

MÉTODOS DE REGIÃO DE CONFIANÇA EM CONJUNTOS

ARBITRÁRIOS E MINIMIZAÇÃO EM BOLAS

Este exemplar corresponde a redação final da tese devidamente corrigida e defendida pela Sra. Sandra Augusta Santos e aprovada pela Comissão Julgadora.

Campinas, 28 de agosto de 1991

Prof. Dr. José Mario Martínez

Dissertação apresentada ao Instituto de Matemática, Estatística e Ciência da Computação, UNICAMP, como requisito parcial para obtenção do Título de Mestre em Matemática Aplicada.

Sa59m

14757/BC

UNICAMP
BIBLIOTECA CENTRAL

MÉTODOS DE REGIÃO DE CONFIANÇA
EM CONJUNTOS ARBITRÁRIOS
E MINIMIZAÇÃO EM BOLAS

SANDRA AUGUSTA SANTOS

ORIENTADOR: PROF. DR. JOSÉ MARIO MARTÍNEZ

DMA - IMECC - UNICAMP

CAMPINAS, AGOSTO DE 1991.

Alayde,

Elmer e

Lúcio :

é pra vocês !

AGRADECIMENTOS

Ao prof. Martínez,
pela excelente orientação.

Aos professores do IMECC,
pelo apoio e incentivo.

A FAPESP,
pelo custeio de meus estudos.

RESUMO

Neste trabalho, definimos dois algoritmos gerais de região de confiança para o problema de minimização restrita a um conjunto fechado arbitrário. Provamos convergência a pontos que satisfazem condições necessárias de primeira ordem e quando usamos a Hessiana da função objetivo no modelo, provamos que condições de segunda ordem são satisfeitas. Considerando-se a implementabilidade destes algoritmos, analisamos o caso em que a restrição é uma bola euclidiana. Desenvolvemos uma implementação computacional e fizemos um conjunto de experimentos numéricos.

ÍNDICE

CAPÍTULO 1.	INTRODUÇÃO	1
CAPÍTULO 2.	MÉTODOS DE REGIÃO DE CONFIANÇA PARA MINIMIZAÇÃO EM CONJUNTOS ARBITRÁRIOS	6
2.1	O PROBLEMA	6
2.2	OS ALGORITMOS I E II	8
2.3	DEFINIÇÕES, HIPÓTESES E LEMAS BÁSICOS	10
2.4	RESULTADOS DE CONVERGÊNCIA	12
CAPÍTULO 3.	MÉTODOS DE REGIÃO DE CONFIANÇA PARA MINIMIZAÇÃO EM BOLAS EUCLIDIANAS	38
3.1	O SUBPROBLEMA	38
3.2	O ALGORITMO III	40
3.3	ANÁLISE DO ALGORITMO III	41
3.3.1	MINIMIZADORES GLOBAIS DE QUADRÁTICAS EM ESFERAS	41
3.3.2	MINIMIZADORES LOCAIS DE QUADRÁTICAS EM ESFERAS	46
3.3.2.1	O ALGORITMO IV	47
3.3.3	MINIMIZADORES DE QUADRÁTICAS EM ESFERAS (N-2) DIMENSIONAIS	50
3.4	O "HARD-CASE"	55
3.4.1	O "HARD-CASE" NA ESFERA DE CONFIANÇA	57
3.4.2	O "HARD-CASE" NA ESFERA DO PROBLEMA	59
3.4.3	O "HARD-CASE" NA (N-2) ESFERA	61

CAPÍTULO 4.	EXPERIMENTOS NUMÉRICOS	63
4.1	CARACTERÍSTICAS DO PROGRAMA	63
4.2	ANÁLISE DO DESEMPENHO COMPUTACIONAL	65
4.2.1	TESTES REALIZADOS	65
4.2.2	RESULTADOS NUMÉRICOS	73
4.3	CONCLUSÕES E TRABALHOS FUTUROS	75
APÊNDICE		82
REFERÊNCIAS		142

CAPÍTULO 1

INTRODUÇÃO

Problemas das mais diversas áreas podem ser resolvidos via otimização não-linear. Para problemas de minimização restrita com função objetivo de várias variáveis, são necessárias técnicas apropriadas para se lidar com as restrições. Em última análise, no entanto, as técnicas escolhidas baseiam-se em soluções eficientes para o problema de minimização irrestrita. Uma opção clássica, com convergência local q -quadrática, é o método de Newton, cuja obtenção é natural quando se considera a aproximação quadrática para a função objetivo. Uma opção alternativa é utilizar variantes do método de Newton, obtidas por aproximações, em algum sentido, para a matriz Hessiana (ex: métodos quase Newton). Para pontos que não estejam suficientemente próximos da solução e nos quais a Hessiana (ou uma aproximação) não é positiva definida, ou quando existem infinitos minimizadores, o método não está bem definido. Surge assim a necessidade de modificar o método escolhido, tornando-o globalmente convergente. A estratégia que analisamos neste trabalho baseia-se em regiões de confiança. Uma outra opção globalizadora é utilizar buscas lineares, modificando-se a Hessiana quando esta não é "suficientemente positiva definida". Tal modificação para o modelo quadrático, apesar de assegurar a convergência, parece ignorar o papel que este modelo tem de ser uma aproximação local para a função

objetivo. Na estratégia de regiões de confiança, ao invés de se modificar o modelo quadrático, a idéia básica consiste em aceitar a direção que minimiza este modelo somente quando ele reflete adequadamente o comportamento da função objetivo original. Desta forma, o passo fica restrito pela região em que a série de Taylor para a função é confiável (região de confiança).

A formulação matemática mais comum desta idéia define $x_{k+1} = x_k + s_k$, onde s_k é a solução do subproblema :

$$\begin{array}{ll} \text{minimizar} & \xi_k^T s + \frac{1}{2} s^T B_k s \\ \text{s/a} & \| s \|_2 \leq \Delta_k \\ & s \in \mathbb{R}^n \end{array} \quad (1.1)$$

para certo $\Delta_k > 0$. Se f é a função objetivo, $\xi_k = \nabla f(x_k)$ e B_k é uma matriz simétrica (p.ex., $B_k = \nabla^2 f(x_k)$ ou $B_k \approx \nabla^2 f(x_k)$).

Pode-se mostrar que, se λ é um escalar tal que a matriz $B_k + \lambda I$ é semi positiva definida , a solução das equações

$$(B_k + \lambda I) s = - \xi_k \quad (1.2)$$

também é solução para o subproblema (1.1) se $\lambda = 0$ e $\| s \|_2 \leq \Delta_k$ ou $\lambda \geq 0$ e $\| s \|_2 = \Delta_k$. Assim, se Δ_k é suficientemente grande (e $B_k = \nabla^2 f(x_k) > 0$), a solução de (1.1) é simplesmente a direção de Newton (i.e., a solução de (1.2) com $\lambda = 0$). Caso contrário, a restrição sobre a norma torna-se ativa e $\| s \|_2 = \Delta_k$.

A direção s_k é encontrada resolvendo-se (1.1) com valores tentativos para Δ_k e avaliando-se a função objetivo nos pontos resultantes. Conforme apresenta [4], p. 114, para Δ_k suficientemente

pequeno, existe um vetor s tal que $f(x_k + s)$ é suficientemente menor que $f(x_k)$.

Em outras palavras, a idéia de uma iteração em um algoritmo de região de confiança para minimização irrestrita é:

Dado um ponto x_k e uma aproximação quadrática para a função objetivo em torno deste ponto:

1) Define-se uma bola centrada em x_k na qual acredita-se que a aproximação represente bem a função (região de confiança).

2) Estabelece-se um passo s_k de tal forma que haja decréscimo suficiente da aproximação na região de confiança.

3) Avalia-se a função objetivo em $x_k + s_k$. Se houve um decréscimo suficiente para o valor obtido, adota-se $x_{k+1} = x_k + s_k$ e aumenta-se o raio da região de confiança. Caso contrário, $x_k + s_k$ é rejeitado e o raio da região de confiança é diminuído.

