

# Implementação do Método de Levenberg-Marquardt

André Felipe Zanella e Júlia Guizardi

Outubro, 2022

## 1 Introdução

Comumente encontramos problemas descritos em sistemas lineares da forma

$$Ax = b,$$

onde  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$  e  $x$  é um vetor em  $\mathbb{R}^n$  cujo valor deseja-se encontrar. Contudo, existem sistemas que não são expressos por equações lineares, chamados sistemas não lineares, iremos abordar uma técnica de resolução deste problema usando o método de Levenberg-Marquardt. Assim, consideremos a função  $R : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ,  $m \geq n$ , que forma o sistema de equações não lineares descrito por

$$R(x) = 0.$$

Sabendo que existe a possibilidade do sistema anterior não possuir solução, queremos minimizar a função objetivo

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

Podemos denotar cada uma das entradas de  $r_i : \mathbb{R}^n \rightarrow \mathbb{R}$ , da forma

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

Com isso, reescrevemos o problema

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

onde cada  $r_i$  é chamada de função residual.

Definamos  $J(x)$  como a matriz jacobiana de  $f$ , assim

$$J(x) = \left[ \frac{\partial r_i}{\partial x_j} \right] = \begin{bmatrix} \nabla r_1(x)^T \\ \nabla r_2(x)^T \\ \vdots \\ \nabla r_m(x)^T \end{bmatrix},$$

$$1 \leq i \leq m, 1 \leq j \leq m.$$

Desta forma, podemos reescrever a derivada de primeira e segunda ordem da função  $f$ , de modo à diminuir o custo computacional. Veja,

$$\nabla f(x) = \sum_{i=1}^m r_i(x) \nabla r_i(x) = J(x)^T r(x)$$

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

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

Com isso, é necessário o cálculo do vetor gradiente da função  $f$ , bem como o produto matricial  $J(x)^T J(x)$  e o termo  $r_i(x) \nabla^2$  para encontrar  $S(x)$ . Contudo, o valor de  $S(x)$  não influencia tanto o método quanto o termo que o acompanha,  $J(x)^T J(x)$  devido ao valor residual ser pequeno ou ao comportamento próximo ao da função ao redor do ponto de convergência, por isso, muitas vezes podemos desconsiderar seu uso.

## 2 Método de Levenberg-Marquardt

Um dos meios de resolver o problema de quadrados mínimos não lineares é o método de Levenberg - Marquardt. Vamos considerar o problema visto em 1, e um ponto inicial  $x_0$ , assim ao resolver o sistema linear

$$(J(x_0)^T J(x_0) + \lambda_0 I) d(\lambda_0) = -J(x_0)^T R(x_0),$$

sendo  $J, R$  já estabelecidos previamente e  $\lambda \in \mathbb{R}_+$  o parâmetro que será discutido posteriormente, chamado parâmetro de *damping*. Vamos encontrar  $d(\lambda)$ , uma direção mais próxima da solução do problema proposto.

Assim, considerando  $x_1 = x_0 + d$ , podemos resolver 3 continuamente, até encontramos  $x_k$  que satisfaça a condição estabelecida.

Veja que este sistema possui solução única, uma vez que a matriz  $J^T J + \lambda I$ , com  $\lambda > 0$  é definida positiva. Portanto, o método está bem definido. Além disso, os resultados a seguir garantem o bom funcionamento do método.

**Teorema 2.1** *Seja  $d(\lambda)$  a solução do problema*

$$(J(x_k)^T J(x_k) + \lambda_k I) d(\lambda) = -J(x_k)^T R(x_k), \quad (3)$$

*para um valor  $\lambda$ . Então,  $\|d(\lambda)\|_2^2$  é uma função contínua decrescente, ou seja,  $\lim_{\lambda \rightarrow \infty} \|d(\lambda)\|_2^2 = 0$ .*

**Teorema 2.2** *Seja  $\gamma(\lambda)$  o ângulo entre a direção  $d(\lambda)$  e a direção máxima de descida. Então,  $\gamma(\lambda)$  é uma função contínua monótona decrescente, tal que  $\lim_{\lambda \rightarrow \infty} \gamma(\lambda) = 0$ .*

