

TEMA Tend. Mat. Apl. Comput., **14**, No. 1 (2013), 69-80.

doi: 10.5540/tema.2013.014.01.0069

© Uma Publicação da Sociedade Brasileira de Matemática Aplicada e Computacional.

Métodos de Regiões de Confiança para Resolução do Problema de Quadrados Mínimos: Implementação e Testes Numéricos¹

J.L.C. Gardenghi², Departamento de Ciência de Computação, IME - Instituto de Matemática e Estatística, USP - Universidade de São Paulo, 05508-090 São Paulo, SP, Brasil.

S.A. Santos³, Departamento de Matemática Aplicada, IMECC - Instituto de Matemática, Estatística e Computação Científica, UNICAMP - Universidade Estadual de Campinas, 13083-859 Campinas, SP, Brasil.

Resumo. O problema de quadrados mínimos possui várias aplicações no campo de otimização. No presente trabalho, abordamos duas estratégias para sua resolução: Levenberg-Marquardt e Gradientes Conjugados. Cada uma explora características próprias do problema, e ambas usam regiões de confiança para a globalização. Nossa contribuição está na implementação de ambos os métodos no *CAS Maxima* e na análise comparativa do desempenho desses métodos na resolução de uma família de problemas de quadrados mínimos da literatura.

Palavras-chave. Quadrados mínimos, regiões de confiança, implementação computacional.

1. Introdução

Neste trabalho, consideramos o problema de quadrados mínimos

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|F(x)\|^2 \equiv f(x), \quad (1.1)$$

sendo $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ a *função residual*, ao menos duas vezes continuamente diferenciável, $F(x)$ o *resíduo do problema* em x e $\|\cdot\|$ a norma Euclidiana. No campo de otimização, quadrados mínimos tem várias aplicações, destacando-se seu uso para ajustar parâmetros de modelos frente a conjuntos de dados coletados em experimentos, e neste contexto $m \gg n$. Para um recente panorama do estado da arte deste problema, tanto do ponto de vista teórico quanto prático, veja [14] e suas referências.

¹Este trabalho contou com os apoios do CNPq (304032/2010-7, 106731/2011-4 e 151749/2011-6), FAPESP (2012/05725-0) e PRONEX-Otimização; Trabalho apresentado no XXXIV CNMAC.

²john@ime.usp.br

³sandra@ime.unicamp.br

Na próxima seção, apresentamos dois métodos para resolução de (1.1): o algoritmo de Levenberg-Marquardt (LMA) e o método de Gradientes Conjugados (MGC). Em seguida, destacamos os detalhes da implementação e os resultados dos experimentos numéricos que a validam e ilustram o desempenho comparativo das duas abordagens distintas. Finalizamos o texto com uma breve conclusão.

2. Métodos de Regiões de Confiança para Quadrados Mínimos

Métodos de regiões de confiança para minimização irrestrita, $\min f(x)$ s.a. $x \in \mathbb{R}^n$, são estratégias iterativas baseadas em modelos quadráticos para a função objetivo f em torno de um dado ponto [2]. Em geral, seja $x_c \in \mathbb{R}^n$ o ponto corrente, define-se um modelo quadrático q_c para f em torno de x_c e o subproblema é formulado como

$$\min_{p \in \mathbb{R}^n} q_c(p) \text{ s.a. } \|p\| \leq \Delta_c, \quad (2.1)$$

em que $0 < \Delta_c \in \mathbb{R}$ é o raio da região de confiança. A solução p_c , exata ou aproximada, deste problema é usada para obter um ponto candidato $x_n = x_c + p_c$.

O bom desempenho de regiões de confiança está ligado ao controle do tamanho do raio, que determina a região em que o modelo é confiável para se aproximar a função objetivo. Nesta abordagem, vamos determinar o tamanho deste raio de acordo com o decréscimo que a solução de (2.1) causou no problema original que aproxima. Para isto, vamos avaliar a razão entre o decréscimo no modelo e na função objetivo, chamado quociente de redução relativa, dado por

$$\rho(p) \stackrel{\text{def}}{=} \frac{f(x_c) - f(x_c + p)}{q_c(0) - q_c(p)}. \quad (2.2)$$

No Algoritmo 1, página 71, apresentamos o algoritmo básico de regiões de confiança. Os parâmetros κ, η_1 e η_2 são determinados experimentalmente, embora haja algumas escolhas clássicas na literatura [2, 11]. Sob esta ótica, escolhemos $\kappa = 10^{-3}$, $\eta_1 = \frac{1}{4}$ e $\eta_2 = \frac{3}{4}$. Para resolver o subproblema (linha 2), implementamos LMA e MGC. Uma breve descrição destes dois métodos é apresentada a seguir.