Referências para o estudo dos métodos de região de confiança para problemas de minimização sem restrições, com ênfase na teoria de convergência, são: [1, 2, 4, 13, 14, 21, 22, 24].

Fazendo um breve histórico, os métodos de região de confiança têm como origem os trabalhos de Levenberg (1944) [8] e Marquardt (1963) [10] para problemas de quadrados mínimos não lineares. A primeira abordagem do método para o problema geral de minimização foi feita por Goldfeld, Quandt e Trotter em 1963 [5]. Powell (1970 e 1975) [18, 19] e Thomas (1975) [23] aplicaram a estratégia de convergência global com regiões de confiança a situações mais gerais

de iterações quase Newton. Reinsch (1971) [20] e Hebden (1973) [7] contribuíram com observações computacionais importantes para a implementação do método. Em 1978, Moré [12] voltou a trabalhar com o problema de quadrados mínimos não lineares. Fletcher (1980) [2] e Sorensen (1982) [22] provaram propriedades de convergência global para algoritmos de região de confiança usando informações de segunda ordem. Gay (1981) [3] acrescentou caracterizações teóricas e observações computacionais para o problema de minimização irrestrita quando a Hessiana é indefinida. Shultz, Schnabel e Byrd (1985) [21] apresentaram uma generalização para o trabalho de Sorensen [22]. Em 1988, Toint [24] provou propriedades de convergência global para uma classe de métodos de região de confiança para minimização não convexa em espaços de Hilbert.

No capítulo 2 deste trabalho, definimos dois algoritmos gerais de região de confiança para o problema de minimização restrita a um conjunto fechado "arbitrário". Provamos convergência a pontos que satisfazem condições necessárias de primeira ordem e quando usamos a Hessiana da função objetivo no modelo, provamos que condições de segunda ordem são satisfeitas . Tendo em vista a implementabilidade destes algoritmos, no capítulo 3 analisamos o caso em que a restrição do problema é uma bola euclidiana (ou, de maneira mais geral, um elipsóide sólido, que com uma mudança de variáveis adequada se reduz a uma bola euclidiana). Tal problema pode ser encarado como uma alternativa para a resolução de problemas irrestritos mal

condicionados (princípio de regularização). Independentemente deste argumento, a frequência do domínio elipsoidal na natureza bastaria para justificar a relevância deste problema. No capítulo 4 desenvolvemos um conjunto de experimentos numéricos para comprovar a robustez dos algoritmos propostos. No apêndice, apresentamos a documentação do programa e das rotinas implementadas.

CAPÍTULO 2

MÉTODOS DE REGIÃO DE CONFIANÇA PARA MINIMIZAÇÃO EM CONJUNTOS ARBITRÁRIOS

Neste capítulo, definiremos dois algoritmos gerais de região de confiança para o problema de minimização com restrições. Provamos convergência a pontos que satisfazem condições necessárias de primeira ordem e quando usamos a Hessiana da função objetivo no modelo, provamos que condições de segunda ordem são satisfeitas.

2.1 O PROBLEMA

Consideremos o problema:

$$\begin{array}{l} \text{minimizar } f(x) \\ \text{s/a } x \in B \end{array} \quad (2.1)$$

onde $f \in C^1(A)$, A é um conjunto aberto contendo B e B é fechado.

Para resolver (2.1) via métodos de região de confiança, definiremos dois algoritmos, que em cada iteração resolvem problemas do tipo:

$$\begin{array}{l} \text{minimizar } \psi_k(w) = w^t g_k + \frac{1}{2} w^t B_k w \\ \text{s/a } x_k + w \in B \\ \quad \|w\| \leq \Delta_k \end{array} \quad (2.2)$$

onde $g_k = \nabla f(x_k)$, x_k é a aproximação atual da solução e B_k é uma matriz simétrica.

Cabe observar que, como Ψ_k é contínua e o conjunto $\left\{ w \in \mathbb{R}^n \mid x_k + w \in B \text{ e } \|w\| \leq \Delta_k \right\}$ é compacto, a solução do problema (2.2) está bem definida.

Os dois algoritmos gerais apresentados a seguir são baseados no subproblema (2.2). O primeiro é a generalização direta de uma formulação clássica de algoritmos de região de confiança sem restrições [14]. O segundo é uma pequena variação do anterior para a qual é possível obter um teorema de convergência mais forte.

2.2 OS ALGORITMOS

ALGORITMO I

0) $k = 0$ $\Delta_0 < \infty$

1) dados x_k e Δ_k , calcular $g_k = \nabla f(x_k)$ e escolher B_k

2) resolver

$$\begin{aligned} \text{minimizar } \Psi_k(w) &= w^t g_k + \frac{1}{2} w^t B_k w \\ \text{s/a } x_k + w &\in B \\ \|w\| &\leq \Delta_k \end{aligned}$$

(2.2)

obtendo s_k . Se $s_k = 0$, parar.

3) se $f(x_k + s_k) < f(x_k) + 10^{-4} \Psi_k(s_k)$

então definir $x_{k+1} = x_k + s_k$

se $f(x_{k+1}) < f(x_k) + 0.75 \Psi_k(s_k)$

escolher $\Delta_{k+1} \geq \Delta_k$

se $f(x_{k+1}) \geq f(x_k) + 0.75 \Psi_k(s_k)$

escolher $\Delta_{k+1} \geq 0.1 \|s_k\|$

$k = k+1$, ir para (1)

se $f(x_k + s_k) \geq f(x_k) + 10^{-4} \Psi_k(s_k)$

escolher $\Delta_k \in [0.1 \|s_k\|, 0.9 \|s_k\|]$ e ir para (2)

Este algoritmo é uma generalização do algoritmo tradicional de região de confiança tal como descrito por Moré [14], no caso $B = \mathbb{R}^n$.

ALGORITMO II

Seja $0 < \Delta_{\min} < \infty$

0) $k = 0$ $\Delta_0 \geq \Delta_{\min}$

1) dados x_k e Δ_k , calcular $g_k = \nabla f(x_k)$ e escolher B_k

2) resolver

$$\text{minimizar } \Psi_k(w) = w^t g_k + \frac{1}{2} w^t B_k w$$

$$\text{s/a } x_k + w \in \mathbb{B}$$

$$\|w\| \leq \Delta_k$$

(2.2)

obtendo s_k . Se $s_k = 0$, parar.

3) se $f(x_k + s_k) < f(x_k) + 10^{-4} \Psi_k(s_k)$

então definir $x_{k+1} = x_k + s_k$

se $f(x_{k+1}) < f(x_k) + 0.75 \Psi_k(s_k)$

escolher $\Delta_{k+1} \geq \max \{\Delta_{\min}, \Delta_k\}$

se $f(x_{k+1}) \geq f(x_k) + 0.75 \Psi_k(s_k)$

escolher $\Delta_{k+1} \geq \max \{\Delta_{\min}, 0.1 \|s_k\|\}$

$k = k+1$, ir para (1)

se $f(x_k + s_k) \geq f(x_k) + 10^{-4} \Psi_k(s_k)$

escolher $\Delta_k \in [0.1 \|s_k\|, 0.9 \|s_k\|]$ e ir para (2)

O objetivo da modificação proposta no algoritmo II é que, ao começar cada iteração, a tentativa inicial seja suficientemente "arrojada", não se permitindo (inicialmente) tamanhos de região menores que $\Delta_{\min}$. Esta restrição, além de ser algoritmicamente razoável, tem influência nos resultados teóricos, como pode ser visto na seção 2.4 deste capítulo.

