.$$

**Lema 5.2.** *Sejam  $A \subset \Omega$  e  $x_* \in A$ . Se  $x_*$  é um minimizador global de  $f_{min}(x)$  sujeito a  $x \in A$ , então  $x_*$  é minimizador de  $f_i(x)$  sujeito a  $x \in A$ , para todo  $i \in I_{min}(x_*)$ . Em particular, tomando  $A = \Omega$ , se  $x_*$  é minimizador global de (5.2) então  $x_*$  é um minimizador de  $f_i(x)$  para todo  $i \in I_{min}(x_*)$ .*

**Demonstração:** Ver [3, Lema 2.1]. □

**Teorema 5.3.** *Se  $x_* \in \Omega$  é minimizador local de (5.2) então, para cada  $i \in I_{min}(x_*)$ ,  $x_*$  é minimizador local de  $f_i(x)$  sujeito a  $x \in \Omega$ .*

**Demonstração:** Ver [3, Teorema 2.1]. □

**Definição 5.4.** [3, p.6] Seja  $x_* \in \Omega$ . Dizemos que:

- a)  $x_*$  é *fortemente crítico* se, para todo  $i \in I_{min}(x_*)$ ,  $x_*$  é ponto estacionário de  $f_i$ .
- b)  $x_*$  é *fracamente crítico* se existe  $i \in I_{min}(x_*)$  tal que  $x_*$  é ponto estacionário de  $f_i$ .

Como visto anteriormente, o problema LOVO é um problema de otimização não-suave. Nos casos (5.1) e (5.2), uma alternativa viável para sua solução é a adaptação de métodos essencialmente suaves. Assim, a cada vez que  $\nabla f_{min}(x)$  é requerida, podemos substituí-la por  $\nabla f_i(x)$ , com  $i \in I_{min}(x)$ . De modo análogo, podemos proceder desta maneira caso o método exija informações acerca da hessiana de  $f_{min}$ . Tal como afirmam Andreani et al [3, p.06], tal estratégia pode ser catastrófica se aplicada a problemas não-suaves, entretanto, neste caso específico, as consequências são menos graves e, necessariamente, temos convergência para pontos fracamente críticos.

No algoritmo definido a seguir, adaptado de [3, Algoritmo U1], assumiremos que as funções  $f_i$  são continuamente diferenciáveis em todo  $x \in \mathbb{R}^n$ .

**Teorema 5.5.** *O Algoritmo 7 está bem definido e termina em  $x_k$  somente se  $x_k$  é fracamente crítico.*

**Demonstração:** Ver [3, Teorema 3.1]. □

**Teorema 5.6.** *Se  $x_*$  é ponto limite de uma sequência gerada pelo Algoritmo 7, então  $x_*$  é fracamente crítico. Além disso, se  $\lim_{k \in K} x_k = x_*$  e o mesmo  $i = v(k) \in I_{min}(x_k)$  é escolhido para infinitos índices  $k \in K$ , então  $i \in I_{min}(x_*)$  e  $\nabla f_i(x_*) = 0$ . Finalmente,  $\lim_{k \in K} \|\nabla f_{v(k)}(x_k)\|_2 = 0$ .*

**Demonstração:** Ver [3, Teorema 3.2]. □

---

**Algoritmo 7:** LOVO IRRESTRITO COM CONVERGÊNCIA PARA PONTOS FRACAMENTE CRÍTICOS

---

**Entrada:**  $\nabla f(x)$ ,  $\theta \in (0, 1)$ ,  $\alpha \in (0, 1)$ ,  $M > 1$ ,  $\beta > 0$ ,  $t_{one} > 0$  e uma aproximação inicial  $x_0 \in \mathbb{R}^n$

**Saída:**  $x_* \in \mathbb{R}^n$ , ponto fracamente crítico de  $f_{min}(x)$ .

1 **início**

2  $k \leftarrow 0;$

3 **repita**

4  $\quad$  Escolha  $v(k) \in I_{min}(x_k);$

5  $\quad$  Obtenha  $d_k$  tal que:

$$\nabla f_{v_k}(x_k)^T d_k \leq -\theta \|d_k\|_2 \|\nabla f_{v(k)}(x_k)\|_2 \quad (5.4)$$

$$\|d_k\|_2 \geq \beta \|\nabla f_{v(k)}(x_k)\|_2 \quad (5.5)$$

6  $\quad$  Calcule  $t_k > 0$ ,  $x_{k+1} \in \mathbb{R}^n$ , tal que:

$$f_{min}(x_{k+1}) \leq f_{min}(x_k) + \alpha t_k \nabla f_{v(k)}(x_k)^T d_k$$

$$t_k \geq t_{one} \text{ ou } f_{min}(x_k + \bar{t}_k d_k) > f_{min}(x_k) + \alpha \bar{t}_k \nabla f_{v(k)}(x_k)^T d_k, \text{ com } \bar{t}_k \leq M t_k$$

7  $\quad$  Faça:

$$k \leftarrow k + 1;$$

8 **até**  $\|\nabla f_{v_k}(x_k)\|_2 = 0;$

9 **fim**

10 **retorna**  $x_k$

---

### 5.3 O Método de Gauss-Newton para problemas LOVO

O algoritmo descrito nesta seção foi proposto por Andreani et al [2, Algoritmo 1], para problemas do tipo OVO (Otimização de Valor Ordenado), do qual o problema LOVO é um caso particular. Aqui, iremos enfatizar o uso deste algoritmo para problemas LOVO.

Os pontos limites das sequências geradas por este algoritmo são pontos estacionários do problema LOVO. Isto é, são pontos fracamente críticos de  $f_{min}(x)$  [2, p.4]. Além disso, assim como os demais algoritmos para problemas OVO que usam iterações variantes de Newton, a convergência do algoritmo geralmente é superlinear [2, p.3].

**Observação 5.7.** [19, p.619] Seja  $(x_k)_{k \in \mathbb{N}}$  uma sequência em  $\mathbb{R}^n$  que converge para  $x_*$ . Dizemos que a ordem de convergência é *superlinear* se

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x_*\|_2}{\|x_k - x_*\|_2} = 0$$

para todo  $k$  suficientemente grande.

Antes de descrevermos o algoritmo, faz-se necessário adotar algumas convenções. Dadas  $r$  funções  $F_i : \Omega \rightarrow \mathbb{R}$ ,  $i = 1, \dots, r$ , de classe  $C^1(\Omega)$ , onde  $\Omega \subset \mathbb{R}^n$ , seja  $p \in \{1, 2, \dots, r\}$  um número inteiro. Assim, como vimos em (5.2), o problema LOVO consiste em

$$\min_{x \in \Omega} f_{min}(x),$$

com

$$f_{min}(x) = \min \left\{ \sum_{j \in C_i} F_j(x), i = 1, \dots, m \right\},$$

onde  $m = \binom{r}{p}$  e cada  $C_i$  é um subconjunto distinto de cardinalidade  $p$  do conjunto  $\{1, \dots, r\}$ .

Fixado  $x_k \in \Omega$  e dado  $v_k \in I_{min}(x_k)$ , lembremos que

$$R_{v_k}(x_k) = \begin{bmatrix} F_{j_1}(x_k) \\ F_{j_2}(x_k) \\ \vdots \\ F_{j_r}(x_k) \end{bmatrix} \text{ e } J_{v_k}(x_k) = \begin{bmatrix} \frac{\partial F_{j_1}(x_k)}{\partial x_1} & \cdots & \frac{\partial F_{j_1}(x_k)}{\partial x_n} \\ \frac{\partial F_{j_2}(x_k)}{\partial x_1} & \cdots & \frac{\partial F_{j_2}(x_k)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_{j_r}(x_k)}{\partial x_1} & \cdots & \frac{\partial F_{j_r}(x_k)}{\partial x_n} \end{bmatrix},$$

onde  $j_1, j_2, \dots, j_r \in C_{v_k}$ . Note ainda que  $\nabla f_{v_k}(x_k) = J_{v_k}(x_k)^T R_{v_k}(x_k)$ , e assim, a direção de Gauss-Newton  $d_k^{GN}$  do problema de minimizar  $f_{v_k}(x)$ , com  $x \in \Omega$ , é dada por

$$J_{v_k}(x_k)^T J_{v_k}(x_k) d_k = -J_{v_k}(x_k)^T R_{v_k}(x_k).$$

Finalmente, estamos em condições de descrever o algoritmo do Método de Gauss-Newton para problemas LOVO. Como dito anteriormente, tal algoritmo foi proposto em [2, Algoritmo 1].

---

**Algoritmo 8: LOVO GAUSS-NEWTON**


---

**Entrada:**  $R(x)$ ,  $J(x)$ ,  $\alpha \in (0, 1)$ ,  $\beta > 0$ ,  $\varepsilon > 0$  e uma aproximação inicial  $x_0 \in \mathbb{R}^n$

**Saída:**  $x_* \in \mathbb{R}^n$ , ponto fracamente crítico de  $f_{min}(x)$ .

**1 início**

**2**  $k \leftarrow 0$ ;

**3 repita**

**4** Escolha  $v(k) \in I_{min}(x_k)$ ;

**5** Caso a matriz  $J_{v_k}(x_k)^T J_{v_k}(x_k)$  seja definida positiva, obtenha  $d_k$  solução de:

$$J_{v_k}(x_k)^T J_{v_k}(x_k) d_k = -J_{v_k}(x_k)^T R_{v_k}(x_k)$$

Caso contrário, obtenha  $d_k$  resolvendo:

$$(J_{v_k}(x_k)^T J_{v_k}(x_k) + \beta I) d_k = -J_{v_k}(x_k)^T R_{v_k}(x_k)$$

Faça  $t = 1$  ;

**6 enquanto**  $f_{min}(x_k + t d_k) > f_{min}(x_k) + \alpha t (J(x_k)^T R(x_k))^T d_k$  **faça**

**7** Escolha novo  $t \in [0.1t, 0.9t]$ .

**8 fim**

**9** Faça:

$$k \leftarrow k + 1;$$

**10 até**  $\|J_{v_k}(x_k)^T R_{v_k}(x_k)\|_2 \leq \varepsilon$ ;

**11 fim**

**12 retorna**  $x_k$

---

# O MÉTODO DE LEVENBERG-MARQUARDT PARA PROBLEMAS LOVO

---

O conteúdo deste capítulo constitui a maior contribuição deste trabalho. Nele, apresentamos uma variação original do Método de Levenberg-Marquardt para problemas LOVO, assim como diversos resultados que garantem a convergência do método. Mais especificamente, este capítulo aborda a adaptação da teoria do Método de Levenberg-Marquardt, apresentada no Capítulo 4, ao contexto dos problemas LOVO, discutida no Capítulo 5. Posteriormente, é apresentado um algoritmo de Levenberg-Marquardt com Busca Linear para problemas LOVO irrestritos e com convergência para pontos fracamente críticos, baseado no Algoritmo 6, proposto no Capítulo 4.

## 6.1 O Método de Levenberg-Marquardt para problemas LOVO

Considere  $r$  funções  $F_1, F_2, \dots, F_r$ , continuamente diferenciáveis em  $\Omega \subset \mathbb{R}^n$ , isto é,  $F_i \in C^1(\Omega)$ ,  $i = 1, 2, \dots, r$ . Seja  $p \in \{1, 2, \dots, r\}$  um inteiro dado.

No intuito de facilitar a compreensão, reintroduziremos alguns conceitos, agora com notação simplificada. Para tal, lembremos as funções vetoriais  $R(x)$  e  $J(x)$  definidas em  $\Omega \subset \mathbb{R}^n$

$$R(x) = \begin{bmatrix} F_1(x) \\ F_2(x) \\ \vdots \\ F_r(x) \end{bmatrix} \quad \text{e} \quad J(x) = \begin{bmatrix} \frac{\partial F_1(x)}{\partial x_1} & \cdots & \frac{\partial F_1(x)}{\partial x_n} \\ \frac{\partial F_2(x)}{\partial x_1} & \cdots & \frac{\partial F_2(x)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_r(x)}{\partial x_1} & \cdots & \frac{\partial F_r(x)}{\partial x_n} \end{bmatrix}.$$

Considere ainda o conjunto  $I_{min}(x) = \{i \in \{1, \dots, m\} | f_i(x) = f_{min}(x)\}$ . Assim, dado  $i \in I_{min}(x)$ , denotamos:

- O vetor coluna  $R^i(x) := R|_{C_i}(x)$ , isto é,  $[R^i(x)]_j = F_j(x)$  para todo  $j \in C_i$ .
- A função  $f^i(x) = \frac{1}{2} \|R^i(x)\|_2^2$ .
- A matriz  $J^i(x) := J|_{C_i}(x)$ , isto é,  $[J^i(x)]_{j,l} = \frac{\partial F_j(x)}{\partial x_l}$  para todo  $j \in C_i$  e  $l = 1, \dots, n$ .