2.1. Método de Levenberg-Marquardt

A principal característica deste método é atacar a estrutura particular do problema (1.1). Originalmente, esta estratégia foi proposta por Levenberg [6] e Marquardt [8], e foi considerada precursora da técnica de regiões de confiança (ver também [7]). Moré [9] propôs uma série de particularidades para implementação, que foram exploradas neste trabalho.

Dado um ponto $x_c \in \mathbb{R}^n$, LMA trabalha com a aproximação

$$q_c(p) \stackrel{\text{def}}{=} \frac{1}{2} \|J_c p + F_c\|^2$$

em torno de x_c , sendo $F_c \in \mathbb{R}^m$ o resíduo em x_c e $J_c \in \mathbb{R}^{m \times n}$ a jacobiana da função residual avaliada em x_c . Deste modo, o subproblema de região de confiança (2.1) é

Algoritmo 1: Regiões de Confiança

Entrada: Ponto inicial $x_0 \in \mathbb{R}^n$, raio inicial da região de confiança Δ_0 , constantes $\kappa \in [0, \frac{1}{4}]$, $0 < \eta_1 \leq \eta_2 < 1$.

```

1 para  $k = 0, 1, \dots$  hacer
2   Encontre  $p_k$  como solução (aproximada) do subproblema (2.1)
3    $x_{k+1} \leftarrow x_k + p_k$ 
4   Avalie o quociente de redução relativa  $\rho(p_k)$  (2.2)
5   se  $\rho(p_k) > \kappa$  então
6     | Aceite o ponto candidato  $x_{k+1}$  e avalie critérios de parada
7     | senão
8       | Descarte o ponto candidato e considere  $x_{k+1} = x_k$ 
9       | fim
10    se  $\rho(p_k) \leq \eta_1$  então Obtenha  $\Delta_{k+1}$  reduzindo  $\Delta_k$ 
11    senão
12      | se  $\rho(p_k) \geq \eta_2$  então Obtenha  $\Delta_{k+1}$  aumentando  $\Delta_k$ 
13      | senão  $\Delta_{k+1} \leftarrow \Delta_k$ 
14      | fim
15    fim
16 fin

```

um problema de quadrados mínimos lineares com passo restrito, cujas condições de otimalidade são dadas por

$$(J_c^T J_c + \lambda^* I)p^* = -J_c^T F_c \quad (2.3a)$$

$$\lambda^*(\|p^*\| - \Delta_c) = 0 \quad (2.3b)$$

$$\lambda^* \geq 0, \|p^*\| \leq \Delta_c. \quad (2.3c)$$

O sistema linear (2.3a) está bem definido somente se $(J_c^T J_c + \lambda^* I)$ for definida positiva. Como $J_c^T J_c$ é ao menos semidefinida positiva, então $(J_c^T J_c + \lambda^* I)$ será definida positiva sempre que $\lambda^* > 0$. Em particular, podemos ter dificuldades numéricas, por mal condicionamento ou por singularidade, quando $\lambda^* = 0$. Neste caso, de acordo com (2.3b), temos que a solução de (2.1) pode ser interna, ou seja, $\|p^*\| < \Delta_c$. O resultado a seguir estabelece condições para que a solução de (2.1) seja interior à região de confiança.

Proposição 2.1. *O problema (2.1) não possui solução global na fronteira se, e somente se, $J_c^T J_c$ é definida positiva e o passo de Newton é interior, ou seja, $\|(J_c^T J_c)^{-1} J_c^T F_c\| < \Delta_c$.*

Demonstração. Ver [4, Corolário 1]. □

Por outro lado, caso o problema possua solução na fronteira, é sempre possível encontrar λ^* que satisfaça $\|p^*\| = \Delta_c$. Assim sendo, consideramos

$$p(\lambda) \stackrel{\text{def}}{=} -(J_c^T J_c + \lambda I)^\dagger J_c^T F_c,$$

em que \dagger é a pseudoinversa de Moore-Penrose, e queremos encontrar $\lambda^* \geq 0$ tal que

$$\|p(\lambda^*)\| = \Delta_c. \quad (2.4)$$

A equação (2.4) é chamada de *equação secular*. Por conseguinte, nosso problema se reduz a encontrar um zero de uma equação em uma variável. Para tal, usaremos o Método de Newton, com algumas salvaguardas que serão apresentadas na Seção 3. Notemos que λ^* atua como um regularizador, ou seja, ele ajusta o sistema (2.3a) para que sempre exista uma solução p^* . É justamente essa a conexão de LMA com regiões de confiança. Originalmente [6, 8], o parâmetro λ^* era computado de forma heurística. Com regiões de confiança, buscamos o multiplicador de Lagrange da restrição quadrática $\lambda^* \geq 0$ de tal maneira que $\|p^*\| = \Delta_c$, salvo o caso particular em que a solução seja interna e então $\lambda^* = 0$.