2.3 DEFINIÇÕES, HIPÓTESES E LEMAS BÁSICOS

Definição 2.1

Dados um conjunto B e um ponto factível $x^* \in B$, dizemos que a função $\alpha : [0,1] \rightarrow \mathbb{R}^n$, $\alpha \in C^2[0,1]$, é um arco factível partindo de x^* se $\alpha(0) = x^*$ e $\alpha(t) \in B$, $\forall t \in [0,1]$.

Definição 2.2

Dados um conjunto B e um ponto factível $x^* \in B$, dizemos que x^* é fracamente regular se para todo arco factível α partindo de x^* e para toda sequência $\{x_k\}_{k \in \mathbb{N}} \subset B$ que converge a x^* , existir uma sequência de arcos factíveis partindo de x_k $\{\alpha_k\}_{k \in \mathbb{N}}$ tal que:

- i) $\alpha_k(0) = x_k$, $\forall k \in \mathbb{N}$
- ii) $\alpha_k(t) \in B$, $\forall t \in [0,1]$, $\forall k \in \mathbb{N}$
- iii) $\lim_{k \in \mathbb{N}} \alpha_k(t) = \alpha(t)$, $\forall t \in [0,1]$
- iv) $\lim_{k \in \mathbb{N}} \alpha'_k(0) = \alpha'(0)$
- v) $\lim_{k \in \mathbb{N}} \alpha''_k(0) = \alpha''(0)$.

Hipótese 2.3

Os pontos de B são fracamente regulares.

Lema 2.4 - (CN1 - Condição Necessária de Primeira Ordem)

Se x^* é solução de (2.1),

então para todo arco factível α partindo de x^* tem-se

$$\left. \frac{df}{dt}(\alpha(t)) \right|_{t=0} = \nabla f^1(x^*) \frac{d\alpha}{dt}(0) \geq 0.$$

Prova: Ver [9], p.169. ■

Lema 2.5 - (CN2 - Condição Necessária de Segunda Ordem)

Se x^* é solução de (2.1), então para todo arco factível α partindo de x^* tem-se a CN1 satisfeita e sempre que

$$\left. \frac{df}{dt}(\alpha(t)) \right|_{t=0} = 0 \text{ tem-se } \left. \frac{d^2f}{dt^2}(\alpha(t)) \right|_{t=0} \geq 0.$$

Prova: Ver [9], p.174. ■

Lema 2.6

Se $s_k = 0$ é minimizador do subproblema (2.2)

então x_k satisfaz a CN1 para (2.1).

Prova:

Se s_k é minimizador de (2.2) então para todo arco factível ω partindo de s_k tem-se $\nabla \Psi_k^t(s_k) \frac{d\omega}{dt}(0) \geq 0$. Mas $\nabla \Psi_k(0) = \xi_k$ e sendo ω arco factível para (2.2), temos $\omega(0) = s_k = 0$ e existe $\delta > 0$ tal que $\omega(t) + x_k \in B$, $\|\omega(t)\| \leq \Delta_k$, $\forall t \in [0, \delta]$.

Chamando $\alpha(t) = \omega(t) + x_k$, temos $\omega(t) = \alpha(t) - x_k$, logo $\frac{d\omega}{dt}(t) = \frac{d\alpha}{dt}(t)$ e $\omega(0) = \alpha(0) - x_k = 0$. Então $\alpha(0) = x_k$ e portanto, a todo ω corresponde α , arco factível para (2.1) partindo de x_k , pois $\alpha(t) \in B$, $\forall t \in [0, \delta]$ com o mesmo δ dado acima. Além disso, $\nabla \Psi_k^t(0) \frac{d\omega}{dt}(0) = \xi_k^t \frac{d\alpha}{dt}(0) \geq 0$.

Assim, x_k satisfaz a CN1 para (2.1). ■

Lema 2.7

Se $s_k = 0$ é minimizador do subproblema (2.2), $f \in C^2(A)$ e

$B_k \equiv \nabla^2 f(x_k)$, $\forall k \in \mathbb{N}$, então x_k satisfaz a CN2 para (2.1).

Prova:

Se s_k é minimizador de (2.2) então para todo arco factível ω partindo de s_k tem-se $\nabla \Psi_k^t(s_k) \frac{d\omega}{dt}(0) \geq 0$ e para ω tal que $\nabla \Psi_k^t(s_k) \frac{d\omega}{dt}(0) = 0$ tem-se $\frac{d^2 \Psi}{dt^2}(s_k) \geq 0$. Sendo ω arco factível para (2.2), temos $\omega(0) = s_k = 0$ e existe $\delta > 0$ tal que $\omega(t) + x_k \in B$, $\|\omega(t)\| \leq \Delta_k$, $\forall t \in [0, \delta]$.

Chamando $\alpha(t) = \omega(t) + x_k$, temos $\omega(t) = \alpha(t) - x_k$, logo $\frac{d\omega}{dt}(t) = \frac{d\alpha}{dt}(t)$ e como $s_k = 0$, temos $\omega(0) = \alpha(0) - x_k = 0$ e então $\alpha(0) = x_k$. Além disso, $\nabla \Psi_k(0) = \xi_k$ e $\frac{d^2 \Psi}{dt^2}(0) = \nabla^2 f(x_k) = \frac{d^2 f}{dt^2}(\alpha(0))$, seguindo que a todo ω corresponde α , arco factível para (2.1) partindo de x_k , pois $\alpha(t) \in B$, $\forall t \in [0, \delta]$, com o mesmo δ dado acima. Desta forma, $\xi_k^t \frac{d\alpha}{dt}(0) \geq 0$ e para α tal que $\xi_k^t \frac{d\alpha}{dt}(0) = 0$ tem-se $\frac{d^2 f}{dt^2}(\alpha(0)) \geq 0$. Ou seja, x_k satisfaz a CN2 para (2.1) ■

2.4 RESULTADOS DE CONVERGÊNCIA

Teorema 2.8 (BOA DEFINIÇÃO - algoritmo I - CN1)

Seja $\{x_k\}_{k \in \mathbb{N}}$ a sequência gerada pelo algoritmo I.

Se x_k não satisfaz a CN1, então é possível obter x_{k+1} .

Em outras palavras, após um número finito de reduções no raio de confiança, ocorridas durante a k -ésima iteração, quando x_k não satisfaz a CN1, consegue-se definir um novo ponto x_{k+1} .

Prova:

Queremos ver que para $\Delta_k^{(i)}$ suficientemente pequeno (onde $\{\Delta_k^{(i)}\}_{i \in \mathbb{N}}$ é a sequência de raios gerada na k -ésima iteração), o

minimizador $s_k^{(i)}$ do subproblema (2.2) é tal que $f(x_k + s_k^{(i)}) < f(x_k) + 10^{-4} \Psi_k(s_k^{(i)})$, e portanto é possível obter o novo ponto $x_{k+1} = x_k + s_k^{(i)}$.

Chamando $\rho_k^{(i)} = \frac{f(x_k + s_k^{(i)}) - f(x_k)}{\Psi_k(s_k^{(i)})}$, vamos mostrar que

$$\lim_{i \in \mathbb{N}} \rho_k^{(i)} = 1.$$

Como por hipótese x_k não satisfaz a CN1, então existe o arco factível para (2.1) tal que $\nabla f^t(x_k) \frac{d\alpha}{dt}(0) < 0$. Daqui em diante usamos a notação: $\alpha' = \frac{d\alpha}{dt}$. Logo, $\xi_k^t \alpha'(0) < 0$ ($\alpha'(0) \neq 0$). Como α é arco factível, $\alpha(0) = x_k$ e $\alpha(t) \in \mathbb{B}$, $\forall t \in [0,1]$.

Para $\Delta_k^{(i)}$ suficientemente pequeno, seja t_i tal que $\|\alpha(t_i) - x_k\| = \Delta_k^{(i)}$.