Com as notações acima,  $\nabla f^i(x) = J^i(x)^T R^i(x)$ .

### 6.1.1 Algoritmo de Levenberg-Marquardt para problemas LOVO e alguns resultados

O algoritmo a seguir é uma contribuição nossa e corresponde a uma adaptação do método de busca linear com Levenberg-Marquardt, no contexto do problema LOVO irrestrito.

---

**Algoritmo 9:** LOVO LEVENBERG-MARQUARDT IRRESTRITO COM CONVERGÊNCIA PARA PONTOS FRACAMENTE CRÍTICOS

---

**Entrada:**  $R(x)$ ,  $J(x)$ , uma aproximação inicial  $x_0 \in \mathbb{R}^n$ , uma constante  $\alpha \in (0, 1)$  e uma tolerância  $\varepsilon > 0$ .

**Saída:**  $x_* \in \mathbb{R}^n$ , ponto fracamente crítico de  $f_{min}(x)$ .

1 início

2  $k \leftarrow 0$  ;

3 repita

4 Escolha  $i = v(k) \in I_{min}(x_k)$  ;

5 Calcule  $\lambda_k$ ;

6 Obtenha  $d^i(\lambda_k)$  solução de:

$$(J^i(x_k)^T J^i(x_k) + \lambda_k I) d^i(\lambda_k) = -J^i(x_k)^T R^i(x_k); \quad (6.1)$$

Faça  $t = 1$ ;

7 enquanto  $f^i(x_k + t d^i(\lambda_k)) > f^i(x_k) + \alpha t \nabla f^i(x_k)^T d^i(\lambda_k)$  faça

8 | Escolha novo  $t \in [0.1t, 0.9t]$ .

9 fim

10 Faça:

$$x_{k+1} \leftarrow x_k + t d^i(\lambda_k);$$

$$k \leftarrow k + 1;$$

11 até  $\|\nabla f^i(x_k)\|_2 \leq \varepsilon$ ;

12 fim

13 retorna  $x_k$

---

O teorema a seguir apresenta algumas particularidades importantes acerca das sequências geradas pelo Algoritmo 9. Note que, na notação atual, a função angular a qual que mede o ângulo entre os vetores  $d^i(\lambda)$  (direção de Levenberg-Marquardt) e  $-g^i$  (direção de máxima descida) é denotada por  $\gamma^i(\lambda)$ .

**Teorema 6.1.** *Seja  $x_*$  ponto limite de uma sequência gerada pelo Algoritmo 9. Suponha que  $\lim_{k \in K} x_k = x_*$  e um mesmo índice  $i = v(k) \in I_{min}(x_k)$  é escolhido para infinitos índices  $k \in K$ , então:*

1.  $i \in I_{min}(x_*)$ ;
2.  $\|d^i(\lambda)\|_2^2$  é uma função contínua decrescente na variável  $\lambda$ , tal que  $\lim_{\lambda \rightarrow \infty} \|d^i(\lambda)\|_2^2 = 0$ ;
3. A função angular  $\gamma^i(\lambda)$  é contínua monótona decrescente na variável  $\lambda$  e  $\lim_{\lambda \rightarrow \infty} \gamma^i(\lambda) = 0$ . Além disso,  $d^i(\lambda)$  rotaciona na direção de  $-g^i$ , quando  $\lambda \rightarrow \infty$ .

**Demonstração:** Seja  $(x_k)_{k \in \mathbb{N}}$  uma sequência gerada pelo Algoritmo 9 cujo limite é  $x_*$ . Defina a sequência infinita  $K = \{k_0, k_1, k_2, \dots\}$  de índices tal que existe  $i \in \{1, 2, \dots, m\}$  com  $i = v(k)$  para todo  $k \in K$  e  $\lim_{k \in K} x_k = x_*$ . Observe que o índice  $i$  e a sequência infinita  $K$  existem uma vez que o conjunto de índices  $\{1, 2, \dots, m\}$  é finito.

Vamos provar o item 1.

Considere a função  $f^i(x) = \frac{1}{2} \|R^i(x)\|_2^2 = \frac{1}{2} \sum_{j \in C_i} F_j^2(x)$ . Por hipótese, como as funções  $F_j$  são continuamente diferenciáveis, claramente  $f^i \in C^1$ . A continuidade de  $f^i$  garante que

$$\lim_{k \in K} f^i(x_k) = f^i(x_*).$$

Como  $i \in I_{min}(x_k)$ , pela definição deste conjunto, para todo  $k \in K$ ,

$$f^i(x_*) \leq f^l(x_k), \forall l \in \{1, 2, \dots, m\}.$$

Aplicando os limites em ambos os lados da desigualdade anterior, concluímos que  $f^i(x_*) \leq f^l(x_*)$  para todo  $l \in \{1, 2, \dots, m\}$  e todo  $k \in K$ . Portanto,  $i \in I_{min}(x_*)$ .

A seguir, provamos o item 2.

Inicialmente, note que para todo elemento da subsequência  $(x_k)_{k \in K}$ , a direção  $d^i(\lambda)$  é dada por

$$d^i(\lambda) = -[J^i(x_k)^T J^i(x_k) + \lambda I]^{-1} J^i(x_k)^T R^i(x_k), \text{ com } i \text{ fixo.} \quad (6.2)$$

Dado que a matriz  $J^i(x_k)^T J^i(x_k) + \lambda I$  é definida positiva, sua inversa existe e assim  $d^i(\lambda)$  é uma função contínua sobre a variável  $\lambda$ . Como a função norma  $\|\cdot\|_2$  é contínua, segue que  $\|d^i(\lambda)\|_2^2$  é função contínua sobre  $\lambda$ .

Uma vez que a matriz  $J^i(x_k)^T J^i(x_k)$  é simétrica e normal, existe uma matriz ortogonal  $Q$  tal que  $Q^T J^i(x_k)^T J^i(x_k) Q = D$ , onde  $D$  é uma matriz diagonal com entradas diagonais positivas. Portanto,

$$J^i(x_k)^T J^i(x_k) = [Q^T]^{-1} D Q^{-1} = Q D Q^T. \quad (6.3)$$

Assim, de (6.2) e (6.3),

$$\begin{aligned} d^i(\lambda) &= -[J^i(x_k)^T J^i(x_k) + \lambda I]^{-1} J^i(x_k)^T R^i(x_k), \\ &= -[Q D Q^T + \lambda I]^{-1} J^i(x_k)^T R^i(x_k), \\ &= -[Q[D + \lambda I]Q^T]^{-1} J^i(x_k)^T R^i(x_k), \\ &= -[Q^T]^{-1}[D + \lambda I]Q^{-1} J^i(x_k)^T R^i(x_k), \\ &= -Q[D + \lambda I]^{-1}Q^T J^i(x_k)^T R^i(x_k). \end{aligned}$$

Definindo  $v = Q^T J^i(x_k)^T R^i(x_k)$ , temos

$$d^i(\lambda) = -Q[D + \lambda I]^{-1}v.$$

Escrevendo  $D_j = [D]_{jj}$ ,

$$\begin{aligned}
\|d^i(\lambda)\|_2^2 &= d^i(\lambda)^T d^i(\lambda), \\
&= [-Q[D + \lambda I]^{-1}v]^T [-Q[D + \lambda I]^{-1}v], \\
&= [v^T [D + \lambda I]^{-1} Q^T] [Q[D + \lambda I]^{-1}v], \\
&= v^T [D + \lambda I]^{-1} Q^T Q [D + \lambda I]^{-1}v, \\
&= v^T [[D + \lambda I]^2]^{-1}v, \\
&= \sum_{j \in C_i} \frac{v_j^2}{(D_j + \lambda)^2}.
\end{aligned}$$

Finalmente,

$$\|d^i(\lambda)\|_2^2 = \sum_{j \in C_i} \frac{v_j^2}{(D_j + \lambda)^2}. \quad (6.4)$$

Tomando  $\lambda \rightarrow \infty$  na equação (6.4), claramente  $\lim_{\lambda \rightarrow \infty} \|d^i(\lambda)\|_2^2 = 0$ . Desse modo,  $\|d^i(\lambda)\|_2^2$  é uma função contínua decrescente.

Por fim, provamos o item 3.

Sabemos que  $f^i(x) = \frac{1}{2} \sum_{j \in C_i} F_j^2(x)$ . Logo, a direção de máxima descida  $-g^i$  do problema  $\min_{x \in \mathbb{R}^n} f_i(x)$  é dada por

$$\begin{aligned}
-g^i &= -\nabla f^i(x) = \sum_{j \in C_i} F_j(x) \frac{\partial F_j(x)}{\partial x_l}, \quad l = 1, \dots, n, \\
&= \begin{bmatrix} \sum_{j \in C_i} F_j(x) \frac{\partial F_j(x)}{\partial x_1} \\ \vdots \\ \sum_{j \in C_i} F_j(x) \frac{\partial F_j(x)}{\partial x_n} \end{bmatrix}, \\
&= -J^i(x)^T R^i(x).
\end{aligned}$$

Precisamos mostrar que a função angular  $\gamma^i(\lambda)$  é contínua monótona decrescente na variável

$\lambda$ . De fato, escrevendo  $\kappa = J^i(x)^T R^i(x)$  e  $\omega = \sum_{j \in C_i} \frac{v_j^2}{(D_j + \lambda)^2}$ ,

$$\begin{aligned} \cos \gamma^i(\lambda) &= \frac{d^i(\lambda)^T (-g^i)}{\|d^i(\lambda)\|_2 \| -g^i \|_2}, \\ &= \frac{[-Q[D + \lambda I]^{-1}v]^T [-J^i(x_k)^T R^i(x_k)]}{\omega^{\frac{1}{2}} \|\kappa\|_2}, \\ &= \frac{v^T [D + \lambda I]^{-1} Q^T J^i(x_k)^T R^i(x_k)}{\omega^{\frac{1}{2}} \|\kappa\|_2}, \\ &= \frac{v^T [D + \lambda I]^{-1} v}{\omega^{\frac{1}{2}} \|\kappa\|_2}, \\ &= \frac{1}{\|\kappa\|_2} \frac{\sum_{j \in C_i} \frac{v_j^2}{D_j + \lambda}}{\omega^{\frac{1}{2}}}. \end{aligned}$$

Derivando a expressão acima em relação a  $\lambda$ , obtemos

$$\frac{\partial}{\partial \lambda} \cos \gamma^i(\lambda) = \frac{1}{\|\kappa\|_2} \frac{-\omega^{\frac{3}{2}} + \omega^{-\frac{1}{2}} \left( \sum_{j \in C_i} \frac{v_j^2}{D_j + \lambda} \right) \left( \sum_{j \in C_i} \frac{v_j^2}{(D_j + \lambda)^3} \right)}{\omega}.$$

Calculando o MMC dos denominadores das frações que compõem o numerador,

$$\frac{\partial}{\partial \lambda} \cos \gamma^i(\lambda) = \frac{1}{\|\kappa\|_2} \frac{-\omega^{\frac{3}{2}} \omega^{\frac{1}{2}} + \left( \sum_{j \in C_i} \frac{v_j^2}{D_j + \lambda} \right) \left( \sum_{j \in C_i} \frac{v_j^2}{(D_j + \lambda)^3} \right)}{\omega^{\frac{1}{2}} \omega}.$$

Realizando as simplificações necessárias, obtemos finalmente

$$\frac{\partial}{\partial \lambda} \cos \gamma^i(\lambda) = \frac{1}{\|\kappa\|_2} \frac{\left( \sum_{j \in C_i} \frac{v_j^2}{D_j + \lambda} \right) \left( \sum_{j \in C_i} \frac{v_j^2}{(D_j + \lambda)^3} \right) - \omega^2}{\omega^{\frac{3}{2}}}. \quad (6.5)$$