2.2. Método de Gradientes Conjugados

Neste método, vamos atacar o problema (1.1) como um problema geral de minimização irrestrita, e resolvê-lo usando a estratégia de região de confiança, considerando, a cada iteração, o subproblema de região de confiança (2.1), tomando

$$q_c(p) \stackrel{\text{def}}{=} \frac{1}{2}p^T H_c p + g_c^T p + f_c$$

o modelo quadrático de f em torno de um ponto corrente x_c , com $H_c \in \mathbb{R}^{n \times n}$, $g_c \in \mathbb{R}^n$ e $f_c \in \mathbb{R}$, respectivamente, a matriz hessiana, o vetor gradiente e a função objetivo f , todos avaliados em x_c .

Para um estudo mais detalhado sobre a estratégia de gradientes conjugados, veja [5, Seção 3]. Para a resolução do subproblema, a ideia é produzir uma sequência $\{d_i\}$ de direções conjugadas e encontrar a solução para (2.1) iterativamente, por meio de $p_{i+1} = p_i + \alpha_i d_i$. MGC seria uma estratégia natural se H_c fosse uma matriz definida positiva, mas, na prática, nada sabemos sobre ela. É justamente neste ponto que as regiões de confiança tornam-se essenciais. As condições necessárias e suficientes de otimalidade de (2.1) são dadas por

$$(H_c + \lambda^* I)p^* = -g_c, \quad (2.5a)$$

$$\lambda^*(\|p^*\| - \Delta_c) = 0, \quad (2.5b)$$

$$\lambda^* \geq 0, \|p^*\| \leq \Delta_c, \quad (2.5c)$$

$$(H_c + \lambda^* I) \text{ é semidefinida positiva.} \quad (2.5d)$$

Toint [13] e Steihaug [12] propuseram algumas modificações no MGC original de tal maneira que se $\|g_c + H_c p_i\|$ for menor que dada tolerância, então temos uma solução interna, pois p_i satisfaz aproximadamente a condição (2.5a) com $\lambda^* = 0$. Por outro lado, se $\|p_{i+1}\| > \Delta_c$, então basta encontrar τ de forma que $\|p_i + \tau d_i\| = \Delta_c$. Por fim, se H_c não é definida positiva, então existe uma direção de curvatura negativa ou nula a partir de p_i . Assim sendo, se encontrarmos uma tal direção d_i , basta definir um passo que atinja a borda da região de confiança. Em outras palavras, se $d_i^T H_c d_i \leq 0$, computamos τ tal que $\|p_i + \tau d_i\| = \Delta_c$.

Deste modo, MGC atua como GC clássico quando H_c é definida positiva. Caso contrário, se for encontrada uma direção de curvatura negativa, determina-se um passo que atinja a borda da região de confiança. De qualquer forma, o passo tem seu tamanho restrito e essa também é uma condição de parada do método. No Algoritmo 2 descrevemos MGC.

Algoritmo 2: Gradientes Conjugados com Regiões de Confiança

Entrada: Ponto corrente $x_c \in \mathbb{R}^n$, a matriz hessiana $H_c \in \mathbb{R}^{n \times n}$ e o gradiente $g_c \in \mathbb{R}^n$ da função objetivo avaliados em x_c , Δ_c o raio da região de confiança, $\varepsilon > 0$ uma tolerância para o erro de otimalidade.

Saída: p direção para computar o novo ponto candidato $x_n = x_c + p$.

```

1  $p_0 \leftarrow 0, r_0 \leftarrow g_c, d_0 \leftarrow r_0$ 
2 para  $i = 0, 1, \dots$  fazer
3   se  $d_i^T H_c d_i \leq 0$  então
4     Encontre  $\tau$  tal que  $\|p_i + \tau d_i\| = \Delta_c$ 
5     retorna  $p_i + \tau d_i$ 
6   fim
7    $\alpha_i \leftarrow (r_i^T r_i) / (d_i^T H_c d_i)$ 
8    $p_{i+1} \leftarrow p_i + \alpha_i d_i$ 
9   se  $\|p_{i+1}\| > \Delta_c$  então
10    Encontre  $\tau$  tal que  $\|p_i + \tau d_i\| = \Delta_c$ 
11    retorna  $p_i + \tau d_i$ 
12  fim
13   $r_{i+1} \leftarrow r_i + \alpha_i H_c d_i$ 
14  se  $\|r_{i+1}\| \leq \varepsilon$  então retorna  $p_{i+1}$ 
15
16   $\beta_i \leftarrow (r_{i+1}^T r_{i+1}) / (r_i^T r_i)$ 
17   $d_{i+1} \leftarrow r_{i+1} + \beta_i d_i$ 
18 fim