Note ainda que o método de Levenberg-Marquardt pode ser entendido como uma evolução do método do gradiente e do método de Gauss-Newton. O método do gradiente consiste na escolha de  $d_k$  como sendo a direção  $-\nabla f(x_k)$  e a sequência gerada é dada por

$$x_{k+1} = x_k - t \nabla f(x_k)$$

Por outro lado, as direções consideradas pelo método de Gauss - Newton são da forma

$$J(x_k)^T J(x_k) d_k = -J^T(x_k) R(x_k).$$

Contudo, o método desenvolvido por Levenberg e melhorado por Marquardt mantém o baixo custo computacional do método de Gauss-Newton e o supera quanto a eficácia de seus iterandos, que estão sempre bem definidos independentemente do problema.

### 3 Parâmetro de *Damping*

Para o funcionamento do método resta entender o cálculo de  $\lambda$ , conhecido como parâmetro de *damping*.

O parâmetro de *damping* é responsável pelo controle da singularidade da matriz  $J(x_k)^T J(x_k)$  através da adição de uma matriz diagonal  $\lambda_k I$ , onde  $\lambda_k > 0$  é o parâmetro.

A escolha de um bom parâmetro de *damping* em uma iteração consiste na redução do valor da função objetivo e, simultaneamente garantir que o tamanho do passo tomado não seja muito pequeno, a fim de obtermos a convergência da sequência.

A tabela 3 apresenta alguns dos parâmetros presentes na literatura.

	Parâmetro	Autoria
DP1	$\lambda_k = \frac{\ J(x_k)^T R(x_k)\ _2^2}{f(x_k)}$	Levenberg, [5]
DP2	$\lambda_k = \ J(x_k)^T R(x_k)\ _2^2$	Levenberg modificado, [5, 1]
DP3	$\lambda_k = \ J(x_k)^T R(x_k)\ _2$	Levenberg modificado 2, [5, 1]
DP4	$\lambda_k = \ R(x_k)\ _2^2 = 2f(x)$	Yamashita e Fukushima, [7]
DP5	$\lambda_k = \ R(x_k)\ _2 = \sqrt{2f(x)}$	Fan e Yuan, [3]

Tabela 1: Alguns parâmetros de *damping*.

Na Seção 5 iremos abordar o impacto do uso de diferentes parâmetros de *damping*, inclusive o que ocorre para um valor fixado constante de  $\lambda$ . Previamente, vamos estudar a implementação do que já foi discutido na Seção 4.

## 4 Algoritmo

A partir da teoria vista nas Seções 2 e 3 podemos construir um algoritmo com intuito de resolver 1. Além do que já foi visto, vamos adicionar a Condição de Armijo, vista no Teorema 4.1, na implementação. Tal condição garante um decréscimo estável, ou seja, proporcional a cada passo que o programa efetua.

**Teorema 4.1** *Seja  $\alpha \in (0, 1)$  uma constante e  $x, d \in \mathbb{R}^n$ , tal que  $\nabla f(x) \neq 0$  e ainda,  $\nabla f(x)^T d < 0$ . Então, existe  $\epsilon$  que satisfaz  $f(x + td) \leq f(x) + \alpha t \nabla f(x)^T d$  para todo  $t \in (0, \epsilon]$ .*

Assim, podemos estabelecer o Algoritmo 1 usando o método de Levenberg-Marquardt visto em [6].

**Algoritmo 1:** Método de Levenberg-Marquardt com busca linear

```

1 Entrada:  $R(x), J(x), x_0, \alpha \in (0, 1)$  e  $\epsilon > 0$ 
2 Saída:  $x^*$  (ponto estacionário)
3  $k = 0$ ;
4 while  $J(x_k)^T R(x_k) > \epsilon$  do
5   Calcule  $\lambda_k$ ;
6   Obtenha  $d(\lambda_k)$ , tal que
7     
$$(J(x_0)^T J(x_0) + \lambda_0 I) d(\lambda_0) = -J(x_0)^T R(x_0)$$

8    $t = 1$ ;
9   while  $f(x + td) \geq f(x) + \alpha t \nabla f(x)^T d$  do
10     $t = 0.5t$ 
11  end
12   $x_{k+1} = x_k + td(\lambda_k)$ 
13   $k = k + 1$ 
14 end