Na equação (6.5), tome o produto  $\left(\sum_{j \in C_i} \frac{v_j^2}{D_j + \lambda}\right) \left(\sum_{j \in C_i} \frac{v_j^2}{(D_j + \lambda)^3}\right)$ . Assim,

$$\begin{aligned} \left(\sum_{j \in C_i} \frac{v_j^2}{D_j + \lambda}\right) \left(\sum_{j \in C_i} \frac{v_j^2}{(D_j + \lambda)^3}\right) &= \frac{v_1^4}{(D_1 + \lambda)^4} + \frac{v_1^2 v_2^2}{(D_1 + \lambda)(D_2 + \lambda)^3} + \cdots + \\ &+ \frac{v_1^2 v_p^2}{(D_1 + \lambda)(D_p + \lambda)^3} + \frac{v_2^2 v_1^2}{(D_2 + \lambda)(D_1 + \lambda)^3} + \frac{v_2^4}{(D_2 + \lambda)^4} + \\ &+ \cdots + \frac{v_2^2 v_p^2}{(D_2 + \lambda)(D_p + \lambda)^3} + \cdots + \frac{v_p^2 v_1^2}{(D_p + \lambda)(D_1 + \lambda)^3} + \\ &+ \frac{v_p^2 v_2^2}{(D_p + \lambda)(D_2 + \lambda)^3} + \cdots + \frac{v_p^4}{(D_p + \lambda)^4}. \end{aligned} \quad (6.6)$$

Calculando o MMC dos denominadores na expressão (6.6),

$$\begin{aligned} \left(\sum_{j \in C_i} \frac{v_j^2}{D_j + \lambda}\right) \left(\sum_{j \in C_i} \frac{v_j^2}{(D_j + \lambda)^3}\right) &= \frac{v_1^4 (D_2 + \lambda)^4 \cdots (D_p + \lambda)^4}{(D_1 + \lambda)^4 (D_2 + \lambda)^4 \cdots (D_p + \lambda)^4} + \\ &+ \frac{v_1^2 v_2^2 (D_1 + \lambda)^3 (D_2 + \lambda) \cdots (D_p + \lambda)^4}{(D_1 + \lambda)^4 (D_2 + \lambda)^4 \cdots (D_p + \lambda)^4} + \cdots + \\ &+ \frac{v_p^4 (D_1 + \lambda)^4 \cdots (D_{p-1} + \lambda)^4}{(D_1 + \lambda)^4 (D_2 + \lambda)^4 \cdots (D_p + \lambda)^4}, \\ &= \frac{\left(\sum_{j \in C_i} v_j^2 \prod_{j' \in C_i, j' \neq j} (D_{j'} + \lambda)\right) \left(\sum_{j \in C_i} v_j^2 \prod_{j' \in C_i, j' \neq j} (D_{j'} + \lambda)^3\right)}{\prod_{j \in C_i} (D_j + \lambda)^4}. \end{aligned} \quad (6.7)$$

De modo análogo, obtemos a expressão

$$\omega^2 = \left(\sum_{j \in C_i} \frac{v_j^2}{(D_j + \lambda)^2}\right)^2 = \frac{\left(\sum_{j \in C_i} v_j^2 \prod_{j' \in C_i, j' \neq j} (D_{j'} + \lambda)^2\right)^2}{\prod_{j \in C_i} (D_j + \lambda)^4}. \quad (6.8)$$

Portanto, de (6.5), (6.7) e (6.8), obtemos

$$\begin{aligned} \frac{\partial}{\partial \lambda} \cos \gamma^i(\lambda) = & \frac{1}{\|\kappa\|_2} \frac{\left( \sum_{j \in C_i} v_j^2 \prod_{j' \in C_i, j' \neq j} (D_{j'} + \lambda) \right) \left( \sum_{j \in C_i} v_j^2 \prod_{j' \in C_i, j' \neq j} (D_{j'} + \lambda)^3 \right)}{\omega^{\frac{3}{2}} \prod_{j \in C_i} (D_j + \lambda)^4} + \\ & - \frac{\left( \sum_{j \in C_i} v_j^2 \prod_{j' \in C_i, j' \neq j} (D_{j'} + \lambda)^2 \right)^2}{\omega^{\frac{3}{2}} \prod_{j \in C_i} (D_j + \lambda)^4}. \end{aligned} \quad (6.9)$$

Note que  $\left( \prod_{j' \in C_i, j' \neq j} (D_{j'} + \lambda) \right) \left( \prod_{j' \in C_i, j' \neq j} (D_{j'} + \lambda)^3 \right) = \left( \prod_{j' \in C_i, j' \neq j} (D_{j'} + \lambda)^2 \right)^2$ . Deste modo, podemos reescrever o numerador da expressão (6.9) como sendo

$$\begin{aligned} & \left( \sum_{j \in C_i} \left( v_j \left( \prod_{j' \in C_i, j' \neq j} (D_{j'} + \lambda) \right)^{\frac{1}{2}} \right)^2 \right) \left( \sum_{j \in C_i} \left( v_j \left( \prod_{j' \in C_i, j' \neq j} (D_{j'} + \lambda)^3 \right)^{\frac{1}{2}} \right)^2 \right) \\ & - \left( \sum_{j \in C_i} \left( v_j \left( \prod_{j' \in C_i, j' \neq j} (D_{j'} + \lambda) \right)^{\frac{1}{2}} \right) \left( v_j \left( \prod_{j' \in C_i, j' \neq j} (D_{j'} + \lambda)^3 \right)^{\frac{1}{2}} \right) \right)^2. \end{aligned} \quad (6.10)$$

A Desigualdade de Cauchy-Bunyakovskii-Schwarz afirma que (6.10) é positivo. Como o denominador de (6.9) e a constante  $c$  também são positivas, podemos concluir que  $\frac{\partial}{\partial \lambda} \cos \gamma^i(\lambda)$  assume apenas valores positivos. Consequentemente, a função  $\gamma^i(\lambda)$  é monótona decrescente.

Agora, vamos analisar o comportamento de  $\gamma^i(\lambda)$  quando  $\lambda \rightarrow \infty$ . Considere a matriz  $J^i(x_k)^T J^i(x_k) + \lambda I$ . Para valores muito grandes de  $\lambda$ , esta matriz é dominada pela diagonal  $\lambda I$ . Então, pela expressão (6.1), quando  $\lambda \rightarrow \infty$  temos que  $d^i(\lambda) \rightarrow \frac{-J^i(x_k)^T R^i(x_k)}{\lambda} = \frac{-g^i}{\lambda}$ . Portanto  $d^i(\lambda)$  e  $-g^i$  tornam-se proporcionais no limite e  $\gamma^i(\lambda)$  tende a zero, assim como conclui Marquardt no caso análogo para o Método de Levenberg-Marquardt clássico [15, Teorema 3]. No entanto, se tomarmos  $\lambda = 0$  na equação (6.1), obtemos a direção de Gauss-Newton e o ângulo formado entre  $d^i(\lambda)$  e  $-g^i$  pertence ao intervalo aberto  $(0, \pi/2)$ . Segue que  $\gamma^i(\lambda)$  é função contínua monótona decrescente na variável  $\lambda$  tal que  $\lim_{\lambda \rightarrow \infty} \gamma^i(\lambda) = 0$ .  $\square$

As hipóteses listadas a seguir são importantes para assegurar a convergência do Algoritmo 9 para pontos fracamente críticos. Mais especificamente, tais hipóteses nos permite garantir a existência de constantes  $\beta > 0$  e  $\theta \in (0, 1)$  que satisfaçam as chamadas Condição  $\beta$  ( $\|d_k\|_2 \geq \beta \|\nabla f(x_k)\|_2$ ) e Condição do Ângulo ( $\nabla f(x_k)^T d_k \leq -\theta \|\nabla f(x_k)\|_2 \|d_k\|_2$ ), na determinação da direção  $d_k$ . A Condição  $\beta$  visa determinar um tamanho mínimo para a direção  $d_k$  de modo a auxiliar a Condição de Armijo na busca do passo, impedindo que os passos se aproximem muito rapidamente de zero [10, p.25]. Já a Condição do Ângulo nos é útil, pois define um cone agudo com eixo na semirreta gerada por  $-\nabla f(x_k)$ , formado por direções de descida admissíveis e tais que os cossenos do ângulos formado entre estas direções e  $\nabla f(x_k)$  estejam uniformemente separadas de 0, impedindo assim a convergência para pontos não estacionários [16, p.103].

Deste modo, considere as seguintes hipóteses:

**Hipótese 6.2.** *A escolha de  $\lambda$  é limitada inferiormente e superiormente por constantes não-negativas, isto é, existem constantes  $L_{min}, L_{max} \in \mathbb{R}_+$ , com  $L_{min} \leq L_{max}$  tais que  $\lambda \in [L_{min}, L_{max}]$ .*

**Hipótese 6.3.** *Suponha que  $x_*$  é ponto limite de uma sequência  $(x_k)_{k \in \mathbb{N}}$  gerada pelo Algoritmo 9 tal que um mesmo índice  $i = v(k) \in I_{min}(x_k)$  é escolhido para infinitos índices  $k \in K$  e  $\lim_{k \in K} x_k = x_*$ . Considere ainda que:*

- i) Existe  $M \in \mathbb{R}$  tal que  $\sigma_1^i(x) < M$  para todo  $x \in \mathbb{R}^n$ ; onde  $\sigma_1^i(x)$  é o maior autovalor da matriz  $J^i(x)^T J^i(x)$ .*
- ii) O ângulo  $\gamma_x^i(\lambda)$  entre a direção de Levenberg-Marquardt e a direção de máxima descida é limitado superiormente por um constante positiva  $\bar{\gamma}^i$  tal que  $0 \leq \gamma_x^i(\lambda) \leq \bar{\gamma}^i < \frac{\pi}{2}$  para todo  $\lambda \in [L_{min}, L_{max}]$  e para todo  $x \in \mathbb{R}^n$ .*

**Lema 6.4.** *Suponha que o Algoritmo (9) é implementado de forma a satisfazer as Hipóteses (6.2) e (6.3). Então existem constantes  $\beta > 0$  e  $\theta \in (0, 1)$  tais que:*

$$\|d^i(\lambda_k)\|_2 \geq \beta \|\nabla f^i(x_k)\|_2,$$

$$\nabla f^i(x_k)^T d^i(\lambda_k) \leq -\theta \|\nabla f^i(x_k)\|_2 \|d^i(\lambda_k)\|_2.$$

**Demonstração:** De fato, combinando  $\nabla f^i(x) = J^i(x)^T R^i(x)$  e (6.1),

$$(J^i(x)^T J^i(x) + \lambda I) d^i(\lambda) = -\nabla f^i(x),$$

$$\|J^i(x)^T J^i(x) + \lambda I\|_2 \|d^i(\lambda)\|_2 \geq \|\nabla f^i(x)\|_2,$$

$$\begin{aligned} \|d^i(\lambda)\|_2 &\geq \frac{1}{\|J^i(x)^T J^i(x) + \lambda I\|_2} \|\nabla f^i(x)\|_2, \\ &\geq \frac{1}{\|J^i(x)^T J^i(x)\|_2 + \lambda} \|\nabla f^i(x)\|_2. \end{aligned}$$

Pela definição da *2-Norma*,

$$\|d^i(\lambda)\|_2 \geq \frac{1}{\sigma_1^i(x) + \lambda} \|\nabla f^i(x)\|_2.$$

Assim, aplicando as Hipóteses (6.2) e (6.3),

$$\begin{aligned} \|d^i(\lambda)\|_2 &\geq \frac{1}{M + \lambda} \|\nabla f^i(x)\|_2, \\ &\geq \frac{1}{M + L_{max}} \|\nabla f^i(x)\|_2. \end{aligned}$$

Portanto, basta tomar  $\beta = \frac{1}{M + L_{max}}$ .