```

3. Implementação Computacional

Nesta seção, apresentamos os detalhes de implementação do LMA que, por explorar as peculiaridades do problema (1.1), possui mais particularidades de implementação que o Algoritmo 2, que é um método para resolução do subproblema de regiões de confiança no caso de minimização irrestrita geral.

3.1. Álgebra matricial empregada na solução do subproblema

A equação (2.3a) pode ser vista como um sistema de equações normais para o problema linear de quadrados mínimos:

$$\begin{pmatrix} J_c \\ \sqrt{\lambda} I \end{pmatrix} p \cong - \begin{pmatrix} F_c \\ 0 \end{pmatrix}. \quad (3.1)$$

Tal equivalência permite resolver o problema sem avaliar o produto $J_c^T J_c$, reduzindo pela metade a quantidade de operações. Deste modo, se calcularmos a fatoração QR

$$J_c = Q \begin{pmatrix} R_c \\ 0 \end{pmatrix}$$

teremos

$$\begin{pmatrix} Q^T & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} J_c \\ \sqrt{\lambda}I \end{pmatrix} = \begin{pmatrix} R_c \\ 0 \\ \sqrt{\lambda}I \end{pmatrix}. \quad (3.2)$$

A matriz que resulta de (3.2) possui uma estrutura particular: R_c é uma matriz triangular superior, enquanto $\sqrt{\lambda}I$ é uma matriz escalar. Diante disto, é possível obter uma matriz triangular superior equivalente eliminando-se os n termos não-nulos da matriz $\sqrt{\lambda}I$ com uma sequência de $n(n+1)/2$ rotações de Givens aplicadas em cada linha i de $\sqrt{\lambda}I$, simultaneamente a cada linha $i+j$ de R_c , para $j = 0, \dots, (n-i)$, começando da linha n . Expressando tais rotações em uma matriz $G_\lambda \in \mathbb{R}^{(m+n) \times (m+n)}$ por meio de todos os sucessivos $n(n+1)/2$ produtos, vem que

$$G_\lambda^T \begin{pmatrix} R_c \\ 0 \\ \sqrt{\lambda}I \end{pmatrix} = \begin{pmatrix} R_\lambda \\ 0 \\ 0 \end{pmatrix}. \quad (3.3)$$

Disto, obtemos a matriz triangular superior R_λ , de modo que $R_\lambda^T R_\lambda = J_c^T J_c + \lambda I$. Por conseguinte, o sistema (2.3a) torna-se menos custoso de ser resolvido para diferentes valores de λ . De fato, a fatoração QR de J_c é calculada uma única vez, no início da iteração interna, enquanto, para cada valor de λ , obtém-se R_λ por (3.3), que consiste em um procedimento mais barato que o cálculo da fatoração QR propriamente dita.

3.2. Raiz da equação secular

O problema (2.4) consiste em encontrar $\lambda^* \geq 0$ tal que $\|p(\lambda^*)\| = \Delta_c$. Então, se definimos $\delta(\lambda) \stackrel{\text{def}}{=} p(\lambda) - \Delta_c$, estamos interessados em encontrar uma raiz para $\delta(\lambda)$. Todavia, uma análise mais apurada [4, Seção 3.2] nos mostra que a função δ pode possuir polos nos autovalores da matriz $J_c^T J_c + \lambda I$ e pode não possuir zeros finitos. Em vista disto, uma estratégia mais estável é resolver a equação secular

$$\phi(\lambda) \stackrel{\text{def}}{=} \frac{1}{\|p(\lambda)\|} - \frac{1}{\Delta_c} = 0, \quad (3.4)$$

que, por ser a recíproca de δ , não possui polos finitos e suas raízes acontecem nos valores negativos dos autovalores da matriz $J_c^T J_c + \lambda I$. Deste modo, para encontrar $\lambda^* \geq 0$ que satisfaça (2.3a), avaliamos primeiro o caso particular em que $\lambda^* = 0$ (solução interna). Se (2.3a) não possui solução neste caso, buscamos uma raiz positiva para a equação secular usando o método de Newton.