```

Para garantir a convergência do método é necessário fazermos algumas considerações. Primeiramente, é preciso que a escolha de  $\lambda$  seja definido entre constantes positivas. Além disso, para todo  $\lambda$  limitado, o ângulo entre a direção de descida do algoritmo e a direção de descida máxima também é limitado por constantes positivas. Por fim, a última hipótese necessária para que o menor autovalor da matriz  $J(x)^T J(x)$  seja limitado superiormente, para qualquer valor que  $x$  assumir.

Por fim, resta garantir a convergência do método.

**Lema 4.2** *Consideremos o algoritmo 1, satisfazendo todas as hipóteses anteriormente citadas. Portanto, existem constantes  $\beta > 0$  e  $\theta \in (0, 1)$  que cumprem*

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

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

**Teorema 4.3** *Se  $x^*$  é ponto limite da sequência de pontos gerada pelo Algoritmo 1, cumprindo as hipóteses anteriormente citadas, então  $\nabla f(x^*) = 0$ .*

Vale a pena citar que o algoritmo implementado neste trabalho, apesar de ter o Algoritmo 1 como base, não o segue perfeitamente. Primeiramente, nosso algoritmo tem como entrada uma matriz de dados  $A$ , e o tipo de função a ser modelada.

Com estes dois dados, é possível construir as funções residuais e posteriormente  $R$ , bem como a matriz jacobiana  $J$ . Além disso, como função auxiliar,

foi estabelecido um método de ter como entrada qual parâmetro de *damping* o usuário deseja utilizar.

Além disso, para facilitar o uso do programa, foi acrescentado, com entrada opcional, o vetor gradiente do modelo, tornando facultativo o cálculo da derivada pela máquina.

Também foi desenvolvido um algoritmo que recebe a matriz de dados e retira os pontos em que ocorre perturbação dos elementos, ou seja, o valor da função residual muito alto, com base na solução disponível para estes testes.

Por fim, foram realizadas algumas implementações que foram de suma importância para a leitura dos dados propostos, bem como os testes que serão analisados na Seção 5.

Todo o método foi implementado na linguagem **Julia** e esta disponível online junto a este relatório.

Visto isso, é possível aplicarmos o método e desenvolvermos testes capazes de testar sua eficácia na Seção 5 a seguir.

## 5 Testes

Foi estabelecida uma precisão de  $\epsilon = 10^{-10}$ , o valor  $\alpha = 0.5$  e um número máximo de iterações  $k = 1000$  para cada conjunto de dados fornecidos.

Os nossos testes foram separados em quatro modelos de funções, sendo eles funções quadráticas (Subseção 5.1), cúbicas (Subseção 5.2), logarítmicas (Subseção 5.3) e gaussianas (Subseção 5.4).

Para todos os modelos de funções, realizamos experimentos em um único conjunto de pontos, avaliando a matriz de dados com e sem perturbações, e também, executamos o algoritmo em vários conjuntos de dados, com o objetivo de avaliar o parâmetro de *damping* e a taxa de sucesso do algoritmo.

Os parâmetros de avaliação foram o número médio de iterações e a média do erro, definido como  $\|x^* - sol\|$ , onde *sol* é a solução real do problema.

### 5.1 Função Quadrática

Inicialmente, consideramos  $x_0 = (1, 0, 0)$  e o parâmetro de *damping* DP3 em um dos conjuntos de dados a serem analisados pela função quadrática. O resultado do método foi o ponto  $x^* = (-1.6568, -6.0813, 0.1420)$  em apenas 14 iterações, enquanto a solução real do problema é  $(-9.5, -8.5, 2.5)$ . Veja o resultado na Figura 1 (a).

Utilizamos o programa para retirar a perturbação da matriz de dados e aplicamos no método. Com isso, naturalmente, obtemos uma melhor solução para o problema, consideravelmente próxima da solução dada. Obtemos, com

o parâmetro DP3 o ponto  $x^* = (-9.5, -8.5, 2.5)$  em 24 iterações. Observe na Figura 1 (b).