Pela definição de  $\gamma_x^i(\lambda)$ , temos que

$$\cos \gamma_x^i(\lambda) = \frac{-\nabla f^i(x)^T d^i(\lambda)}{\|\nabla f^i(x)\|_2 \|d^i(\lambda)\|_2}.$$

O Teorema 6.1 e a Hipótese (6.2) implicam que  $\gamma_x^i(\lambda) \leq \gamma_x^i(L_{min})$ , para todo  $\lambda \in [L_{min}, L_{max}]$  e para todo  $x \in \mathbb{R}^n$ . Desse modo, como a função cosseno é contínua e decrescente no intervalo  $[0, \pi/2]$ , a Hipótese (6.3) garante que para todo  $\lambda \in [L_{min}, L_{max}]$  e para todo  $x \in \mathbb{R}^n$ ,

$$\cos \gamma_x^i(\lambda) \geq \cos \gamma_x^i(L_{min}) \geq \cos \bar{\gamma}^i.$$

Portanto,

$$\frac{-\nabla f^i(x)^T d^i(\lambda)}{\|\nabla f^i(x)\|_2 \|d^i(\lambda)\|_2} \geq \cos \bar{\gamma}^i \iff \nabla f^i(x)^T d^i(\lambda) \leq -\cos \bar{\gamma}^i \|\nabla f^i(x)\|_2 \|d^i(\lambda)\|_2.$$

Note que  $\cos(\bar{\gamma}^i) \neq 0$ , pela Hipótese (6.3). Assim, tomando  $\theta = \cos(\bar{\gamma}^i)$ , a condição do ângulo é satisfeita.  $\square$

**Teorema 6.5.** *Se  $x_*$  é ponto limite de uma sequência gerada pelo Algoritmo 9 sob as Hipóteses (6.2) e (6.3), então  $\nabla f^i(x_*) = 0$ .*

**Demonstração:** Pelo Lema 6.4, existem constantes  $\beta > 0$  e  $\theta \in (0, 1)$ , satisfazendo as chamadas condições  $\beta$  e do ângulo  $\theta$ , na escolha da direção  $d(\lambda_k)$ . Então, o Algoritmo 9 pode ser visto como um algoritmo de descida com busca linear, assim como o Algoritmo 1. Portanto, pelo Teorema 2.30, temos  $\nabla f^i(x_*) = 0$ .  $\square$

---

# EXPERIMENTOS NUMÉRICOS

---

Neste capítulo apresentamos alguns experimentos numéricos comparando o Método de Gauss-Newton, devido a Andreani et al [2, Algoritmo 1], e a adaptação do Método de Levenberg-Marquardt no contexto dos problemas LOVO, proposto no Capítulo 6. Para fins de comparação, os testes envolvendo este último método foram realizados com diversas escolhas para o parâmetro de *damping*.

Para a realização dos testes, consideramos uma família com 6 funções modelo, as quais encontram-se explicitadas na Tabela 7.1. Os testes foram executados em conjuntos contendo 100, 1000 e 5000 pontos, com uma taxa de confiança de 90%, isto é, apenas 10% de todos os pontos considerados consistiam de avaliações erradas dos modelos. Os conjuntos com dados de 100 e 1000 pontos foram obtidos através da avaliação da função modelo em um mesmo intervalo. Deste modo, podemos ver o conjunto maior como um refinamento das informações presentes no menor. Já para o caso dos conjuntos com 5000 pontos, foram considerados intervalos cinco vezes maiores que os originais. Durante a execução dos algoritmos, foram tomados como critérios de parada a norma do gradiente da função objetivo menor que uma tolerância ( $\varepsilon = 10^{-4}$ ), o número máximo de iterações ( $\mathcal{K}_{max} = 500$ ) e o tempo máximo de execução (1 hora).

Nas tabelas apresentadas na Seção 7.2, encontram-se os dados relativos a cada teste numérico. O Método de Gauss-Newton para problemas LOVO está indicado pela sigla GN, já o Método de Levenberg-Marquardt pela sigla LM. Os diferentes parâmetros de *damping* testados estão indicados pelas siglas PD1, PD2, ..., PD7, PD7H1, ..., PD8, PD8H1, ..., PD8H3, conforme indicado na Tabela 7.1.

Sigla	Autoria	Parâmetro de <i>damping</i>
PD1	Levenberg, ver [13]	$\lambda_k = \frac{\ J(x_k)^T R(x_k)\ _2^2}{f(x_k)}$
PD2	Levenberg modificado, ver [13], [5]	$\lambda_k = \ J(x_k)^T R(x_k)\ _2^2$
PD3	Levenberg modificado 2, ver [13], [5]	$\lambda_k = \ J(x_k)^T R(x_k)\ _2$
PD4	Yamashita & Fukushima, ver [23]	$\lambda_k = \ R(x_k)\ _2^2 = 2f(x)$
PD5	Fan & Yuan, ver [9]	$\lambda_k = \ R(x_k)\ _2 = \sqrt{2f(x)}$
PD6	Benatti, ver [5]	$\lambda_k = \frac{2\ J(x_k)^T R(x_k)\ _2}{3k}$
PD7	Schwertner 01	$\lambda_k = \frac{2\sqrt{f(x_k)}}{3k}$
PD7H1	Schwertner 01, híbrido 01	$\lambda_k = \max \left\{ \frac{2\sqrt{f(x_k)}}{3k}, 10^{-2} \right\}$
PD7H2	Schwertner 01, híbrido 02	$\lambda_k = \max \left\{ \frac{2\sqrt{f(x_k)}}{3k}, 1 \right\}$
PD7H3	Schwertner 01, híbrido 03	$\lambda_k = \max \left\{ \frac{2\sqrt{f(x_k)}}{3k}, 10 \right\}$
PD8	Schwertner 02	$\lambda_k = \frac{\ R(x_k)\ _2^2}{3k} = \frac{2f(x)}{3k}$
PD8H1	Schwertner 02, híbrido 01	$\lambda_k = \max \left\{ \frac{2f(x)}{3k}, 10^{-2} \right\}$
PD8H2	Schwertner 02, híbrido 02	$\lambda_k = \max \left\{ \frac{2f(x)}{3k}, 1 \right\}$
PD8H3	Schwertner 02, híbrido 03	$\lambda_k = \max \left\{ \frac{2f(x)}{3k}, 10 \right\}$

Tabela 7.1: Parâmetros de *damping* empregados nos testes numéricos.

Os algoritmos foram implementados na linguagem Python 3. Os *scripts* em Python 3, bem como os arquivos contendo os dados, estão disponibilizados no repositório aberto:

[https://github.com/aschwertner/LOVO\\_LM.git](https://github.com/aschwertner/LOVO_LM.git)

## 7.1 Funções testadas e resultados

A Tabela 7.2 apresenta todas as informações essenciais acerca das funções modelo adotadas para gerar os dados empregados nos testes.

Modelo	Expressão Matemática	Parâmetros adotados	Intervalo dos dados
Polinomial de Grau 1	$r(x) = ax + b$	$a = -3.2531, b = 15.2347$	$[-10.0, 10.0]$ $[-50.0, 50.0]$
Polinomial de Grau 3	$r(x) = ax^3 + bx^2 + cx + d$	$a = 1.12548, b = 2.53168$ $c = 3.14724, d = 0.589134$	$[-10.0, 10.0]$ $[-50.0, 50.0]$
Exponencial	$r(x) = ae^{bx+c} + d$	$a = 5.3127, b = -0.252$ $c = -0.755, d = -25.64$	$[-14.0, -6.0]$ $[-30.0, 10.0]$
Senoidal(Tipo 1)	$r(x) = a \sin(bx + c) + d$	$a = 40.5367, b = 2.345$ $c = -5.234, d = 24.12$	$[-0.5, 0.5]$ $[-2.5, 2.5]$
Senoidal(Tipo 2)	$r(x) = a \sin(bx) + c \cos(dx) + e$	$a = 13.535, b = -11.86$ $c = 12.8239, d = -6.30077$ $e = 4.82982$	$[-0.5, 0.5]$ $[-2.5, 2.5]$
Logístico	$r(x) = \frac{a}{1 + e^{bx+c}}$	$a = -10.5772, b = -4.52081$ $c = 19.6434$	$[2.0, 6.0]$ $[-6.0, 14.0]$

Tabela 7.2: Modelos utilizados para os testes.

Note que a coluna “Intervalo dos dados” apresenta dois intervalos para cada modelo, tais intervalos foram tomados para gerar os dados referentes aos testes com 100 e 1000 pontos, e com 5000 pontos, respectivamente.

### 7.1.1 Análise dos Resultados

A seguir, apresentamos um breve comentário sobre os resultados evidenciados pelos testes. Para facilitar a escrita e não tornar a leitura cansativa, serão utilizadas as abreviações “pts.” e “its.”, para as palavras “pontos” e “iterações”.

- Polinomial de Grau 1

Analisando-se os dados apresentados nas Tabelas 7.4, 7.5 e 7.6, podemos concluir que ambos os métodos obtiveram sucesso na determinação dos possíveis parâmetros para o

modelo, nos três cenários possíveis (100, 1000 e 5000 pts.). O Método de Gauss-Newton conseguiu resolver o problema LOVO no menor número de iterações (100 pts.: 3 its.; 1000 pts.: 3 its.; 5000 pts.: 5 its.). Destacaram-se ainda LM PD7 (100 pts.: 5 its.; 1000 pts.: 4 its.; 5000 pts.: 9 its.) e LM PD7H1 (100 pts.: 5 its.; 1000 pts.: 4 its.; 5000 pts.: 9 its.), sendo que último registrou o menor tempo de execução.

Contudo, convém destacar que LM PD2 não encontrou a solução em nenhum dos casos tratados, excedendo o número máximo de iterações, e dada a grandeza da norma do gradiente nas últimas iterações, é provável que os últimos iterandos não estivessem próximos de algum ponto fracamente crítico. De fato, ao analisar estes iterandos, nota-se que apesar de ser aceito passo 1 na Busca de Armijo, a direção de descida dada por PD2 não era suficiente para que houvesse um bom decréscimo no valor da norma do gradiente da função objetivo, de modo a convergir dentro do limite de iterações proposto. Tal comportamento repetiu-se em praticamente todos os testes realizados com este parâmetro de *damping*.

- Polinomial de Grau 3

Assim como pode ser visto nas Tabelas 7.7, 7.8 e 7.9, GN convergiu para a solução do problema no menor número de iterações (100 pts.: 3 its.; 1000 pts.: 3 its., 5000 pts.: 11 its.), sendo também o mais veloz.

Para o problema com 100 pontos, LM PD7 se destaca por realizar apenas 5 iterações, enquanto LM PD1 levou 69 iterações. Já para o segundo problema, LM PD7 e LM PD7H1 convergem em apenas 4 iterações, apenas uma a mais que GN. Por fim, LM PD8 e LM PD8H1, resolveram o problema LOVO com 5000 pontos em 15 iterações, em detrimento de LM PD6 que levou 141 iterações e também aos parâmetros de *damping* PD1, PD2 e PD3, que atingiram o número máximo de iterações.

- Exponencial

Para o conjunto de 100 pontos (Tabela 7.10), GN atingiu um ponto fracamente crítico com 12 iterações e os parâmetros de *damping* com melhor desempenho foram PD7 e PD7H1, os quais convergiram em 25 iterações. Como nos casos anteriores LM PD2 não

convergiu para uma solução dentro do número de iterações admissível. Os parâmetros PD4 e PD8 falharam durante a execução do algoritmo, uma vez que para as iterações 250 e 39, respectivamente, erros numéricos afetam a positividade da matriz  $J(x_k)^T J(x_k) + \lambda_k I$ , tornando-a singular.

Ao analisarmos o problema com 1000 pontos (Tabela 7.11), o desempenho numérico dos parâmetros PD7 e PD7H1 nos chamam a atenção pois atingem um ponto estacionário em apenas 9 iterações, enquanto GN leva 19 iterações e é cerca de 3.5 vezes mais lento.

Na Tabela 7.3 estão discriminados alguns dados de cada iteração realizada pelos métodos de Gauss-Newton e Levenberg-Marquardt com parâmetro de *damping* PD7. Observe que em ambos os casos, a ordem de convergência é quadrática, contudo, as escolhas das direções realizadas por LM, além de reduzir o valor gradiente aplicando no iterando mais rapidamente, mostram-se mais consistentes, uma vez que as variações abruptas nos termos da sequência definida por  $\left\{ \frac{\|x_{k+1} - x_*\|_2}{\|x_k - x_*\|_2^2} \right\}_{k \in \mathbb{N}}$ , são menos frequentes.

Para o caso em que são avaliados 5000 pontos, conforme a Tabela 7.12, nenhum método conseguiu resolver o problema. O método de Gauss-Newton falhou devido a um erro de *overflow*, já o método de Levenberg-Marquardt falhou para todos os parâmetros de *damping* testados, atingindo o tempo máximo de execução no caso de PD6 e o número máximo de iterações nos demais casos. Ao analisarmos a norma do gradiente na última iteração, nota-se que a mesma é relativamente pequena para PD3, PD4, PD5, PD7H2, PD7H3, PD8H2 e PD8H3, o que pode indicar o algoritmo estava convergindo muito lentamente.