O método de Newton aplicado ao problema (3.4) gera uma sequência $\{\lambda_j\}$ tal que

$$\lambda_{j+1} = \lambda_j - \frac{\phi(\lambda_j)}{\phi'(\lambda_j)}. \quad (3.5)$$

Em contrapartida, seria interessante reduzir o custo do cálculo de λ_{j+1} e ainda obter uma expressão numericamente mais estável. Isso é possível, conforme estabelece o resultado a seguir.

Teorema 3.1. *Considere o problema (3.4) de encontrar a raiz da equação secular. Seja R_λ a matriz triangular calculada em (3.3), tal que $R_\lambda^T R_\lambda p_j = -J_c^T F_c$ e $R_\lambda^T q_j = p_j$ e seja Δ_c o raio da região de confiança corrente. A sequência $\{\lambda_j\}$ gerada pelo método de Newton definida em (3.5) é equivalente a:*

$$\lambda_{j+1} = \lambda_j + \left(\frac{\|p_j\|}{\|q_j\|} \right)^2 \left(\frac{\|p_j\| - \Delta_c}{\Delta_c} \right). \quad (3.6)$$

Demonstração. Ver [4, Lema 1]. \square

Por fim, para que o método de Newton não convirja para uma raiz indesejada de $\phi(\lambda)$ (pois estamos interessados em raízes positivas), usamos uma salvaguarda de modo que, a cada iteração, impomos $\lambda_j \in [\ell_j, u_j]$, sendo $\ell_0 = 0$ a estimativa para o menor autovalor de $J_c^T J_c + \lambda I$ e u_0 obtido pelo quociente de Rayleigh

$$(\mu_1 + \lambda)^2 \leq \frac{p^T (J_c^T J_c + \lambda I)^T (J_c^T J_c + \lambda I) p}{p^T p} \leq (\mu_n + \lambda)^2,$$

sendo μ_1, μ_n o menor e o maior autovalores de $J_c^T J_c$, respectivamente. Como zero é um limitante inferior para μ_1 , e estamos procurando $\|p\| = \Delta_c$, vem que

$$\lambda \leq \frac{\|J_c^T F_c\|}{\Delta_c} = u_0.$$

O intervalo da salvaguarda é atualizado a cada iteração da seguinte maneira:

$$\text{se } \delta(\lambda_j) \geq 0, \text{ então } \ell_{j+1} = \lambda_j + \frac{\|q_j\|^2}{\|p_j\|}, \text{ caso contrário } u_{j+1} = \lambda_j.$$

Para maiores detalhes, ver [4, Seção 4.2.1].

3.3. Quociente de redução relativa

Nesta subseção, nosso objetivo é eliminar a ocorrência de *underflows* e *overflows* no cálculo de (2.2). Usando LMA, podemos calcular o quociente como

$$\rho(p) = \frac{\|F_c\|^2 - \|F_n\|^2}{\|F_c\|^2 - \|F_c + J_c p\|^2}, \quad (3.7)$$

em que $F_n = F(x_c + p)$. De (2.3a), temos que

$$J_c^T J_c p = -J_c^T F_c - \lambda p \quad (3.8a)$$

$$-F_c^T J_c = \lambda p^T + p^T J_c^T J_c. \quad (3.8b)$$

Algumas manipulações com (3.8) em (3.7) implicam que

$$\|F_c\|^2 - \|F_c + J_c p\|^2 = \|J_c p\|^2 + 2\lambda \|p\|^2.$$

Portanto, podemos reescrever o quociente (3.7) dividindo todos os termos por $\|F_c\|^2$ e obtendo a seguinte relação:

$$\rho(p) = \frac{1 - \left(\frac{\|F_n\|}{\|F_c\|}\right)^2}{\left(\frac{\|J_c p\|}{\|F_c\|}\right)^2 + 2\lambda \left(\frac{\|p\|}{\|F_c\|}\right)^2}. \quad (3.9)$$

Deste modo, o denominador de (3.9) será sempre não-negativo e não irá gerar *overflows*. Por outro lado, o numerador de (3.9) jamais gerará *overflows* a não ser que $\|F_n\| \gg \|F_c\|$. Neste caso, temos que o modelo não foi reduzido com relação à função objetivo, portanto não é necessário calcular $\rho(p)$.