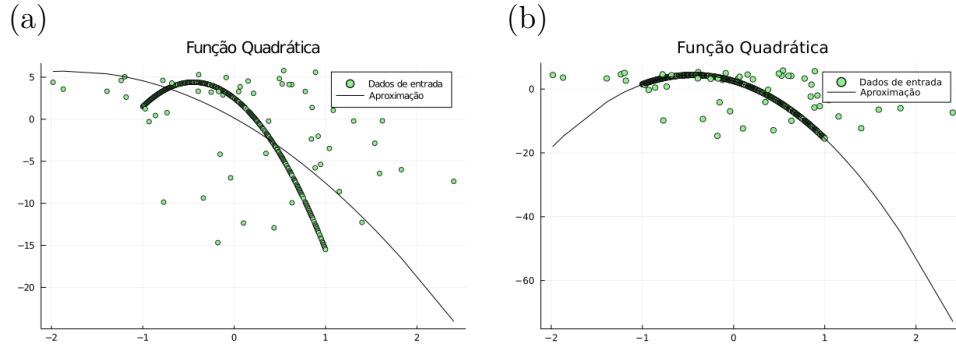


Figura 1: Função Quadrática - (a) Aproximação usando todo conjunto de dados. (b) Aproximação feita sem perturbação dos pontos de entrada.

Ademais, como um teste geral foram avaliados 262 conjuntos de dados com aproximações quadráticas em cada um dos parâmetros, bem como para o valor fixado  $\lambda = 0.5$ . O ponto inicial foi  $x_0 = (0, 0, 0)$ .

Foi possível perceber que para os parâmetros DP1, DP3, DP4, DP5 e 0.5 como valor fixado, foi possível encontrar soluções com baixa taxa de erro. Vale a pena destacar que o parâmetro DP1 obteve esse resultado com uma média de poucas iterações, sendo o mais eficiente.

Vale notar que na maioria dos casos a maior diferença entre os resultados obtidos por parâmetros diferentes é o número médio de iterações.

Parâmetro	Média das iterações	Média do erro
DP1	16.7290	3.6247
DP2	990.9580	9.6436
DP3	61.1946	3.6247
DP4	837.4885	3.5187
DP5	228.3893	3.6247
0.5	59.4923	3.6247

## 5.2 Função Cúbica

Seja  $x_0 = (1, 0, 0, 0)$  e o parâmetro de *damping* DP3 em um dos conjuntos de dados a serem analisados pela função cúbica. O resultado do método foi o ponto  $x^* = (-0.0517, 3.9730, -9.4655, -3.4345)$  em apenas 24 iterações, enquanto a solução real do problema era  $(0, 5, -10, -6)$ . Veja o resultado na Figura 2 (a).

Realizamos os mesmos testes, porém desconsiderando a perturbação. Com o parâmetro DP3 obtemos o ponto  $x^* = (1.2181 \cdot 10^{-16}, 5.0, -10.0, -5.9999)$  em apenas 27 iterações. Observe na Figura 2 (b).

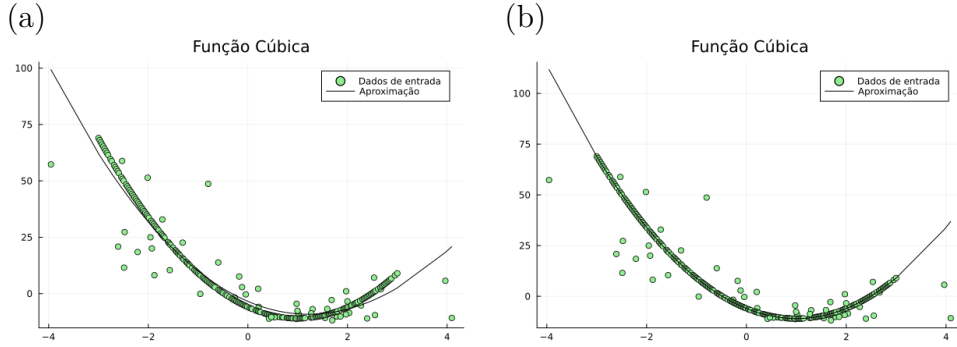


Figura 2: Função Cúbica - (a) Aproximação usando todo conjunto de dados. (b) Aproximação feita sem perturbação dos pontos de entrada.