Mét.	It.	Iterando $x_k$	$\frac{\ x_{k+1} - x_*\ _2}{\ x_k - x_*\ _2}$	$\frac{\ x_{k+1} - x_*\ _2}{\ x_k - x_*\ _2^2}$
GN	0	[0.0, 0.0, 0.0, 0.0]	—	—
	1	[3.21184902, 0.0, 0.0, 3.21184902]	$1.11685341 * 10^0$	0.04323018
	2	[0.22318091, -1.19381991, -9.59915072, 0.22318091]	$9.59259488 * 10^{-1}$	0.03324534
	3	[0.78092905, -0.98332093, -9.47467209, -1.55628518]	$9.36293582 * 10^{-1}$	0.03382756
	4	[0.86603981, -0.97076128, -9.40820663, -1.60146979]	$9.97157346 * 10^{-1}$	0.0384778
	5	[1.0369841, -0.94604865, -9.26016208, -1.69400697]	$9.94050827 * 10^{-1}$	0.03846728
	6	[0.94425659, -0.94002455, -9.08640542, -1.7182833]	$9.83128738 * 10^{-1}$	0.03827231
	7	[1.03558059, -0.93404543, -9.09938117, -1.7428143]	$1.01312733 * 10^0$	0.04011695
	8	[1.20195292, -0.91028249, -8.92708922, -1.84206471]	$9.93511815 * 10^{-1}$	0.03883049
	9	[1.51822436, -0.86443745, -8.54694584, -2.04899826]	$9.86394883 * 10^{-1}$	0.0388041
	10	[2.03686504, -0.77886791, -7.68352967, -2.49701734]	$9.70889364 * 10^{-1}$	0.03872093
	11	[1.99868244, -0.77858536, -7.66077093, -2.4990291]	$9.99708382 * 10^{-1}$	0.04106573
	12	[3.64197372, -0.4894547, -4.37635352, -4.55972764]	$8.83045840 * 10^{-1}$	0.03628409
	13	[3.75091, -0.48918438, -4.40216377, -4.56657944]	$1.00004839 * 10^0$	0.04653403
	14	[4.79518229, -0.21269284, -0.48519241, -11.58390267]	$6.58596182 * 10^{-1}$	0.03064417
	15	[3.04301118, -0.27505467, -0.5574836, -23.95758101]	$1.21308695 * 10^{-1}$	0.00857041
	16	[3.15201923, -0.25127336, -0.22577078, -25.47938593]	$9.36264584 * 10^{-2}$	0.05452757
	17	[3.1503522, -0.25190324, -0.23102529, -25.65104678]	$6.92096780 * 10^{-2}$	0.4305125
	18	[3.14995189, -0.25199996, -0.23228641, -25.64000181]	$6.25282299 * 10^{-4}$	0.05619896
19	[3.14994534, -0.252, -0.23228492, -25.64]	0.0	0.0	
LM PD7	0	[0.0, 0.0, 0.0, 0.0]	—	—
	1	[2.82163488, 0.0, 0.0, 2.82163488]	1.10094389	0.04257563
	2	[1.40710093, -0.53415837, -3.99129832, 1.40710093]	$9.61355644 * 10^{-1}$	0.03376874
	3	[2.65127906, -0.36588784, -2.24061411, -4.71360383]	$7.68345855 * 10^{-1}$	0.02807394
	4	[3.00660455, -0.34194286, -1.2985471, -15.80340139]	$4.70506472 * 10^{-1}$	0.02237461
	5	[3.16010786, -0.29375349, -0.83702334, -20.88738661]	$4.83868475 * 10^{-1}$	0.04890482
	6	[3.31728078, -0.25519257, -0.34033995, -24.62663892]	$2.11971295 * 10^{-1}$	0.04427658
	7	[3.33125541, -0.25235158, -0.29398219, -25.56078378]	$7.82490794 * 10^{-2}$	0.07710796
	8	[3.33283961, -0.25199858, -0.28870483, -25.64002365]	$3.81176082 * 10^{-4}$	0.00480028
	9	[3.33283419, -0.252, -0.28872287, -25.64]	0.0	0.0

Tabela 7.3: Convergência dos métodos de Gauss-Newton (GN) e Levenberg-Marquardt (LM) com *damping* PD7, para o teste Exponencial com 1000 pontos.

- Senoidal (Tipo 1)

Para o conjunto de 100 pontos, LM PD7 convergiu em apenas 18 iterações, enquanto GN levou 108 iterações, sendo este último cerca de 6.7 vezes mais lento. Como podemos ver na Tabela 7.13, dos 14 parâmetros de *damping* distintos testados, 10 parâmetros obtiveram sucesso em encontrar a solução do problema. Note ainda que GN realizou mais iterações que 70% dos parâmetros que obtiveram sucesso. Os parâmetros PD2, PD4, PD7H3 e PD8H3, excederam o número máximo de iterações.

Ao analisar a Tabela 7.14, GN convergiu em apenas 4 iterações, enquanto que LM PD7 e LM PD7H1 o fizeram em 12 iterações. Dado que os intervalos para 100 e 1000 pontos são os mesmos, pode-se supor que o maior número de informações dadas sobre o modelo neste segundo caso, auxiliou tais métodos a terem um melhor desempenho. Assim como no caso anterior, os parâmetros PD2 e PD4 também não convergiram, entretanto, PD7H3 e PD8H3 conseguiram encontrar uma solução com 334 iterações e 381 iterações, respectivamente.

Para o problema com 5000 pontos, GN, LM PD7 e LM PD7H1 continuaram apresentando os melhores resultados, convergindo em 14 iterações. Ao analisarmos a Tabela 7.15, pode-se notar que PD7H2 também apresenta boa convergência, com apenas 16 iterações. Por outro lado, LM PD1 e LM PD3 necessitam de 287 e 256 iterações, respectivamente, para determinar uma solução. Já LM PD2 e LM PD4 excederam o número máximo de iterações.

- Senoidal (Tipo 2)

Neste problema, apenas GN, LM PD1 e LM PD6 conseguiram resolver o problema para 100 pontos, com 28, 253 e 224 iterações, respectivamente. Conforme a Tabela 7.16, os demais parâmetros de *damping* excederam o número máximo de iterações.

No caso do conjunto de dados com 1000 pontos (Tabela 7.17), PD7, PD7H1 e PD7H2 convergem em apenas 10 iterações, enquanto GN leva 23 iterações, PD7H3 demora 12 iterações e PD6 converge com 406 iterações. Os demais parâmetros de *damping* atingiram o número máximo de iterações.

Por fim, ao analisarmos a Tabela 7.18, o problema com 5000 pontos é resolvido por GN em 39 iterações, enquanto LM PD3 necessita de apenas 29 iterações. O parâmetro

PD7 apresenta desempenho semelhante a GN, necessitando de 38 iterações. Já LM PD2 e LM PD5 atingem o número máximo de iterações e LM PD1 excede o tempo máximo de execução proposto.

- Logístico

Conforme exposto na Tabela 7.19, o problema com 100 pontos foi resolvido por praticamente todos os algoritmos, com exceção do parâmetro PD2, que atingiu o limite máximo de iterações. Com respeito ao número de iterações, GN convergiu para uma solução com apenas 8 iterações, já PD7 atingiu um ponto fracamente crítico em 12 iterações.

Já no caso em que são considerados 1000 pontos (Tabela 7.20), PD7 e PD7H1 resolvem este problema com apenas 9 iterações, 12 vezes mais rapidamente que GN, o qual leva pouco mais de 64 segundos para executar as 52 iterações necessárias.

Por outro lado, ao trabalharmos com 5000 pontos (Tabela 7.21), os métodos não chegam a realizar a primeira iteração, uma vez que a norma do gradiente é da ordem de  $10^{-10}$ , valor muito abaixo de nosso critério de parada. Contudo, ao alterarmos o ponto inicial para  $x_0 = (1.0, 0.0, 0.0)$ , GN consegue resolver o problema LOVO em apenas 2 iterações, já os parâmetros de *damping* PD5, PD7, PD7H1 e PD8 convergem para uma solução em 5 iterações. Vale a pena notar que este é o único problema para o qual LM PD2 conseguiu encontrar uma solução.

## 7.2 Tabelas

A seguir, são apresentadas tabelas com os dados referentes a cada teste numérico realizado. No corpo das tabelas são utilizadas as abreviações “Sol.”, “Nº. It.” e “T. (s)”, as quais significam “Solução”, “Número de Iterações” e “Tempo em segundos”, respectivamente.

Método		100 pontos		
		Sol.	Nº. It.	T. (s)
GN		$[-3.2531, 15.2347]$	3	0.105859
LM	PD1	$[-3.2531, 15.23469923]$	44	0.247535
	PD2	Número máximo de iterações atingido! Norma do gradiente: 6854.380694	...	...
	PD3	$[-3.2531, 15.2346999]$	25	0.151996
	PD4	$[-3.2531, 15.23469997]$	129	0.706702
	PD5	$[-3.2531, 15.23470001]$	7	0.055723
	PD6	$[-3.2531, 15.2347]$	10	0.106863
	PD7	$[-3.2531, 15.2347]$	5	0.045540
	PD7H1	$[-3.2531, 15.23470003]$	5	0.042755
	PD7H2	$[-3.2531, 15.23469992]$	6	0.048989
	PD7H3	$[-3.2531, 15.23469981]$	9	0.105394
	PD8	$[-3.2531, 15.2347]$	14	0.106038
	PD8H1	$[-3.2531, 15.23470004]$	14	0.106612
	PD8H2	$[-3.2531, 15.2347]$	15	0.105657
PD8H3	$[-3.2531, 15.23469914]$	17	0.105693	

Tabela 7.4: Desempenho dos algoritmos para o modelo Polinomial de Grau 1 para 100 pontos, com ponto inicial  $x_0 = (0.0, 0.0)$ .

Método		1000 pontos		
		Sol.	Nº. It.	T. (s)
GN		$[-3.2531, 15.2347]$	3	2.012894
LM	PD1	$[-3.2531, 15.23469991]$	49	8.377822
	PD2	Número máximo de iterações atingido! Norma do gradiente: 69046.231041	...	...
	PD3	$[-3.2531, 15.23469995]$	25	13.231418
	PD4	$[-3.2531, 15.2347]$	129	66.442164
	PD5	$[-3.2531, 15.2347]$	5	3.013325
	PD6	$[-3.2531, 15.2347]$	10	5.517157
	PD7	$[-3.2531, 15.23469995]$	4	2.455991
	PD7H1	$[-3.2531, 15.23469995]$	4	2.442593
	PD7H2	$[-3.2531, 15.2347]$	5	2.947074
	PD7H3	$[-3.2531, 15.23469999]$	6	3.447363
	PD8	$[-3.2531, 15.2347]$	14	7.547038
	PD8H1	$[-3.2531, 15.2347]$	14	7.621730
	PD8H2	$[-3.2531, 15.23470001]$	14	7.622357
PD8H3	$[-3.2531, 15.23469995]$	15	8.121022	

Tabela 7.5: Desempenho dos algoritmos para o modelo Polinomial de Grau 1 para 1000 pontos, com ponto inicial  $x_0 = (0.0, 0.0)$ .

Método	5000 pontos			
	Sol.	Nº. It.	T. (s)	
GN	$[-3.2531, 15.2347]$	5	26.652711	
LM	PD1	$[-3.25309994, 15.2346975]$	95	433.362808
	PD2	Número máximo de iterações atingido! Norma do gradiente: 108202.988044	...	...
	PD3	$[-3.2531, 15.2347]$	67	302.656063
	PD4	$[-3.2531, 15.23469999]$	140	627.057011
	PD5	$[-3.2531, 15.2347]$	11	52.450256
	PD6	$[-3.2531, 15.2347]$	34	157.438272
	PD7	$[-3.2531, 15.2347]$	9	43.601339
	PD7H1	$[-3.2531, 15.2347]$	9	43.588809
	PD7H2	$[-3.2531, 15.2347]$	10	48.003442
	PD7H3	$[-3.2531, 15.23469999]$	13	61.569564
	PD8	$[-3.2531, 15.2347]$	19	88.095573
	PD8H1	$[-3.2531, 15.2347]$	19	88.094177
	PD8H2	$[-3.2531, 15.23469999]$	21	97.336551
	PD8H3	$[-3.25309999, 15.23469971]$	26	119.797291

Tabela 7.6: Desempenho dos algoritmos para o modelo Polinomial de Grau 1 para 5000 pontos, com ponto inicial  $x_0 = (0.0, 0.0)$ .