Pela definição do subproblema, $\Psi_k(s_k^{(i)}) \leq \Psi_k(\alpha(t_i) - x_k)$, onde $t_i > 0$ é tal que $\|\alpha(t_i) - x_k\| = \Delta_k^{(i)}$. Mas,

$$\Psi_k(\alpha(t_i) - x_k) = \frac{1}{2} (\alpha(t_i) - x_k)^t B_k (\alpha(t_i) - x_k) + \xi_k^t (\alpha(t_i) - x_k).$$

$$\text{Agora } (\alpha(t_i) - x_k)^t B_k (\alpha(t_i) - x_k) \leq$$

$$\leq |(\alpha(t_i) - x_k)^t B_k (\alpha(t_i) - x_k)| \leq \|\alpha(t_i) - x_k\|^2 \|B_k\|.$$

$$\text{Logo, } \Psi_k(s_k^{(i)}) \leq \Psi_k(\alpha(t_i) - x_k) \leq \xi_k^t (\alpha(t_i) - x_k) + \frac{1}{2} \|B_k\| \|\alpha(t_i) - x_k\|^2.$$

Dividindo por t_i , obtemos:

$$\frac{\Psi_k(s_k^{(i)})}{t_i} \leq \frac{\xi_k^t (\alpha(t_i) - x_k)}{t_i} + \frac{\|B_k\|}{2} \frac{(\alpha(t_i) - x_k)^t (\alpha(t_i) - x_k)}{t_i}.$$

$$\text{Como } \lim_{t_i \rightarrow 0} \frac{\xi_k^t (\alpha(t_i) - x_k)}{t_i} = \xi_k^t \alpha'(0) \text{ e}$$

$$\lim_{t_i \rightarrow 0} \frac{(\alpha(t_i) - x_k)^t (\alpha(t_i) - x_k)}{t_i} = \alpha'(0)^t 0 = 0, \text{ segue que para } i$$

suficientemente grande, $\frac{\Psi_k(s_k^{(i)})}{t_i} \leq \frac{\epsilon_k^t \alpha'(0)}{2} = a < 0$.

Logo, $\left| \frac{\Psi_k(s_k^{(i)})}{t_i} \right| \geq |a|$ e então $\frac{1}{\left| \frac{\Psi_k(s_k^{(i)})}{t_i} \right|} \leq \frac{1}{|a|}$.

$$\begin{aligned} \text{Agora, } |\rho_k^{(i)} - 1| &= \left| \frac{f(x_k + s_k^{(i)}) - f(x_k) - \Psi_k(s_k^{(i)})}{\Psi_k(s_k^{(i)})} \right| \leq \\ &\leq \frac{|f(x_k + s_k^{(i)}) - f(x_k) - \epsilon_k^t s_k^{(i)} - \frac{1}{2} (s_k^{(i)})^t B_k s_k^{(i)}|}{|\Psi_k(s_k^{(i)})|} \leq \\ &\leq \frac{|f(x_k + s_k^{(i)}) - f(x_k) - \epsilon_k^t s_k^{(i)}| + \frac{1}{2} |(s_k^{(i)})^t B_k s_k^{(i)}|}{|\Psi_k(s_k^{(i)})|} \leq \\ &\leq \frac{L \|s_k^{(i)}\|^2 + \frac{\|B_k\|}{2} \|s_k^{(i)}\|^2}{|\Psi_k(s_k^{(i)})|} = \frac{\left(L + \frac{\|B_k\|}{2} \right) \frac{\|s_k^{(i)}\|^2}{t_i}}{\left| \frac{\Psi_k(s_k^{(i)})}{t_i} \right|} \leq \\ &\leq \frac{2L + \|B_k\|}{2|a|} \frac{\|s_k^{(i)}\|^2}{t_i} \leq \frac{2L + \|B_k\|}{2|a|} \frac{(\Delta_k^{(i)})^2}{t_i} = \\ &= \frac{2L + \|B_k\|}{2|a|} \frac{\|\alpha(t_i) - x_k\|^2}{t_i} \xrightarrow{t_i \rightarrow 0} 0. \end{aligned}$$

Portanto, $\lim_{i \in \mathbb{N}} \rho_k^{(i)} = 1$. Isto significa que para i

suficientemente grande, $\rho_k^{(i)} > 10^{-4}$.

Logo, $f(x_k + s_k^{(i)}) < f(x_k) + 10^{-4} \Psi_k(s_k^{(i)})$, como

queríamos provar . ■

Teorema 2.9 (CONVERGENCIA GLOBAL -algoritmo I - CN1)

Se $\|B_k\|$ é uniformemente limitada e a sequência gerada pelo algoritmo I está contida em um conjunto compacto, então existe uma subsequência cujo limite satisfaz a CN1 para (2.1).

Prova:

Existem duas possibilidades para $\left\{ \Delta_k \right\}_{k \in \mathbb{N}}$:

- i) $\inf_{k \in \mathbb{N}} \Delta_k = 0$
- ii) $\inf_{k \in \mathbb{N}} \Delta_k > 0$.

Consideremos inicialmente o caso (i): $\inf_{k \in \mathbb{N}} \Delta_k = 0$.

Seja $k_1 \in \mathbb{N}$ o primeiro número natural tal que $\Delta_{(k_1+1)} < \frac{\Delta_0}{2}$. Claramente $\Delta_{(k_1+1)} < \Delta_{k_1}$ (senão $\Delta_{(k_1-1)}$ seria o primeiro número).

Seja $k_2 \in \mathbb{N}$ o primeiro número tal que $\Delta_{(k_2+1)} < \frac{\Delta_{(k_1+1)}}{2}$ (claramente $\Delta_{(k_2+1)} < \Delta_{k_2}$).

E assim por diante, seja $k_j \in \mathbb{N}$ o primeiro número tal que $\Delta_{(k_j+1)} < \frac{\Delta_{(k_{j-1}+1)}}{2}$ (claramente $\Delta_{(k_j+1)} < \Delta_{k_j}$).

Seja $K_1 = \{k_1, k_2, \dots\}$. A sequência $\left\{ \Delta_k \right\}_{k \in K_1}$ é tal que

$\lim_{k \in K_1} \Delta_{k+1} = 0$. Como $\Delta_{k+1} < \Delta_k$, $\forall k \in K_1$, temos $f(x_k + s_k) \geq f(x_k) + 0.75 \Psi_k(s_k)$. Logo, $\rho_k = \frac{f(x_k + s_k) - f(x_k)}{\Psi_k(s_k)} \leq 0.75$, $\forall k \in K_1$.

Assim, para $k \in \mathbb{K}_1$ temos $\Delta_{k+1} \geq 0.1 \|s_k\|$ e portanto,

$$\lim_{k \in \mathbb{K}_1} \|s_k\| = 0.$$

Como $\{x_k\}_{k \in \mathbb{N}}$ está contida em um compacto, tomemos $\mathbb{K}_2 \subseteq \mathbb{K}_1$

tal que $\lim_{k \in \mathbb{K}_2} x_k$ existe e chamemos $x^* = \lim_{k \in \mathbb{K}_2} x_k$. Naturalmente,

$$\lim_{k \in \mathbb{K}_2} \|s_k\| = 0.$$

Suponhamos que x^* não satisfaça a CN1 para (2.1). Daqui em diante usaremos a notação $\xi_* = \nabla f(x^*)$.

Então existe α arco factível para (2.1) partindo de x^* , tal que $\xi_*^t \alpha'(0) < 0$. Pela regularidade de B (hipótese 2.3), existe a sequência de arcos factíveis $\{\alpha_k\}_{k \in \mathbb{K}_2}$ satisfazendo:

- i) $\alpha_k(0) = x_k, \forall k \in \mathbb{K}_2$
- ii) $\alpha_k(t) \in B, \forall t \in [0,1], \forall k \in \mathbb{K}_2$
- iii) $\lim_{k \in \mathbb{K}_2} \alpha_k(t) = \alpha(t), \forall t \in [0,1]$
- iv) $\lim_{k \in \mathbb{K}_2} \alpha'_k(0) = \alpha'(0).$

Agora, existe k_1 tal que para todo $k \in \mathbb{K}_2, k \geq k_1$, o arco α_k atravessa a bola de centro em x_k e raio $\|s_k\|$.