Ademais, como um teste geral foram avaliados 252 conjuntos de dados com aproximações cúbicas em cada um dos parâmetros, bem como para o valor fixado  $\lambda = 0.5$ . Tomando novamente  $x_0 = (0, 0, 0, 0)$  obtivemos um resultado semelhante ao das funções quadráticas, apenas com uma taxa de erro maior. Novamente, o parâmetro DP1 obteve o melhor resultado com relação ao número de iterações.

Na maioria dos casos a maior diferença entre os resultados obtidos por parâmetros diferentes continua sendo o número médio de iterações.

Parâmetro	Média das iterações	Média do erro
DP1	13.3174	13.7759
DP2	998.8968	11.4691
DP3	49.9761	13.7759
DP4	953.4047	9.5601
DP5	359.9047	13.7759
0.5	46.0992	13.7759

### 5.3 Função Logarítmica

Podemos considerar  $x_0 = (1, 10, 10)$  e o parâmetro DP3. A análise em um único conjunto de dados forneceu o ponto  $x^* = (0.2550, 232.8216, 214.2468)$ , muito divergente da solução real, também atingindo o número máximo de iterações, 1000. O resultado pode ser visto na Figura 3 (a).



Com a análise com parâmetro DP3 e a matriz sem perturbações obtivemos  $x^* = (0.27102, 149.6531, 149.1531)$ , também atingindo o número máximo de iterações. Como ilustra a Figura 4 (b).

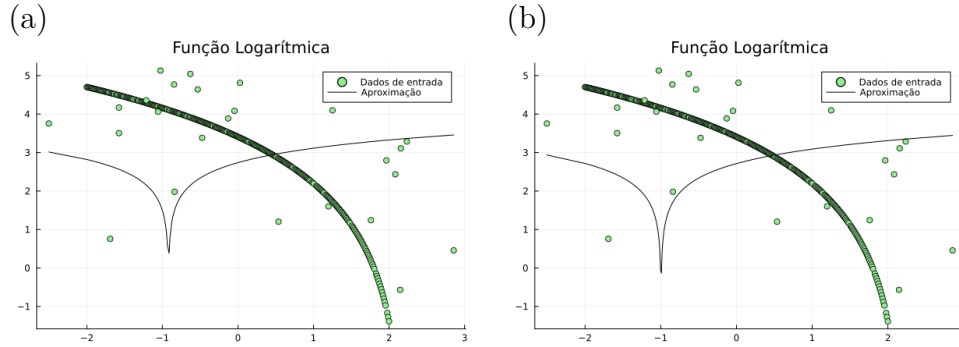


Figura 3: Função Logarítmica - (a) Aproximação usando todo conjunto de dados. (b) Aproximação feita sem perturbação dos pontos de entrada.

Foram consideradas 10 conjuntos de dados para os testes gerais. O ponto inicial considerado foi  $x_0 = (1, 10, 10)$ . A função logarítmica obteve resultados com um número bem maior de iterações com relação aos outros problemas. Contudo, mais uma vez, o parâmetro DP1 foi o que obteve maior sucesso, e o que vale destacar foi a média do erro, que foi bem menor comparado aos outros parâmetros.

Parâmetro	Média das iterações	Média do erro
DP1	586.9	5.6312
DP2	900.0	17.3617
DP3	820.6	160.8292
DP4	900.0	19.7859
DP5	816.1	45.7833
0.5	740.5	11.3073

## 5.4 Função Gaussiana

Para a função gaussiana, consideramos o ponto  $x_0 = (1, 10, 10)$  e o parâmetro de *damping* DP3, obtivemos o ponto  $x^* = (-4.5002, -2.9811, 7.0802)$  em apenas 62 iterações, e a solução do problema é  $(-4.5, -3, -7)$ . Veja o resultado na Figura 4 (a).