Método	100 pontos			
	Sol.	Nº. It.	T. (s)	
GN	[1.125481, 2.531678, 3.147236, 0.589134]	3	0.034963	
LM	PD1	[1.125481, 2.53167803, 3.14723602, 0.58913205]	69	0.434630
	PD2	Número máximo de iterações atingido! Norma do gradiente: 8659252.028610	...	...
	PD3	[1.125481, 2.53167801, 3.14723601, 0.58913335]	29	0.209846
	PD4	[1.125481, 2.531678, 3.147236, 0.589134]	19	0.135640
	PD5	[1.125481, 2.531678, 3.147236, 0.58913397]	5	0.052243
	PD6	[1.125481, 2.531678, 3.147236, 0.589134]	12	0.106925
	PD7	[1.125481, 2.531678, 3.147236, 0.589134]	5	0.049327
	PD7H1	[1.125481, 2.531678, 3.147236, 0.589134]	6	0.050495
	PD7H2	[1.125481, 2.531678, 3.147236, 0.58913371]	10	0.054577
	PD7H3	[1.125481, 2.53167804, 3.14723603, 0.58913161]	9	0.107872
	PD8	[1.125481, 2.531678, 3.14723596, 0.58913375]	8	0.107702
	PD8H1	[1.125481, 2.53167802, 3.14723586, 0.58913305]	8	0.108521
	PD8H2	[1.125481, 2.531678, 3.147236, 0.58913391]	10	0.106788
	PD8H3	[1.125481, 2.53167803, 3.14723602, 0.58913237]	13	0.107537

Tabela 7.7: Desempenho dos algoritmos para o modelo Polinomial de Grau 3 para 100 pontos, com ponto inicial  $x_0 = (0.0, 0.0, 0.0, 0.0)$ .

Método	1000 pontos			
	Sol.	Nº. It.	T. (s)	
GN	[1.125481, 2.531678, 3.147236, 0.589134]	3	1.908246	
LM	PD1	[1.125481, 2.531678, 3.147236, 0.58913381]	90	44.818915
	PD2	Número máximo de iterações atingido! Norma do gradiente: 81099066.430522	...	...
	PD3	[1.125481, 2.531678, 3.147236, 0.589134]	35	17.565545
	PD4	[1.125481, 2.531678, 3.147236, 0.589134]	19	9.828387
	PD5	[1.125481, 2.531678, 3.147236, 0.589134]	5	2.855866
	PD6	[1.125481, 2.531678, 3.147236, 0.589134]	14	7.324220
	PD7	[1.125481, 2.531678, 3.147236, 0.58913399]	4	2.413086
	PD7H1	[1.125481, 2.531678, 3.147236, 0.58913399]	4	2.362329
	PD7H2	[1.125481, 2.531678, 3.147236, 0.589134]	5	2.851239
	PD7H3	[1.125481, 2.531678, 3.147236, 0.58913398]	6	3.416607
	PD8	[1.125481, 2.531678, 3.147236, 0.589134]	9	4.833584
	PD8H1	[1.125481, 2.531678, 3.147236, 0.589134]	9	4.826819
	PD8H2	[1.125481, 2.531678, 3.147236, 0.589134]	10	5.317053
PD8H3	[1.125481, 2.531678, 3.147236, 0.58913392]	10	5.329230	

Tabela 7.8: Desempenho dos algoritmos para o modelo Polinomial de Grau 3 para 1000 pontos, com ponto inicial  $x_0 = (0.0, 0.0, 0.0, 0.0)$ .

Método		5000 pontos		
		Sol.	Nº. It.	T. (s)
GN		[1.125481, 2.531678, 3.147236, 0.589134]	11	51.517229
LM	PD1	Número máximo de iterações atingido! Norma do gradiente: 2215.664853	...	...
	PD2	Número máximo de iterações atingido! Norma do gradiente: 167202111.745127	...	...
	PD3	Número máximo de iterações atingido! Norma do gradiente: 1947.807213	...	...
	PD4	[1.125481, 2.531678, 3.147236, 0.589134]	31	138.241856
	PD5	[1.125481, 2.531678, 3.147236, 0.589134]	20	91.401425
	PD6	[1.125481, 2.531678, 3.147236, 0.589134]	141	621.220938
	PD7	[1.12551715, 2.53155621, 3.1448488, 0.59190148]	23	104.250589
	PD7H1	[1.12551715, 2.53155621, 3.1448488, 0.59190148]	23	104.256350
	PD7H2	[1.12551715, 2.53155621, 3.1448488, 0.59190148]	24	108.354452
	PD7H3	[1.12551715, 2.53155621, 3.1448488, 0.5919015]	24	108.031295
	PD8	[1.125481, 2.531678, 3.147236, 0.589134]	15	68.954953
	PD8H1	[1.125481, 2.531678, 3.147236, 0.589134]	15	69.093730
	PD8H2	[1.12546573, 2.53125387, 3.14440377, 0.58718087]	16	73.581539
	PD8H3	[1.12543905, 2.53056059, 3.14032112, 0.58652329]	21	95.408044

Tabela 7.9: Desempenho dos algoritmos para o modelo Polinomial de Grau 3 para 5000 pontos, com ponto inicial  $x_0 = (0.0, 0.0, 0.0, 0.0)$ .

Método	100 pontos			
	Sol.	Nº. It.	T. (s)	
GN	[9.67494675, -0.252, -1.35443956, -25.63999991]	12	0.106459	
LM	PD1	[2.77473237, -0.25200029, -0.10545855, -25.63995646]	92	0.604968
	PD2	Número máximo de iterações atingido! Norma do gradiente: 6822.296610	...	...
	PD3	[1.3143019, -0.252, 0.64179451, -25.63999983]	103	0.707909
	PD4	Falhou! Matriz não definida positiva.	...	...
	PD5	[2.10933185, -0.252, 0.16872894, -25.64]	27	0.207104
	PD6	[1.41970308, -0.252, 0.56465243, -25.64]	54	0.408147
	PD7	[3.92662318, -0.252, -0.45267963, -25.64]	25	0.307541
	PD7H1	[3.92662318, -0.252, -0.45267963, -25.64000002]	25	0.306738
	PD7H2	[3.92399775, -0.25200015, -0.45201305, -25.63997772]	33	0.349665
	PD7H3	[3.31752846, -0.25200036, -0.28412535, -25.63994626]	96	0.708670
	PD8	Falhou! Matriz não definida positiva.	...	...
	PD8H1	[1.26094854, -0.252, 0.6832359, -25.63999961]	39	0.407838
	PD8H2	[1.25865447, -0.25200017, 0.6850543, -25.63997434]	47	0.432719
	PD8H3	[1.24497612, -0.25200035, 0.69597854, -25.639948]	110	0.821605

Tabela 7.10: Desempenho dos algoritmos para o modelo Exponencial para 100 pontos, com ponto inicial  $x_0 = (0.0, 0.0, 0.0, 0.0)$ .

Método	1000 pontos			
	Sol.	Nº. It.	T. (s)	
GN	[3.14994534, -0.252, -0.23228492, -25.64]	19	18.155945	
LM	PD1	[2.78047763, -0.25200003, -0.10752301, -25.63999549]	105	54.815678
	PD2	Número máximo de iterações atingido! Norma do gradiente: 9782.533701	...	...
	PD3	[1.31201815, -0.252, 0.64353366, -25.64]	118	61.839602
	PD4	[1.21403916, -0.252, 0.72114723, -25.64]	253	132.895697
	PD5	[3.35830099, -0.252, -0.29633501, -25.64]	14	7.732650
	PD6	[1.41685929, -0.252, 0.56665753, -25.64]	56	33.679803
	PD7	[3.33283419, -0.252, -0.28872287, -25.64]	9	5.419249
	PD7H1	[3.33283419, -0.252, -0.28872287, -25.64000001]	9	5.420480
	PD7H2	[3.33283218, -0.25200001, -0.28872237, -25.63999896]	11	6.418732
	PD7H3	[3.33147286, -0.25200002, -0.2883146, -25.63999729]	20	11.030430
	PD8	[1.25399535, -0.252, 0.68876545, -25.64]	31	22.155629
	PD8H1	[1.25399527, -0.252, 0.68876551, -25.64]	31	22.071200
	PD8H2	[1.2539704, -0.25200001, 0.68878523, -25.63999892]	32	22.682760
	PD8H3	[1.2533817, -0.25200003, 0.68925451, -25.63999601]	40	26.787572

Tabela 7.11: Desempenho dos algoritmos para o modelo Exponencial para 1000 pontos, com ponto inicial  $x_0 = (0.0, 0.0, 0.0, 0.0)$ .

Método		5000 pontos		
		Sol.	Nº. It.	T. (s)
GN		Falhou! Erro de <i>Overflow</i> .	...	...
LM	PD1	Número máximo de iterações atingido! Norma do gradiente: 182,666156	...	...
	PD2	Número máximo de iterações atingido! Norma do gradiente: 8899,21724	...	...
	PD3	Número máximo de iterações atingido! Norma do gradiente: 0,002485	...	...
	PD4	Número máximo de iterações atingido! Norma do gradiente: 0,016574	...	...
	PD5	Número máximo de iterações atingido! Norma do gradiente: 0,023525	...	...
	PD6	Falhou! Tempo máximo de execução atingido!	...	...
	PD7	Número máximo de iterações atingido! Norma do gradiente: 70.894313	...	...
	PD7H1	Número máximo de iterações atingido! Norma do gradiente: 17.1506335	...	...
	PD7H2	Número máximo de iterações atingido! Norma do gradiente: 0.020630	...	...
	PD7H3	Número máximo de iterações atingido! Norma do gradiente: 0.005883	...	...
	PD8	Número máximo de iterações atingido! Norma do gradiente: 38.786088	...	...
	PD8H1	Número máximo de iterações atingido! Norma do gradiente: 37.620207	...	...
	PD8H2	Número máximo de iterações atingido! Norma do gradiente: 0.153775	...	...
	PD8H3	Número máximo de iterações atingido! Norma do gradiente: 0.011494	...	...

Tabela 7.12: Desempenho dos algoritmos para o modelo Exponencial para 5000 pontos, com ponto inicial  $x_0 = (0.0, 0.0, 0.0, 0.0)$ .

Método		100 pontos		
		Sol.	Nº. It.	T. (s)
GN		$[-4.62228501, .96099116, 5.9810296, 54.66424742]$	108	0.810164
LM	PD1	$[40.53491566, 2.34506672, 7.33235701, 24.12182799]$	90	0.604968
	PD2	Número máximo de iterações atingido! Norma do gradiente: 6054.300176	...	...
	PD3	$[40.53588799, -2.34503036, 2.09241354, 24.12083187]$	166	1.046664
	PD4	Número máximo de iterações atingido! Norma do gradiente: 3083.666480	...	...
	PD5	$[40.53667953, -2.34500076, -10.47396311, 24.12002097]$	96	0.607413
	PD6	$[40.53666959, -2.34500114, -4.19077773, 24.12003115]$	44	0.322381
	PD7	$[40.53669995, 2.345, 7.33237061, 24.12000006]$	16	0.119594
	PD7H1	$[40.53651296, 2.34500699, 7.33236919, 24.12019162]$	18	0.137377
	PD7H2	$[40.53470153, 2.34507473, 7.33235537, 24.12204736]$	243	1.437511
	PD7H3	Número máximo de iterações atingido! Norma do gradiente: 0.063714	...	...
	PD8	$[40.5367, -2.345, 2.09240735, 24.12]$	56	0.415863
	PD8H1	$[40.53640498, -2.34501103, 2.0924096, 24.12030224]$	58	0.435731
	PD8H2	$[40.53468843, -2.34507522, 2.09242269, 24.12206078]$	284	1.7403340
	PD8H3	Número máximo de iterações atingido! Norma do gradiente: 0.074577	...	...

Tabela 7.13: Desempenho dos algoritmos para o modelo Senoidal (Tipo 1) para 100 pontos, com ponto inicial  $x_0 = (1.0, 1.0, 1.0, 1.0)$ .

Método		1000 pontos		
		Sol.	Nº. It.	T. (s)
GN		$[-0.77020757, 97.89359038, -11.1561248, 54.86377487]$	84	54.730941
LM	PD1	$[40.53651413, 2.34500691, 7.3323692, 24.12019047]$	123	66.668952
	PD2	Número máximo de iterações atingido! Norma do gradiente: 60683.494326	...	...
	PD3	$[40.5367, -2.345, 2.09240735, 24.12]$	191	105.924230
	PD4	Número máximo de iterações atingido! Norma do gradiente: 31262.209174	...	...
	PD5	$[40.5367, 2.345, 7.33237061, 24.12]$	36	19.247519
	PD6	$[40.5367, -2.345, -4.19077796, 24.12]$	46	27.963433
	PD7	$[40.5367, 2.345, 7.33237061, 24.12]$	12	6.618232
	PD7H1	$[40.53669838, 2.34500006, 7.3323706, 24.12000166]$	12	6.621149
	PD7H2	$[40.53648818, 2.34500787, 7.33236901, 24.12021705]$	42	21.849049
	PD7H3	$[40.53648189, 2.34500811, 7.33236896, 24.1202235]$	334	171.866487
	PD8	$[40.5367, -2.345, 2.09240735, 24.12]$	58	38.290948
	PD8H1	$[40.53667948, -2.34500076, 2.0924075, 24.12002103]$	58	38.214715
	PD8H2	$[40.53650946, -2.34500708, 2.09240879, 24.12019525]$	90	54.717686
	PD8H3	$[40.53648535, -2.34500798, 2.09240898, 24.12021996]$	381	204.13667

Tabela 7.14: Desempenho dos algoritmos para o modelo Senoidal (Tipo 1) para 1000 pontos, com ponto inicial  $x_0 = (1.0, 1.0, 1.0, 1.0)$ .