De fato, suponhamos que $\|\alpha_k(t) - \alpha_k(0)\| \leq \|s_k\|, \forall k \in \mathbb{K}_3 \subseteq \mathbb{K}_2$, para todo $t \in [0,1]$. Então $\lim_{k \in \mathbb{K}_3} \|\alpha_k(t) - \alpha_k(0)\| \leq \lim_{k \in \mathbb{K}_3} \|s_k\|, \forall t \in [0,1] \Rightarrow \|\alpha(t) - \alpha(0)\| \leq 0, \forall t \in [0,1] \Rightarrow \alpha(t) = \alpha(0), \forall t \in [0,1]$, o que é absurdo pois $\alpha'(0) \neq 0$.

Assim, existe k_1 tal que para $k \in \mathbb{K}_2, k \geq k_1$, podemos tomar

$$t_k = \min \left\{ t \in [0,1] : \|\alpha_k(t) - \alpha_k(0)\| = \|s_k\| \right\}.$$

Vejamos agora que $\lim_{k \in K_2} t_k = 0$.

$$\text{Temos } \|s_k\| = \|\alpha_k(t_k) - x_k\| = \|\alpha_k(t_k) - \alpha_k(0)\| = \left\| \begin{bmatrix} \int_0^{t_k} (\alpha_k^1)'(t) dt \\ \vdots \\ \int_0^{t_k} (\alpha_k^n)'(t) dt \end{bmatrix} \right\|.$$

Pela regularidade de B , como $\lim_{k \in K_2} \alpha_k'(0) = \alpha'(0) \neq 0$, existe

$i \in \{1, \dots, n\}$ tal que $(\alpha_k^i)'(t) \geq \eta > 0$ para todo $t \in [0, 1]$.

$$\text{Então, } \left\| \begin{bmatrix} \int_0^{t_k} (\alpha_k^1)'(t) dt \\ \vdots \\ \int_0^{t_k} (\alpha_k^n)'(t) dt \end{bmatrix} \right\| \geq \xi \left| \int_0^{t_k} (\alpha_k^i)'(t) dt \right| =$$

$$= \xi \int_0^{t_k} (\alpha_k^i)'(t) dt \geq \xi \int_0^{t_k} \eta dt = \xi \eta t_k, \text{ onde } \xi \text{ é uma constante não negativa.}$$

Portanto $\|s_k\| \geq \xi \eta t_k \geq 0$ e como $\lim_{k \in K_2} \|s_k\| = 0$, segue que

$$\lim_{k \in K_2} t_k = 0.$$

Pela definição do subproblema, $\Psi(s_k) \leq \Psi(\alpha_k(t_k) - x_k)$. Mas

$$\Psi(\alpha_k(t_k) - x_k) = g_k^l(\alpha_k(t_k) - x_k) + \frac{1}{2} (\alpha_k(t_k) - x_k)^l B_k(\alpha_k(t_k) - x_k).$$

Como $\|B_k\| \leq M$, $\forall k \in \mathbb{N}$, segue que

$$\Psi(s_k) \leq \Psi(\alpha_k(t_k) - x_k) \leq g_k^l(\alpha_k(t_k) - x_k) + \frac{M}{2} \|\alpha_k(t_k) - x_k\|^2 \text{ e}$$

$$\text{então } \frac{\Psi(s_k)}{t_k} \leq g_k^l \frac{[\alpha_k(t_k) - \alpha_k(0)]}{t_k} + \frac{M}{2t_k} \|\alpha_k(t_k) - \alpha_k(0)\|^2.$$

Aplicando o teorema do valor médio a cada componente de

$$\alpha_k(t_k) - \alpha_k(0) \text{ temos que } \frac{\Psi(s_k)}{t_k} \leq g_k^l v_k + M \|v_k\|^2 t_k, \text{ onde}$$

$$v_k = \begin{bmatrix} (\alpha_k^1)'(\xi_k^1) \\ \vdots \\ (\alpha_k^n)'(\xi_k^n) \end{bmatrix}, \quad 0 \leq \xi_k^i \leq tk, \quad i = 1, \dots, n.$$

$$\text{Portanto, } \lim_{k \in \mathbb{K}_2} \frac{\Psi_k(s_k)}{tk} \leq \varepsilon_*^l \alpha'(0) < 0$$

$$\text{Logo, } \frac{\Psi_k(s_k)}{tk} \leq \frac{\varepsilon_*^l \alpha'(0)}{2} = a < 0 \text{ para todo } k \in \mathbb{K}_2, k \geq k_2 \geq k_1.$$

$$\text{Então, } \left| \frac{\Psi_k(s_k)}{tk} \right| \geq |a| \text{ e } \frac{1}{\left| \frac{\Psi_k(s_k)}{tk} \right|} \leq \frac{1}{|a|}, \forall k \in \mathbb{K}_2, k \geq k_2.$$

Agora,

$$\rho_k = \frac{f(x_k + s_k) - f(x_k)}{\Psi_k(s_k)} = 1 + \frac{f(x_k + s_k) - f(x_k) - \Psi_k(s_k)}{\Psi_k(s_k)}$$

$$\text{Mas } |f(x_k + s_k) - f(x_k) - \Psi_k(s_k)| =$$

$$= |f(x_k + s_k) - f(x_k) - \varepsilon_k^l s_k - \frac{1}{2} s_k^l B_k s_k| \leq$$

$$\leq |f(x_k + s_k) - f(x_k) - \varepsilon_k^l s_k| + \frac{1}{2} |s_k^l B_k s_k| \leq L \|s_k\|^2 + \frac{1}{2} \|B_k\| \|s_k\|^2 \leq$$

$\leq (L + \frac{M}{2}) \|s_k\|^2$, onde L é a constante de Lipschitz para ∇f e usamos o fato de que $\|B_k\| \leq M, \forall k \in \mathbb{N}$.

Dessa forma, temos :

$$|f(x_k + s_k) - f(x_k) - \Psi_k(s_k)| \leq \left(\frac{2L + M}{2} \right) \|s_k\|^2.$$

Assim, para $k \in \mathbb{K}_2, k \geq k_2$ temos:

$$|\rho_k - 1| = \left| \frac{f(x_k + s_k) - f(x_k) - \Psi_k(s_k)}{\Psi_k(s_k)} \right| \leq \frac{\left(\frac{2L + M}{2} \right) \frac{\|s_k\|^2}{tk}}{\left| \frac{\Psi_k(s_k)}{tk} \right|} \leq$$

$$\leq \left(\frac{2L + M}{2|a|} \right) \frac{\|s_k\|^2}{t_k} = \frac{2L + M}{2|a|} \frac{\|a_k(t_k) - x_k\|^2}{t_k} =$$

$$= \frac{2L + M}{2|a|} \|v_k\|^2 t_k \xrightarrow{k \in \mathbb{K}_2} 0 \text{ e portanto } \lim_{k \in \mathbb{K}_2} \rho_k = 1,$$

o que contradiz o fato de que $\rho_k \leq 0.75$ para $\forall k \in \mathbb{K}_2 \subset \mathbb{K}_1$.

Consideremos agora o caso (ii): $\inf_{k \in \mathbb{N}} \Delta_k > 0$

Seja $\mathbb{K}_1 \subset \mathbb{N}$ tal que $\lim_{k \in \mathbb{K}_1} B_k$ existe. Chamemos $B = \lim_{k \in \mathbb{K}_1} B_k$.

Seja $\mathbb{K}_2 \subset \mathbb{K}_1$ tal que $\lim_{k \in \mathbb{K}_2} x_k$ existe. Chamemos $x^* = \lim_{k \in \mathbb{K}_2} x_k$.

Temos $f(x_{k+1}) < f(x_k) + 10^{-4} \Psi_k(s_k)$, $\forall k \in \mathbb{K}_2$.