3.4. Atualização do raio da região de confiança

Utilizamos a aproximação inicial $\Delta_0 = \|J_0^T F_0\|/10$, clássica da literatura [2]. O controle do raio da região consiste em três instâncias: manter, aumentar ou reduzir. Os casos em que cada uma é adotada são expostos no Algoritmo 1. Para aumentar o tamanho do raio, fazemos $\Delta_n = 2\|p^*\|$. Para reduzir o tamanho do raio, fazemos $\Delta_n = \vartheta^* \Delta_c$. A fim de definir um valor para ϑ^* , seguimos Moré [9]: definimos $\gamma(\vartheta) \stackrel{\text{def}}{=} 1/2\|F(x_c + \vartheta p^*)\|^2$ e encontramos o minimizador ϑ^* da parábola interpolante

$$m_q(\vartheta) \stackrel{\text{def}}{=} [\gamma(1) - \gamma(0) - \gamma'(0)]\vartheta^2 + \gamma'(0)\vartheta + \gamma(0).$$

Como m_q é suave e convexa, então temos que $m'_q(\vartheta^*) = 0$, ou seja,

$$\vartheta^* = -\frac{\gamma'(0)}{2[\gamma(1) - \gamma(0) - \gamma'(0)]} = -\frac{(p^*)^T J_c^T F_c}{\|F_n\|^2 - \|F_c\|^2 - 2(p^*)^T J_c^T F_c}. \quad (3.10)$$

Pré-multiplicando (2.3a) avaliado em p^* por $(p^*)^T$, temos

$$-(p^*)^T (J_c^T J_c + \lambda^* I) p^* = (p^*)^T J_c^T F_c,$$

donde segue que

$$\frac{(p^*)^T J_c^T F_c}{\|F_c\|^2} = -\left[\left(\frac{\|J_c p^*\|}{\|F_c\|}\right)^2 + \lambda^* \left(\frac{\|p^*\|}{\|F_c\|}\right)^2 \right]. \quad (3.11)$$

Dividindo todos os termos de (3.10) por $\|F_c\|^2$, multiplicando o numerador e o denominador por $-1/2$ e substituindo (3.11), que denotamos por α , obtemos

$$\vartheta^* = \frac{\frac{1}{2}\alpha}{\alpha + \frac{1}{2} \left[1 - \left(\frac{\|F_n\|}{\|F_c\|}\right)^2 \right]}. \quad (3.12)$$

Pelos mesmos motivos expostos na seção anterior, (3.11) é uma expressão numericamente estável, e (3.12) é confiável e barata para encontrar ϑ^* e definir Δ_n .

4. Experimentos Numéricos

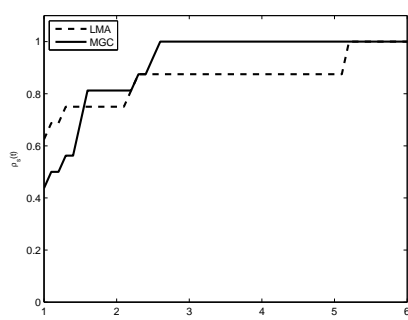
Ambos os métodos foram implementados no *CAS Maxima* [1, 15]. O *Maxima* é um sistema algébrico computacional livre de código aberto. Possui uma linguagem própria, interpretada, além de executar programas escritos em Lisp. Escolhemos o *Maxima* por ser um sistema gratuito e livre, além de não possuir ferramentas de otimização implementadas, como é o caso do Matlab[®]. Os testes foram rodados num computador Intel Dual-Core T3400 2.16 GHz e sistema operacional Windows 7 Professional 64 bits. Os problemas-teste foram extraídos de [10] e estão numerados de acordo com a referência original, bem como as dimensões e os pontos iniciais adotados (ver detalhes em [4, 5]).

Como critérios de parada, adotamos a satisfação relativa das condições de primeira ordem, isto é, x_k é solução em LMA se $|x_k^T J_k^T F_k / \max\{\|F_k\|, 1\}| \leq 10^{-6}$ e em MGC se $\|g_k\|/\|g_0\| \leq 10^{-6}$. Além disso, em LMA impusemos parada se o resíduo fosse pequeno, ou seja, $\|F_k\| \leq 10^{-6}$. Por fim, usamos critérios que implicam parada se a variação entre os iterandos fosse da ordem de 10^{-12} e se a quantidade de iterações ultrapassasse 200 para LMA e 300 para MGC, todavia nos testes efetuados nenhum problema parou por tais critérios.