Método	5000 pontos			
	Sol.	Nº. It.	T. (s)	
GN	[18.70405868, 3.73152002, 2.59924567, 1.86299439]	14	78.570088	
LM	PD1	[40.5366658, 2.34500024, 1.04918526, 24.11996481]	287	1336.242692
	PD2	Número máximo de iterações atingido! Norma do gradiente: 1389.992044	...	...
	PD3	[40.53669953, 2.345, 1.04918531, 24.11999951]	256	1222.902665
	PD4	Número máximo de iterações atingido! Norma do gradiente: 44.561831	...	...
	PD5	[40.5367, 2.345, 1.04918531, 24.12]	51	211.733412
	PD6	[40.53669929, 2.345, 1.04918531, 24.11999931]	50	206.846086
	PD7	[40.5367, -2.345, 2.09240735, 24.12]	14	71.954003
	PD7H1	[40.5367, -2.345, 2.09240735, 24.12]	14	65.746305
	PD7H2	[40.53669987, -2.345, 2.09240735, 24.11999994]	16	81.184821
	PD7H3	[40.53669977, -2.345, 2.09240735, 24.11999978]	33	145.348243
	PD8	[40.5367, 2.345, 1.04918531, 24.12]	49	231.536584
	PD8H1	[40.53669948, 2.345, 1.04918531, 24.1199995]	50	237.0060432
	PD8H2	[40.53669873, 2.34500001, 1.04918531, 24.11999875]	61	294.617120
	PD8H3	[40.53665265, 2.34500033, 1.04918524, 24.11995129]	138	619.365303

Tabela 7.15: Desempenho dos algoritmos para o modelo Senoidal (Tipo 1) para 5000 pontos, com ponto inicial  $x_0 = (1.0, 1.0, 1.0, 1.0)$ .

Método		100 pontos		
		Sol.	Nº. It.	T. (s)
GN		$[-6.31896963, -20.16163018, 12.8251189, 4.43782843, -2.18673255]$	28	0.205536
LM	PD1	$[1.30158205 \cdot 10^3, 1.25376964 \cdot 10^{-2}, 1.06984756 \cdot 10^1, 6.09056718, 2.46013828]$	253	3.512366
	PD2	Número máximo de iterações atingido! Norma do gradiente: 698.322380	...	...
	PD3	Número máximo de iterações atingido! Norma do gradiente: 0.549613	...	...
	PD4	Número máximo de iterações atingido! Norma do gradiente: 24.561983	...	...
	PD5	Número máximo de iterações atingido! Norma do gradiente: 0.802211	...	...
	PD6	$[9.14700729 \cdot 10^2, 1.78406444 \cdot 10^{-2}, 1.06984756 \cdot 10^1, 6.09056715, 2.46013823]$	224	1.40844
	PD7	Número máximo de iterações atingido! Norma do gradiente: 0.030271	...	...
	PD7H1	Número máximo de iterações atingido! Norma do gradiente: 0.030271	...	...
	PD7H2	Número máximo de iterações atingido! Norma do gradiente: 0.036325	...	...
	PD7H3	Número máximo de iterações atingido! Norma do gradiente: 0.182500	...	...
	PD8	Número máximo de iterações atingido! Norma do gradiente: 0.147130	...	...
	PD8H1	Número máximo de iterações atingido! Norma do gradiente: 0.147130	...	...
	PD8H2	Número máximo de iterações atingido! Norma do gradiente: 0.1471230	...	...
	PD8H3	Número máximo de iterações atingido! Norma do gradiente: 0.210834	...	...

Tabela 7.16: Desempenho dos algoritmos para o modelo Senoidal (Tipo 2) para 100 pontos, com ponto inicial  $x_0 = (5.0, 5.0, 5.0, 5.0, 5.0)$ .

Método		1000 pontos		
		Sol.	Nº. It.	T. (s)
GN		$[-1.9347969, -30.02984379, 11.72168703, 5.35908927, 4.56569784]$	23	15.268214
LM	PD1	Número máximo de iterações atingido! Norma do gradiente: 0.000179	...	...
	PD2	Número máximo de iterações atingido! Norma do gradiente: 8183.953621	...	...
	PD3	Número máximo de iterações atingido! Norma do gradiente: 0.006045	...	...
	PD4	Número máximo de iterações atingido! Norma do gradiente: 257.717991	...	...
	PD5	Número máximo de iterações atingido! Norma do gradiente: 3.550304	...	...
	PD6	$[2.11437118 \cdot 10^3, 7.78940997 \cdot 10^{-3}, 1.05820001 \cdot 10^1, 6.15986727, 2.78765965]$	406	244.72609
	PD7	$[13.535, -11.86, 12.8239, 6.30077, 4.82982]$	10	6.218296
	PD7H1	$[13.535, -11.86, 12.8239, 6.30077, 4.82982]$	10	6.220834
	PD7H2	$[13.53499999, -11.86, 12.82389991, 6.30077006, 4.82982015]$	10	6.247277
	PD7H3	$[13.535, -11.86, 12.82389994, 6.30077004, 4.8298201]$	12	7.421560
	PD8	Número máximo de iterações atingido! Norma do gradiente: 1.547905	...	...
	PD8H1	Número máximo de iterações atingido! Norma do gradiente: 1.547905	...	...
	PD8H2	Número máximo de iterações atingido! Norma do gradiente: 1.547905	...	...
	PD8H3	Número máximo de iterações atingido! Norma do gradiente: 1.547905	...	...

Tabela 7.17: Desempenho dos algoritmos para o modelo Senoidal (Tipo 2) para 1000 pontos, com ponto inicial  $x_0 = (5.0, 5.0, 5.0, 5.0, 5.0)$ .

Método		5000 pontos		
		Sol.	Nº. It.	T. (s)
GN		[4.17751493, 4.5585277, 4.00583654, 5.73524163, 5.13502768]	39	200.243135
LM	PD1	Falhou! Tempo máximo de execução atingido!	...	...
	PD2	Número máximo de iterações atingido! Norma do gradiente: 1961.968976	...	...
	PD3	[4.85234377, 4.57892232, 4.86747609, 5.69819619, 4.95116078]	29	133.992000
	PD4	[4.24195139, 4.55441717, 4.17780306, 5.72755911, 5.16562759]	44	198.835192
	PD5	Número máximo de iterações atingido! Norma do gradiente: 44.447555	...	...
	PD6	[4.7661643, 4.56834657, 4.82972822, 5.70265406, 5.05764927]	37	204.364287
	PD7	[4.17751493, 4.5585277, 4.00583654, 5.73524163, 5.13502768]	38	199.131515
	PD7H1	[4.17751493, 4.5585277, 4.00583654, 5.73524163, 5.13502768]	38	169.470308
	PD7H2	[4.17751493, 4.5585277, 4.00583654, 5.73524163, 5.13502768]	38	177.690588
	PD7H3	[4.17751493, 4.5585277, 4.00583654, 5.73524163, 5.13502768]	39	172.691716
	PD8	[4.76616435, 4.56834657, 4.82972814, 5.70265406, 5.05764925]	36	173.571812
	PD8H1	[4.76616435, 4.56834657, 4.82972814, 5.70265406, 5.05764925]	36	150.452836
	PD8H2	[4.76616435, 4.56834657, 4.82972814, 5.70265406, 5.05764925]	36	165.153008
PD8H3	[4.76616435, 4.56834657, 4.82972814, 5.70265406, 5.05764925]	36	156.833659	

Tabela 7.18: Desempenho dos algoritmos para o modelo Senoidal (Tipo 2) para 5000 pontos, com ponto inicial  $x_0 = (5.0, 5.0, 5.0, 5.0, 5.0)$ .

Método		100 pontos		
		Sol.	Nº. It.	T. (s)
GN		$[-10.5772, -4.52081, 19.6434]$	8	0.106527
LM	PD1	$[-10.57721548, -4.52074336, 19.64311313]$	49	0.305148
	PD2	Número máximo de iterações atingido! Norma do gradiente: 139.949964	...	...
	PD3	$[-10.57720157, -4.52080284, 19.64336918]$	53	0.332963
	PD4	$[-10.5772, -4.52081, 19.6434]$	224	1.352620
	PD5	$[-10.57720665, -4.52078045, 19.64327274]$	34	0.218241
	PD6	$[-10.5772, -4.52080999, 19.64339996]$	21	0.151216
	PD7	$[-10.57720026, -4.52080765, 19.64338983]$	12	0.106045
	PD7H1	$[-10.57720129, -4.52080445, 19.64337612]$	13	0.107661
	PD7H2	$[-10.57721906, -4.52072799, 19.64304695]$	53	0.344844
	PD7H3	$[-10.57722287, -4.52071155, 19.64297619]$	420	2.511331
	PD8	$[-10.5772, -4.52080997, 19.64339985]$	20	0.136464
	PD8H1	$[-10.57720109, -4.52080529, 19.64337974]$	21	0.143546
	PD8H2	$[-10.577219, -4.52072825, 19.64304806]$	61	0.405484
PD8H3	$[-10.57722252, -4.52071306, 19.64298269]$	428	2.533959	

Tabela 7.19: Desempenho dos algoritmos para o modelo Logístico para 100 pontos, com ponto inicial  $x_0 = (0.0, 0.0, 0.0)$ .

Método	1000 pontos			
	Sol.	Nº. It.	T. (s)	
GN	$[-10.5772, -4.52080999, 19.64339996]$	52	64.039182	
LM	PD1	$[-10.57720188, -4.52080135, 19.64336288]$	54	27.482617
	PD2	Número máximo de iterações atingido! Norma do gradiente: 250.466654	...	...
	PD3	$[-10.57720005, -4.52080977, 19.64339903]$	52	27.066106
	PD4	$[-10.5772, -4.52080999, 19.64339997]$	226	116.990977
	PD5	$[-10.57720027, -4.52080863, 19.64339412]$	17	9.530740
	PD6	$[-10.57720011, -4.52080931, 19.64339702]$	21	12.480539
	PD7	$[-10.5772, -4.52080999, 19.64339995]$	9	5.317855
	PD7H1	$[-10.57720017, -4.52080921, 19.64339659]$	9	5.130045
	PD7H2	$[-10.5772012, -4.5208045, 19.64337638]$	15	8.13716
	PD7H3	$[-10.57720181, -4.52080166, 19.64336418]$	60	30.815881
	PD8	$[-10.57719998, -4.52080989, 19.64339952]$	20	10.868267
	PD8H1	$[-10.57720093, -4.5208054, 19.64338024]$	20	10.872850
	PD8H2	$[-10.57720208, -4.52080044, 19.64335894]$	26	13.852144
	PD8H3	$[-10.57720178, -4.5208018, 19.64336481]$	71	36.483479

Tabela 7.20: Desempenho dos algoritmos para o modelo Logístico para 1000 pontos, com ponto inicial  $x_0 = (0.0, 0.0, 0.0)$ .

Método		5000 pontos		
		Sol.	Nº. It.	T. (s)
GN		[0.0, 0.0, 0.0]	0	4.014437
LM	PD1	[0.0, 0.0, 0.0]	0	3.851189
	PD2	[0.0, 0.0, 0.0]	0	3.847559
	PD3	[0.0, 0.0, 0.0]	0	3.819020
	PD4	[0.0, 0.0, 0.0]	0	3.847733
	PD5	[0.0, 0.0, 0.0]	0	3.833313
	PD6	[0.0, 0.0, 0.0]	0	3.844449
	PD7	[0.0, 0.0, 0.0]	0	3.838295
	PD7H1	[0.0, 0.0, 0.0]	0	3.853158
	PD7H2	[0.0, 0.0, 0.0]	0	3.914814
	PD7H3	[0.0, 0.0, 0.0]	0	3.831182
	PD8	[0.0, 0.0, 0.0]	0	3.914235
	PD8H1	[0.0, 0.0, 0.0]	0	3.913872
	PD8H2	[0.0, 0.0, 0.0]	0	3.849020
	PD8H3	[0.0, 0.0, 0.0]	0	3.847608

Tabela 7.21: Desempenho dos algoritmos para o modelo Logístico para 5000 pontos, com ponto inicial  $x_0 = (0.0, 0.0, 0.0)$ .