Logo, $\Psi_k(s_k) > 10^4 (f(x_{k+1}) - f(x_k))$, $\forall k \in \mathbb{K}_2$

e $|\Psi_k(s_k)| < 10^4 (f(x_k) - f(x_{k+1}))$, $\forall k \in \mathbb{K}_2$

Agora, $f(x_k) - f(x_{k+1}) \longrightarrow 0$ (pois caso contrário, $f(x_k) \longrightarrow -\infty$), portanto $\lim_{k \in \mathbb{K}_2} \Psi_k(s_k) = 0$.

Vamos agora definir $\Psi_*(s) = s^t \xi_* + \frac{1}{2} s^t B s$.

Como $\inf_{k \in \mathbb{N}} \Delta_k > 0$, temos $\Delta_k \geq \Delta > 0$, $\forall k \in \mathbb{K}_2$.

Seja $\bar{s}$ solução do problema

$$\begin{aligned} &\text{minimizar } \Psi_*(s) \\ &s/a \quad x^* + s \in B \\ &\quad \quad \|s\| \leq \Delta/2 \end{aligned}$$

(2.3)

Existe $\bar{k}$ tal que para $k \geq \bar{k}$, $k \in \mathbb{K}_2$, temos $\|x^* - x_k\| \leq \Delta/2$

Definindo $\bar{x} = x^* + \bar{s}$, temos:

$$\|\bar{x} - x_k\| = \|x^* + \bar{s} - x_k\| \leq \|x^* - x_k\| + \|\bar{s}\| \leq \Delta \leq \Delta_k, \quad k \in \mathbb{K}_2, \quad k \geq \bar{k},$$

ou seja, $\|\bar{x} - x_k\| \leq \Delta_k$ e como $\bar{x} = x^* + \bar{s} \in B$, então $\bar{x} - x_k$ é factível para o subproblema (2.2) e $\Psi_k(\bar{x} - x_k) \geq \Psi_k(s_k)$.

Agora, no limite para $k \in \mathbb{K}_2$ temos $\xi_k \rightarrow \xi_*$, $B_k \rightarrow B$, $\Psi_k(s_k) \rightarrow 0$ e $\bar{x} - x_k \rightarrow \bar{s}$. Logo $\Psi_*(\bar{s}) \geq \Psi_*(0) = 0$ e $s = 0$ minimiza (2.3). Portanto, vale a CN1 para (2.3). Assim, para todo arco factível ω ($\omega(0) = 0$) tem-se $\nabla \Psi_*^t(0) \frac{d\omega}{dt}(0) \geq 0$. Chamando $\omega(t) = \alpha(t) - x^*$, como $\nabla \Psi_*(0) = \xi_*$ e $\frac{d\omega}{dt}(t) = \frac{d\alpha}{dt}(t)$, então para todo α arco factível para (2.1) ($\alpha(0) = x^*$) teremos $\xi_*^t \frac{d\alpha}{dt}(0) \geq 0$. Ou seja, vale a CN1 para (2.1). ■

Teorema 2.10 (BOA DEFINIÇÃO - algoritmo I - CN2)

Supondo que $f \in C^2(A)$, A aberto, $A \supset B$ e que $\nabla^2 f$ é Lipschitz contínua em A , seja $\{x_k\}_{k \in \mathbb{N}}$ a sequência gerada pelo algoritmo I, com $B_k \equiv \nabla^2 f(x_k)$, $\forall k \in \mathbb{N}$.

Se x_k não satisfaz a CN2, então é possível obter x_{k+1} .

Prova:

Se x_k satisfaz a CN1 mas não satisfaz a CN2, queremos ver que para $\Delta_k^{(i)}$ suficientemente pequeno (onde $\{\Delta_k^{(i)}\}_{i \in \mathbb{N}}$ é a sequência de raios gerada na k -ésima iteração), o minimizador $s_k^{(i)}$ do subproblema (2.2) é tal que $f(x_k + s_k^{(i)}) < f(x_k) + 10^{-4} \Psi_k(s_k^{(i)})$, e portanto é possível obter o novo ponto $x_{k+1} = x_k + s_k^{(i)}$.

Chamando $\rho_k^{(i)} = \frac{f(x_k + s_k^{(i)}) - f(x_k)}{\Psi_k(s_k^{(i)})}$, vamos mostrar que

$$\lim_{i \in \mathbb{N}} \rho_k^{(i)} = 1.$$

Como, por hipótese, x_k satisfaz a CN1 mas não satisfaz a CN2, existe arco factível α partindo de x_k tal que

$$\left. \frac{df}{dt}(\alpha(t)) \right|_{t=0} = g_k^t \alpha'(0) = 0 \quad \text{e} \quad \left. \frac{d^2 f}{dt^2}(\alpha(t)) \right|_{t=0} < 0.$$

$$\text{Logo, } \alpha'(0)^t \nabla^2 f_k \alpha'(0) + g_k^t \alpha''(0) < 0.$$

Sendo α factível, $\alpha(0) = x_k$ e $\alpha(t) \in B$, $\forall t \in [0,1]$.

Para $\Delta_k^{(i)}$ suficientemente pequeno, seja t_i tal que

$$t_i = \min \left\{ t \in [0,1] : \|\alpha(t) - x_k\| = \Delta_k^{(i)} \right\}.$$

Temos $\lim_{i \in \mathbb{N}} t_i = 0$. De fato, como $s_k^{(i-1)}$ é tal que

$$f(x_k + s_k^{(i-1)}) \geq f(x_k) + 10^{-4} \Psi_k(s_k^{(i-1)})$$

$$\text{então } \Delta_k^{(i)} \in [0.1 \|s_k^{(i-1)}\|, 0.9 \|s_k^{(i-1)}\|]$$

$$\text{e } \Delta_k^{(i)} \leq 0.9 \|s_k^{(i-1)}\| \leq 0.9 \Delta_k^{(i-1)}.$$

Portanto, $\lim_{i \in \mathbb{N}} \Delta_k^{(i)} = 0$ e como t_i é tal que $\|\alpha(t_i) - x_k\| = \Delta_k^{(i)}$,

$$\text{segue que } \lim_{i \in \mathbb{N}} \|\alpha(t_i) - x_k\| = \lim_{i \in \mathbb{N}} \|\alpha(t_i) - \alpha(0)\| = 0.$$

Ou seja, $\|\alpha(t_i) - \alpha(0)\| \xrightarrow{i \in \mathbb{N}} 0 \Rightarrow \alpha(t_i) \xrightarrow{i \in \mathbb{N}} \alpha(0) \Rightarrow t_i \xrightarrow{i \in \mathbb{N}} 0$.

Pela definição do subproblema,

$$\Psi_k(s_k^{(i)}) \leq \Psi_k(\alpha(t_i) - x_k) = \Psi_k(\alpha(t_i) - \alpha(0)).$$

Mas $\Psi_k(\alpha(t_i) - \alpha(0)) =$

$$= \frac{1}{2} (\alpha(t_i) - \alpha(0))^t B_k (\alpha(t_i) - \alpha(0)) + g_k^t (\alpha(t_i) - \alpha(0)).$$

Daqui em diante, utilizaremos a notação $\nabla^2 f_k = \nabla^2 f(x_k)$.

$$\frac{\Psi_k(s_k^{(i)})}{t_i^2} \leq \frac{\xi_k^t(\alpha(t_i) - \alpha(0))}{t_i^2} + \frac{1}{2} \frac{(\alpha(t_i) - \alpha(0))^t}{t_i} \nabla^2 f_k \frac{(\alpha(t_i) - \alpha(0))}{t_i}$$

Temos que $\lim_{t_i \rightarrow 0} \frac{\xi_k^t(\alpha(t_i) - \alpha(0))}{t_i^2} = \frac{1}{2} \xi_k^t \alpha''(0).$