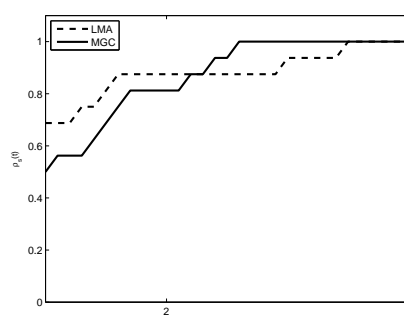
Os resultados dos experimentos são expostos na Tabela 1, enquanto os perfis de desempenho [3] na Figura 1. Neles, consideramos VF a função objetivo avaliada na solução, AH a quantidade de avaliações da matriz hessiana (para o MGC), AJ a quantidade de avaliações da matriz Jacobiana (para o LMA), ITE a quantidade de iterações externas, ITI a quantidade de iterações internas (isto é, para resolução dos subproblemas de regiões de confiança) e EI o esforço interno por problema, dado pela razão entre a quantidade de iterações internas e avaliações matriciais. A quantidade de avaliações da função objetivo é igual à quantidade de ITE acrescida de um. Os problemas marcados com * são aqueles que pararam pelo tamanho do resíduo em LMA; os demais pararam por satisfazer condições de primeira ordem.

As avaliações matriciais representam quantos dos pontos gerados foram aceitos. Deste modo, o número de pontos rejeitados, para cada problema, é dado pela diferença entre a quantidade de avaliações de funções e matriciais. Neste contexto, o esforço interno por problema representa quantas iterações internas foram despendidas, em média, para cada ponto aceito.

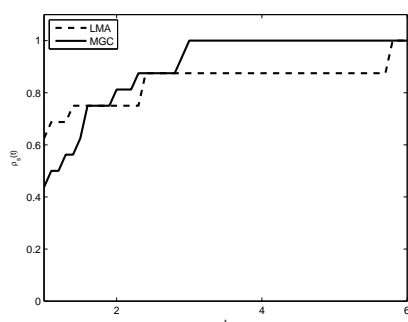
Acerca dos resultados apresentados, notamos que LMA é mais eficiente em avaliações de função, avaliações matriciais e quantidade de iterações internas, enquanto MGC, apenas em quantidade de iterações internas despendidas. Do ponto de vista de robustez, LMA se sobressai na quantidade de iterações internas, enquanto MGC em termos de avaliações, tanto de função quanto matriciais, e de iterações externas. Isto se deve ao fato de que LMA, por aproveitar melhor a estrutura do problema, trabalha mais internamente, consumindo menos iterações internas porém mais avaliações que MGC. Considerando-se, problema a problema, o número de iterações internas despendido para gerar a sequência de pontos aceitos, o esforço interno de LMA é inferior ao de MGC. Com relação ao número de pontos rejeitados, no entanto, o resultado se inverte: MGC rejeita menos pontos que LMA.



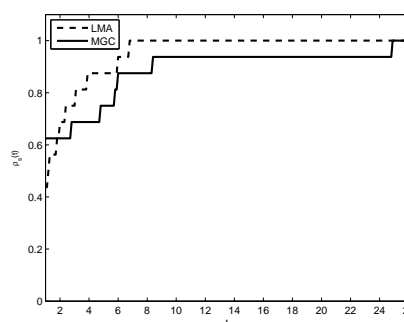
(a) Avaliações de Função.



(b) Avaliações Matriciais.



(c) Iterações Externas.



(d) Iterações Internas.

Figura 1: Perfis de desempenho dos algoritmos LMA e MGC.

Tabela 1: Quadro comparativo entre os resultados das rotinas LMA e MGC.

Função	MGC					LMA				
	VF	AH	ITE	ITI	EI	VF	AJ	ITE	ITI	EI
1*	0.34654 E-08	22	22	43	2.0	0.0	14	18	51	3.6
2	0.69988 E 01	8	7	14	1.8	0.69988 E 01	28	40	95	3.4
6	0.11151 E 02	9	8	16	1.8	0.11151 E 02	9	19	49	5.4
8	0.90635 E-01	14	14	42	3.0	0.90635 E-01	6	5	5	0.8
10	0.93779 E 01	199	216	826	4.2	0.93779 E 01	124	143	299	2.4
12*	0.92759 E-04	13	13	34	2.6	0.62754 E-10	12	12	34	2.8
13	0.75374 E-03	13	12	48	3.7	0.19361 E-03	9	8	8	0.9
15	0.17535 E-01	11	11	35	3.2	0.17536 E-01	11	12	43	3.9
16	0.29295 E 03	8	7	33	4.1	0.29295 E 03	24	40	127	5.3
17	0.73924 E-02	34	40	201	5.9	0.73924 E-02	16	18	35	2.2
19	0.20034 E 00	19	22	179	9.4	0.20034 E 00	14	14	38	2.7
20	0.22180 E-03	13	12	149	11.5	0.38213 E-04	5	4	6	1.2
27*	0.89312 E-05	7	6	13	1.9	0.19146 E-09	11	14	23	2.1
32	0.67082 E 01	4	3	3	0.8	0.67082 E 01	5	4	7	1.4
33	0.34826 E 01	2	2	1	0.5	0.34826 E 01	3	2	6	2.0
34	0.36917 E 01	2	2	1	0.5	0.36917 E 01	2	1	2	1.0
Média	-	23.6	24.8	102.4	3.5	-	18.3	22.1	51.8	2.6