Método	5000 pontos			
	Sol.	Nº. It.	T. (s)	
GN	[4.54220755, 0.49111652, 17.56182617]	2	13.833576	
LM	PD1	$[8.44993112 \cdot 10^{-5}, -1.15560333, 4.27720328 \cdot 10^{-1}]$	33	167.160442
	PD2	$[1.21091853 \cdot 10^{-4}, -1.34899182, 5.02368440 \cdot 10^{-1}]$	131	649.585568
	PD3	$[5.94468155 \cdot 10^{-6}, -1.16014369, 4.56289663 \cdot 10^{-1}]$	9	48.319645
	PD4	$[9.39132411 \cdot 10^{-10}, -1.21905025, 5.62095706 \cdot 10^{-1}]$	8	43.303053
	PD5	$[3.26782714 \cdot 10^{-6}, -4.19209250 \cdot 10^{-1}, 3.84968257 \cdot 10^{-1}]$	5	28.487410
	PD6	$[4.61031920 \cdot 10^{-8}, -1.23884210, 6.29550703 \cdot 10^{-1}]$	8	43.397223
	PD7	$[2.52340910 \cdot 10^{-11}, -3.06284228 \cdot 10^{-1}, 4.26358584 \cdot 10^{-1}]$	5	28.393891
	PD7H1	$[1.32624891 \cdot 10^{-8}, -3.06284228 \cdot 10^{-1}, 4.26358584 \cdot 10^{-1}]$	5	28.617207
	PD7H2	$[1.03696998 \cdot 10^{-6}, -3.22243513 \cdot 10^{-1}, 4.18344537 \cdot 10^{-1}]$	6	33.579359
	PD7H3	$[2.92608442 \cdot 10^{-6}, -4.47369988 \cdot 10^{-1}, 3.69078932 \cdot 10^{-1}]$	13	68.151358
	PD8	$[4.58839428 \cdot 10^{-7}, -5.13852112 \cdot 10^{-1}, 4.91624563 \cdot 10^{-1}]$	5	28.577758
	PD8H1	$[8.16018276 \cdot 10^{-9}, -5.13972117 \cdot 10^{-1}, 4.91630444 \cdot 10^{-1}]$	6	33.47947
	PD8H2	$[1.33634538 \cdot 10^{-6}, -5.50538858 \cdot 10^{-1}, 4.31305604 \cdot 10^{-1}]$	8	43.341238
PD8H3	$[9.87202145 \cdot 10^{-6}, -6.91363394 \cdot 10^{-1}, 3.88107399 \cdot 10^{-1}]$	24	122.170345	

Tabela 7.22: Desempenho dos algoritmos para o modelo Logístico para 5000 pontos, com ponto inicial  $x_0 = (1.0, 0.0, 0.0)$ .

---

# CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

---

Esta dissertação teve como foco o estudo do problema de Otimização de Menor Valor Ordenado (LOVO), onde dadas uma família  $\mathcal{F}$  composta por  $r \in \mathbb{N}$  funções reais definidas em  $\Omega \in \mathbb{R}^n$ , e um inteiro  $p \in \{1, 2, \dots, r\}$ , desejamos minimizar a soma das funções, tomados os  $p$  menores valores, isto é, pretendemos resolver o problema de minimização

$$\min S_p(x) = \sum_{j=1}^p F_{i_j(x)}(x),$$

onde  $F \in \mathcal{F}$ ,  $\{i_1(x), i_2(x), \dots, i_r(x)\} = \{1, 2, \dots, r\}$  e  $F_{i_1(x)}(x) \leq F_{i_2(x)}(x) \leq \dots \leq F_{i_p(x)}(x) \leq \dots \leq F_{i_r(x)}(x)$ . De modo particular, nos debruçamos sobre uma generalização do problema de quadrados mínimos não linear, o qual pode ser visto como um subproblema LOVO.

Após uma breve revisão de conceitos relativos à Álgebra Linear e Otimização, introduzimos o problema de quadrados mínimos linear e não linear, assim como os métodos de Newton e Gauss-Newton para o caso não linear.

Em seguida, apresentamos a teoria do Método de Levenberg-Marquardt e um algoritmo clássico. Propomos dois novos parâmetros de *damping* baseados na estratégia apresentada na Seção 4.2, além de seis estratégias híbridas. Também desenvolvemos uma variação de Levenberg-Marquardt com Busca Linear, o qual apresenta convergência global, sob determinadas hipóteses.

Ao abordarmos o problema LOVO e o subproblema de quadrados mínimos não linear generalizado, estudamos algoritmos com convergência para pontos fracamente críticos,

uma vez que a convergência para pontos fortemente críticos é, de fato, um problema muito complexo, fugindo ao escopo deste trabalho.

No âmbito do problema LOVO irrestrito, sugerimos a implementação de uma modificação do Método de Levenberg-Marquardt com Busca Linear, o qual converge globalmente para pontos fracamente críticos caso o problema satisfaça as hipóteses de convergência. Vale destacar que sistemas lineares satisfazem naturalmente tais hipóteses e portanto, há um conjunto não trivial de problemas cujas especificidades satisfazem as condições de convergência global.

Por fim, ao analisarmos comparativamente os dados relativos a performance do método proposto e do Método de Gauss-Newton, evidencia-se que os problemas lineares foram solucionados mais rapidamente por este último. Já ao observarmos os dados relacionados ao conjunto de problemas não lineares, há uma tendência de que o Método de Levenberg-Marquardt com Busca Linear resolva mais rapidamente, em termos de tempo e número de iterações, os problemas de maior dimensão, principalmente ao escolher os parâmetros PD7 e PD7H1. Tal comportamento justifica-se pois o método de Levenberg-Marquardt associa o baixo custo computacional do método de Gauss-Newton (as segundas derivadas da função objetivo não são necessárias) à boa definição dos iterandos (as direções  $d(\lambda_k)$  sempre estão bem definidas, dada positividade da matriz  $J(x_k)^T J(x_k) + \lambda_k I$ ).

Na Tabela 8.1, estão reunidas informações acerca do desempenho dos métodos de Gauss-Newton e Levenberg-Marquardt com Busca Linear. Note que entre os algoritmos mais robustos (maior número de sucessos), encontram-se GN, LM PD6, LM PD7, LM PD7H1 e LM PD7H2. Dentre os algoritmos que apresentam maior velocidade de execução (em termos de menor número de iterações e menor tempo), destacam-se principalmente GN, LM PD7, LM PD7H1. Deste modo, podemos concluir que os parâmetros de *damping* propostos PD7 (Schwertner 01) e PD7H1 (Schwertner 01, híbrido 01) apresentaram os melhores desempenhos quando comparados a Gauss-Newton.

Sigla	Desempenho		
	Nº. de Sucessos (%)	Velocidade (Iterações)	Velocidade (Tempo)
GN	18 (94,74 %)	11	8
PD1	15 (78,95 %)	0	0
PD2	1 (5,26 %)	0	0
PD3	15 (78,95 %)	1	2
PD4	12 (63,16 %)	0	0
PD5	15 (78,95 %)	0	0
PD6	18 (94,74 %)	0	0
PD7	17 (89,47 %)	6	5
PD7H1	17 (89,47 %)	5	3
PD7H2	17 (89,47 %)	1	0
PD7H3	16 (84,21 %)	0	0
PD8	16 (84,21 %)	0	0
PD8H1	16 (84,21 %)	0	0
PD8H2	16 (84,21 %)	0	0
PD8H3	15 (78,95 %)	0	0

Tabela 8.1: Performance dos métodos empregados nos testes numéricos.

Tendo em vista o aprofundamento dos temas contemplados nesta dissertação, pode-se tratar o Método de Levenberg-Marquardt como um caso particular do Método de Região de Confiança, estendendo resultados clássicos desta abordagem para o contexto dos problemas LOVO. Outras propostas de extensão poderiam incluir o estudo de condições para convergência à pontos fortemente críticos, o estudo do comportamento do método em conjuntos de dados espaçados de maneira heterogênea, ou ainda, a solução de problemas LOVO mal escalados.

---

## REFERÊNCIAS BIBLIOGRÁFICAS

---

- [1] Allaire, G., e Kaber, S. M. (2008), *Numerical linear algebra* (Vol. 55), New York: Springer.
- [2] Andreani, R., Cesar, G., Cesar-Jr, R. M., Martínez, J. M., e Silva, P. J. S. (2007). Efficient curve detection using a Gauss-Newton method with applications in agriculture., *Proc. 1st International Workshop on Computer Vision Applications for Developing Regions in Conjunction with ICCV, 2007-CVDR-ICCV07*.
- [3] Andreani, R., Martínez, J. M., Martínez, L., e Yano, F. S. (2005). Low order-value optimization and applications (Technical report MCDO 051013). Campinas: IMECC, Universidade Estadual de Campinas.
- [4] Behling, R. (2011), *The method and the trajectory of Levenberg-Marquardt*, Tese de Doutorado, IMPA, Rio de Janeiro, RJ.
- [5] Benatti, K. A. (2017), O Método de Levenberg-Marquardt para o Problema de Quadrados Mínimos não Linear, Dissertação de Mestrado, Universidade Federal do Paraná, Curitiba, PR.
- [6] Cabral, I., Perdigão, C., e Saiago, C. (2012), *Álgebra Linear: Teoria, Exercícios resolvidos e Exercícios propostos com soluções*, 3ª edição, Lisboa: Escolar Editora.
- [7] Davies, M., e Whitting, I. J. (1972), A modified form of Levenberg's correction, In Lootsma, L. A. (editor), *Numerical Methods for Nonlinear Optimization*, London: Academic Press.
- [8] Demmel, J. W. (1997), *Applied numerical linear algebra* (Vol. 56), Philadelphia: Siam.

- 
- [9] Fan, J. Y., e Yuan, Y. X. (2001). On the convergence of a new Levenberg-Marquardt method. *Technical Report, AMSS, Chinese Academy of Sciences*, 1-11.
- [10] Friedlander, A. (2012), *Elementos de programação não-linear*, Campinas: UNICAMP.
- [11] Guidorizzi, H. L. (2001), *Um curso de cálculo* (Vol. 1), Rio de Janeiro: LTC.
- [12] Intriligator, M. D. (2002), *Mathematical Optimization and Economic Theory*, Philadelphia: Siam.
- [13] Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares, *Quarterly of Applied Mathematics*, 2(2), 164-168.
- [14] Lima, E. L. (2006). Análise Real: Funções de uma variável (Vol. 1). *Coleção Matemática Universitária*, Rio de Janeiro: IMPA.
- [15] Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters, *Journal of the Society for Industrial and Applied Mathematics*, 11(2), 431-441.
- [16] Martínez, J. M., e Santos, S. A. (1995). Métodos computacionais de otimização. *Colóquio Brasileiro de Matemática*, Apostilas, 20.
- [17] Meyer, C. D. (2000). *Matrix analysis and applied linear algebra* (Vol. 71), Philadelphia: Siam.
- [18] Milies, C. P. (2004). Breve história da álgebra abstrata. *II Bienal da Sociedade Brasileira de Matemática* ([www.bienasbm.ufba.br/M18.pdf](http://www.bienasbm.ufba.br/M18.pdf)).
- [19] Nocedal, J., e Wright, S. J. (2006), *Numerical Optimization*, 2 edition, New York: Springer.
- [20] Santos, R. N. (2007), *Uma breve história do desenvolvimento das teorias dos determinantes e das matrizes*, São Paulo: IME, USP (<http://milanesa.ime.usp.br/imath/files/1/43.pdf>).
- [21] Santos Fonseca, J. (2017), Método de Newton generalizado e Aplicações, Dissertação de Mestrado, Universidade Federal do Pará, Belém, PA.

- [22] Watkins, D. S. (2010), *Fundamentals of matrix computations* (Vol. 64), New York: John Wiley & Sons.
- [23] Yamashita, N., e Fukushima, M. (2001). On the rate of convergence of the Levenberg-Marquardt method. *Topics in Numerical Analysis*, p.239-249, Vienna: Springer.
- [24] Yano, F. S. (2006), Otimização da menor soma de valores ordenados, Tese de Doutorado, Universidade Estadual de Campinas, Campinas, SP.