De fato, chamando $F(t_i) = \frac{\xi_k^t(\alpha(t_i) - \alpha(0))}{t_i}$ e $G(t_i) = t_i$,

como $\lim_{i \in \mathbb{N}} F(t_i) = \xi_k^t \alpha'(0) = 0$ e $\lim_{i \in \mathbb{N}} G(t_i) = 0$, pelo teorema de

L'Hospital, segue que

$$\begin{aligned} \lim_{i \in \mathbb{N}} \frac{F(t_i)}{G(t_i)} &= \lim_{i \in \mathbb{N}} \frac{F'(t_i)}{G'(t_i)} = \lim_{i \in \mathbb{N}} \frac{\frac{d}{dt_i} \frac{\xi_k^t(\alpha(t_i) - \alpha(0))}{t_i}}{1} = \\ &= \lim_{i \in \mathbb{N}} \xi_k^t \frac{d}{dt_i} \left[\frac{(\alpha(t_i) - \alpha(0))}{t_i} \right] = \lim_{i \in \mathbb{N}} \xi_k^t \left[- \frac{(\alpha(t_i) - \alpha(0))}{t_i^2} + \frac{\alpha'(t_i)}{t_i} \right] = \\ &= - \lim_{i \in \mathbb{N}} \xi_k^t \left[\frac{(\alpha(t_i) - \alpha(0))}{t_i^2} \right] + \lim_{i \in \mathbb{N}} \xi_k^t \left[\frac{\alpha'(t_i) - \alpha'(0)}{t_i} \right] + \\ &\quad + \lim_{i \in \mathbb{N}} \xi_k^t \frac{\alpha'(0)}{t_i} \end{aligned}$$

Portanto, $2 \lim_{i \in \mathbb{N}} \xi_k^t \left[\frac{(\alpha(t_i) - \alpha(0))}{t_i^2} \right] = \xi_k^t \alpha''(0)$

e então, $\lim_{i \in \mathbb{N}} \xi_k^t \left[\frac{(\alpha(t_i) - \alpha(0))}{t_i^2} \right] = \frac{1}{2} \xi_k^t \alpha''(0).$

Agora,

$$\lim_{i \in \mathbb{N}} \frac{(\alpha(t_i) - \alpha(0))^t}{t_i} \nabla^2 f_k \frac{(\alpha(t_i) - \alpha(0))}{t_i} = \alpha'(0)^t \nabla^2 f_k \alpha'(0).$$

$$\text{Logo, } \lim_{i \in \mathbb{N}} \frac{\Psi_k(s_k^{(i)})}{t_i^2} \leq \frac{1}{2} \langle \xi_k^t \alpha''(0) + \alpha'(0)^t \nabla^2 f_k \alpha'(0) \rangle < 0$$

e para i suficientemente grande (t_i suficientemente pequeno)

$$\frac{\Psi_k(s_k^{(i)})}{t_i^2} \leq \frac{1}{4} \langle \xi_k^t \alpha''(0) + \alpha'(0)^t \nabla^2 f_k \alpha'(0) \rangle = a < 0.$$

$$\text{Então } \left| \frac{\Psi_k(s_k^{(i)})}{t_i^2} \right| \geq |a| \quad \text{e} \quad \frac{1}{\left| \frac{\Psi_k(s_k^{(i)})}{t_i^2} \right|} \leq \frac{1}{|a|}.$$

$$\begin{aligned} \text{Agora, } |\rho_k^{(i)} - 1| &= \left| \frac{f(x_k + s_k^{(i)}) - f(x_k) - \Psi_k(s_k^{(i)})}{\Psi_k(s_k^{(i)})} \right| \leq \\ &\leq \frac{\left| f(x_k + s_k^{(i)}) - f(x_k) - \xi_k^t s_k^{(i)} - \frac{1}{2} (s_k^{(i)})^t \nabla^2 f_k s_k^{(i)} \right|}{|\Psi_k(s_k^{(i)})|} \leq \\ &\leq \frac{\frac{M}{3!} \|s_k^{(i)}\|^3}{|\Psi_k(s_k^{(i)})|} \quad \left[\text{pela Lipschitz continuidade de } \nabla^2 f \text{ em } A \right]. \end{aligned}$$

$$\text{Mas } \frac{\frac{M}{3!} \|s_k^{(i)}\|^3}{|\Psi_k(s_k^{(i)})|} = \frac{\frac{M}{3!} \frac{\|s_k^{(i)}\|^3}{t_i^2}}{\left| \frac{\Psi_k(s_k^{(i)})}{t_i^2} \right|} \leq \frac{M}{3!|a|} \frac{\|s_k^{(i)}\|^3}{t_i^2} \leq$$

$$\leq \frac{M}{3!|a|} \frac{(\Delta_k^{(i)})^3}{t_i^2} = \frac{M}{3!|a|} \frac{\|\alpha(t_i) - x_k\|^3}{t_i^2} =$$

$$= \frac{M}{3!|a|} \left[\frac{(\alpha(t_i) - \alpha(0))^t}{t_i} \frac{(\alpha(t_i) - \alpha(0))^{3/2}}{t_i} \right] t_i \xrightarrow{i \in \mathbb{N}} 0$$

Logo $\lim_{i \in \mathbb{N}} \rho_k^{(i)} = 1$ e portanto para i suficientemente grande,

$\rho_k^{(i)} > 10^{-4}$. Assim, $f(x_k + s_k^{(i)}) < f(x_k) + 10^{-4} \Psi(s_k^{(i)})$. Ou seja,

$x_{k+1} = x_k + s_k^{(i)}$ está bem definido. ■

Teorema 2.11 (CONVERGENCIA GLOBAL - algoritmo I - CN2)

Supondo que $f \in C^2(A)$, A aberto, $A \supset B$ e que $\nabla^2 f$ é Lipschitz contínua em A , se a sequência gerada pelo algoritmo I está contida em um conjunto compacto e se $B_k \equiv \nabla^2 f(x_k)$, $\forall k \in \mathbb{N}$, então existe uma subsequência cujo limite satisfaz a CN2 para (2.1).

Prova:

Existem duas possibilidades para $\left\{ \Delta_k \right\}_{k \in \mathbb{N}}$:

i) $\inf_{k \in \mathbb{N}} \Delta_k = 0$

ii) $\inf_{k \in \mathbb{N}} \Delta_k > 0$.

Consideremos inicialmente o caso (i): $\inf_{k \in \mathbb{N}} \Delta_k = 0$.

Seja $k_1 \in \mathbb{N}$ o primeiro número natural tal que $\Delta_{(k_1+1)} < \frac{\Delta_0}{2}$. Claramente $\Delta_{(k_1+1)} < \Delta_{k_1}$ (senão $\Delta_{(k_1-1)}$ seria o primeiro número).

Seja $k_2 \in \mathbb{N}$ o primeiro número tal que $\Delta_{(k_2+1)} < \frac{\Delta_{(k_1+1)}}{2}$ (claramente $\Delta_{(k_2+1)} < \Delta_{k_2}$).

E assim por diante, seja $k_j \in \mathbb{N}$ o primeiro número tal que $\Delta_{(k_j+1)} < \frac{\Delta_{(k_{j-1}+1)}}{2}$ (claramente $\Delta_{(k_j+1)} < \Delta_{k_j}$).

Seja $K_1 = \{k_1, k_2, \dots\}$. A sequência $\left\{ \Delta_k \right\}_{k \in K_1}$ é tal que

$\lim_{k \in K_1} \Delta_{k+1} = 0$. Como $\Delta_{k+1} < \Delta_k$, $\forall k \in K_1$, temos

$$f(x_k + s_k) \geq f(x_k) + 0.75 \Psi_k(s_k). \text{ Logo } \rho_k = \frac{f(x_k + s_k) - f(x_k)}{\Psi_k(s_k)} \leq 0.75,$$

para todo $k \in K_1$.