5. Conclusão

O algoritmo LMA é uma proposta atraente para a resolução de (1.1), por ser um método mais barato, se comparado ao método de Newton. Em LMA, as informações de segunda ordem da função residual são substituídas pelo termo regularizador λI . Em contrapartida, o método perde desempenho em problemas extremamente não lineares ou se o resíduo for muito grande na solução x^* .

Em comparação com MGC, LMA aproveita a estrutura do problema (2.3a) e é mais eficiente em alguns quesitos. Todavia, exige muitas fatorações de matrizes, e trabalha mais internamente, o que pode ser ruim em alguns casos. Em vista disso, MGC é um método competitivo, mostrando-se mais robusto que LMA em parte dos quesitos analisados. Gradientes conjugados é bem conhecido por ser uma boa estratégia para resolução de sistemas lineares com matrizes definidas positivas, todavia aliado à estratégia de regiões de confiança pode ser uma alternativa interessante também para problemas não convexos.

Abstract. The least squares problem has many applications in the field of optimization. In the present work, we use two strategies for its resolution: Levenberg-Marquardt and Conjugate Gradients. Each one exploits some problem features, and both are globalized by the trust-region strategy. Our contribution consists in the implementation of both methods using the CAS Maxima and in the comparative analysis of these methods in the resolution of a family of least squares problems from the literature.

Keywords. Least squares, trust region, computational implementation.

Referências

- [1] L.N. Andrade, Maxima: um completo programa de computação algébrica, *Revista do Professor de Matemática*, n. 77, 2012.
- [2] A.R. Conn, N.I.L. Gould, Ph.L. Toint, “Trust-region Methods”. SIAM, Philadelphia, 2000.
- [3] E.D. Dolan, J.J. Moré, Benchmarking optimization software with performance profiles, *Mathematical Programming*, v. 91, pp. 201-213, 2002.
- [4] J.L.C. Gardenghi, S.A. Santos, “Sistemas não-lineares via região de confiança: o algoritmo de Levenberg-Marquardt”. Relatório de Pesquisa, 2011. Disponível em http://www.ime.unicamp.br/sites/default/files/rel_pesq/rp03-11.pdf. Acesso em 05 mar. 2013.
- [5] J.L.C. Gardenghi, S.A. Santos, “Minimização irrestrita usando gradientes conjugados e regiões de confiança”. Relatório de Pesquisa, 2012. Disponível em http://www.ime.unicamp.br/sites/default/files/rel_pesq/rp04-12.pdf. Acesso em 05 mar. 2013.
- [6] K. Levenberg, A method for the solution of certain non-linear problems in least squares, *The Quarterly of Applied Mathematics* 2, pp. 164-168, 1944.

-
- [7] K. Madsen, An algorithm for the minimax solution of overdetermined systems of nonlinear equations. *Journal of the Institute of Mathematics and its Applications*. v. 16(3), pp. 321-328, 1975.
 - [8] D.W. Marquardt, An algorithm for least-squares estimation of nonlinear parameters, *SIAM Journal on Applied Mathematics* 11, pp. 431-441, 1963.
 - [9] J.J. Moré, The Levenberg-Marquardt algorithm: implementation and theory. *Lecture Notes in Mathematics 630: Numerical Analysis*, Springer-Verlag, New York, pp. 105-116, 1978.
 - [10] J.J. Moré, B.S. Garbow, K.E. Hillstom, Testing unconstrained optimization software, *ACM Transactions on Mathematical Software*, v. 7, pp. 17-41, 1981.
 - [11] J. Nocedal, S.J. Wright, "Numerical Optimization". Springer, New York, 1999.
 - [12] T. Steihaug, The conjugate gradient method and trust region in large scale optimization. *SIAM Journal on Numerical Analysis*, v. 20(3), pp. 626-637, 1983.
 - [13] Ph.L. Toint, Towards an efficient sparsity exploiting Newton method for minimization, em "Sparse Matrix and Their Uses" (I. Duff, ed.), Academic Press, pp. 57-88, 1981.
 - [14] Y. Yuan, Recent advances in numerical methods for nonlinear equations and nonlinear least-squares. *Numerical Algebra, Control and Optimization*, v. 1, n. 1, pp. 15-34, 2011.
 - [15] <http://maxima.sourceforge.net>.