Sem a perturbação dos dados e com o parâmetro DP3, encontramos o ponto  $x^* = (-4.5, -2.9999, 6.9999)$  em 52 iterações. Observe na Figura 4 (b).

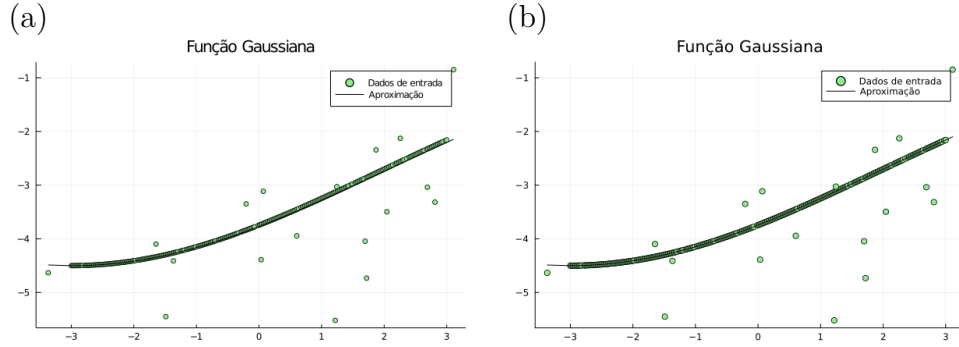


Figura 4: Função Gaussiana - (a) Aproximação usando todo conjunto de dados. (b) Aproximação feita sem perturbação dos pontos de entrada.

Aqui utilizamos o ponto de entrada  $x_0 = (1, 10, 10)$  e 10 conjuntos de dados. A função gaussiana obteve o melhor resultado para o parâmetro fixado como 0.5, apesar de não ter sido o menor número de iterações, foi encontrado uma média de erro melhor comparado aos outros parâmetros. Percebemos também que para os parâmetros DP4 e DP5 atingimos o número máximo de iterações, apesar disso, ele ainda obteve um erro menor que o encontrado em DP3.

Parâmetro	Média das iterações	Média do erro
DP3	446.0	80.5062
DP4	1000.0	15.0798
DP5	1000.0	13.4543
0.5	797.5	9.8600

## 6 Conclusão

O foco deste trabalho foi uma implementação eficaz do algoritmo gerado pelo método de Levenberg-Marquardt para o problema de quadrados mínimos não linear. Uma vez que tal problema tem grande aplicabilidade em diversas áreas do conhecimento.

Inicialmente, na Seção 1 apresentamos o problema a ser estudado, descrito por

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

A seguir, na Seção 2 vimos a abordagem proposta por Levenberg e aprimorada por Marquardt para resolver tal problema. Consequentemente, estudamos o parâmetro de *damping*, que pudemos observar pelos resultados vistos

na Seção 5 possui grande influência no resultado encontrado pelo algoritmo abordado na Seção 4, principalmente em relação ao número de iterações necessárias para a convergência do método.

Por fim, vale destacar que pudemos comprovar a boa convergência do método em muitos casos, além de expor resultados numéricos do uso do algoritmo.

## Referências

- [1] BENATTI, K. A. O método de levenberg-marquardt para o problema de quadrados mínimos não linear. In *II Simpósio de Métodos Numéricos em Engenharia* (2017).
- [2] BENATTI, K. A. O método de levenberg-marquardt para o problema de quadrados mínimos não linear. In *II Simpósio de Métodos Numéricos em Engenharia* (2017).
- [3] FAN, J., AND YUAN, Y. On the convergence of a new levenberg-marquardt method. *Thechnical Report, AMSS, Chinese Academy of Sciences* (2001).
- [4] FRIEDLANDER, A. *Elementos de programação não-linear*. Editora da UNICAMP, 1994.
- [5] LEVENBERG, K. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics* (1944).
- [6] SCHWERTNER, A. E. O método de levenberg-marquardt para problemas de otimização de menor valor ordenado.
- [7] YAMASHITA, N., AND FUKUSHIMA, M. On the rate of convergence of the levenberg-marquardt method. In *Topics in numerical analysis*. Springer, 2001.