Assim, para $k \in K_1$ temos $\Delta_{k+1} \geq 0.1 \|s_k\|$ e portanto,

$$\lim_{k \in K_1} \|s_k\| = 0.$$

Como $\{x_k\}_{k \in \mathbb{N}}$ está contida em um compacto, tomemos $K_2 \subseteq K_1$

tal que $\lim_{k \in K_2} x_k$ existe e chamemos $x^* = \lim_{k \in K_2} x_k$. Naturalmente,

$$\lim_{k \in K_2} \|s_k\| = 0.$$

Suponhamos que x^* não satisfaça a CN2 para (2.1). Usaremos a notação $g_* = \nabla f(x^*)$ e $\nabla^2 f_* = \nabla^2 f(x^*)$.

Então existe α arco factível para (2.1) partindo de x^* ($\alpha'(0) \neq 0$) tal que $\frac{df}{dt}(\alpha(0)) = g_*^t \alpha'(0) = 0$ mas $\frac{d^2 f}{dt^2}(\alpha(0)) < 0$, ou seja, $\alpha'(0)^t \nabla^2 f_* \alpha'(0) + g_*^t \alpha''(0) < 0$. Pela regularidade de $\mathbb{B}$, (hipótese 2.3) existe a sequência de arcos factíveis $\{\alpha_k(t)\}_{k \in K_2}$ satisfazendo:

- i) $\alpha_k(0) = x_k$, $\forall k \in K_2$
- ii) $\alpha_k(t) \in \mathbb{B}$, $\forall t \in [0,1], \forall k \in K_2$
- iii) $\lim_{k \in K_2} \alpha_k(t) = \alpha(t)$, $\forall t \in [0,1]$
- iv) $\lim_{k \in K_2} \alpha_k'(0) = \alpha'(0)$
- v) $\lim_{k \in K_2} \alpha_k''(0) = \alpha''(0)$.

Agora, existe k_1 tal que para $k \in K_2$, $k \geq k_1$, o arco α_k

atravessa a bola de centro em x_k e raio $\|s_k\|$.

De fato, suponhamos que $\|\alpha_k(t) - \alpha_k(0)\| \leq \|s_k\|$, $\forall t \in [0,1]$, $\forall k \in K_1 \subset K_2$. Então $\lim_{k \in K_1} \|\alpha_k(t) - \alpha_k(0)\| \leq \lim_{k \in K_1} \|s_k\|$, $\forall t \in [0,1]$. Como $K_1 \subset K_2$, temos $0 \leq \|\alpha(t) - \alpha(0)\| \leq 0$, $\forall t \in [0,1]$ e então $\alpha(t) = \alpha(0)$, $\forall t \in [0,1]$, o que é absurdo pois $\alpha'(0) \neq 0$.

Assim, para $k \in K_2$, $k \geq k_1$, podemos tomar

$$t_k = \min \left\{ t \in [0,1] : \|\alpha_k(t) - x_k\| = \|s_k\| \right\}$$

Vejam agora que $\lim_{k \in K_2} t_k = 0$.

$$\text{Temos } \|s_k\| = \|\alpha_k(t_k) - x_k\| = \|\alpha_k(t_k) - \alpha_k(0)\| = \left\| \begin{bmatrix} \int_0^{t_k} (\alpha_k^1)'(t) dt \\ \vdots \\ \int_0^{t_k} (\alpha_k^n)'(t) dt \end{bmatrix} \right\|.$$

Pela regularidade de B , como $\lim_{k \in K_2} \alpha_k'(0) = \alpha'(0) \neq 0$, existe

$i \in \{1, \dots, n\}$ tal que $(\alpha_k^i)'(t) \geq \eta > 0$ para todo $t \in [0,1]$.

$$\begin{aligned} \text{Então, } \left\| \begin{bmatrix} \int_0^{t_k} (\alpha_k^1)'(t) dt \\ \vdots \\ \int_0^{t_k} (\alpha_k^n)'(t) dt \end{bmatrix} \right\| &\geq \xi \left| \int_0^{t_k} (\alpha_k^i)'(t) dt \right| = \\ &= \xi \int_0^{t_k} (\alpha_k^i)'(t) dt \geq \xi \int_0^{t_k} \eta dt = \xi \eta t_k, \text{ onde } \xi \text{ é uma constante não} \\ &\text{negativa.} \end{aligned}$$

Portanto $\|s_k\| \geq \xi \eta t_k \geq 0$ e como $\lim_{k \in K_2} \|s_k\| = 0$, segue que

$$\lim_{k \in K_2} t_k = 0.$$

Pela definição do subproblema, $\Psi_k(s_k) \leq \Psi_k(\alpha_k(t_k) - x_k)$. Mas

$$\Psi_k(\alpha_k(t_k) - x_k) = \xi_k^t(\alpha_k(t_k) - x_k) + \frac{1}{2} (\alpha_k(t_k) - x_k)^t \nabla^2 f_k(\alpha_k(t_k) - x_k).$$

$$\begin{aligned} \text{Então } \frac{\Psi_k(s_k)}{(t_k)^2} &\leq \frac{\xi_k^t(\alpha_k(t_k) - \alpha_k(0))}{(t_k)^2} + \\ &+ \frac{1}{2} \frac{(\alpha_k(t_k) - \alpha_k(0))^t}{t_k} \nabla^2 f_k \frac{(\alpha_k(t_k) - \alpha_k(0))}{t_k}. \end{aligned}$$

$$\text{Chamando } F(t_k) = \frac{\xi_k^t(\alpha_k(t_k) - \alpha_k(0))}{t_k} \text{ e } G(t_k) = t_k, \text{ como}$$

$$\lim_{k \in K_2} F(t_k) = \xi_*^t \alpha'(0) = 0 \text{ e } \lim_{k \in K_2} G(t_k) = 0, \text{ aplicando o teorema de}$$

L'Hospital temos :

$$\begin{aligned} \lim_{k \in K_2} \frac{\xi_k^t(\alpha_k(t_k) - \alpha_k(0))}{(t_k)^2} &= \lim_{k \in K_2} \frac{F(t_k)}{G(t_k)} = \lim_{k \in K_2} \frac{\frac{d}{dt_k} F(t_k)}{\frac{d}{dt_k} G(t_k)} = \\ &= \lim_{k \in K_2} \xi_k^t \left[- \frac{(\alpha_k(t_k) - \alpha_k(0))}{(t_k)^2} + \frac{\alpha'_k(t_k)}{t_k} \right] = \\ &= - \lim_{k \in K_2} \xi_k^t \left[\frac{(\alpha_k(t_k) - \alpha_k(0))}{(t_k)^2} \right] + \lim_{k \in K_2} \xi_k^t \left[\frac{\alpha'_k(t_k) - \alpha'_k(0)}{t_k} \right] + \\ &\quad + \lim_{k \in K_2} \xi_k^t \frac{\alpha'_k(0)}{t_k}. \end{aligned}$$

$$\text{Portanto } 2 \lim_{k \in K_2} \frac{\xi_k^t(\alpha_k(t_k) - \alpha_k(0))}{(t_k)^2} = \xi_*^t \alpha''(0) + \lim_{k \in K_2} \xi_k^t \frac{\alpha'_k(0)}{t_k}.$$

Aplicando novamente o teorema de L'Hospital segue que

$$\lim_{k \in K_2} \xi_k^t \frac{\alpha'_k(0)}{t_k} = 0$$

$$\text{e portanto } \lim_{k \in \mathbb{K}_2} \frac{\xi_k^t (\alpha_k(t_k) - \alpha_k(0))}{(tk)^2} = \frac{1}{2} \xi_*^t \alpha''(0).$$

Agora,

$$\lim_{k \in \mathbb{K}_2} \frac{(\alpha_k(t_k) - \alpha_k(0))^t}{tk} \nabla^2 f_k \frac{(\alpha_k(t_k) - \alpha_k(0))}{tk} = \alpha'(0)^t \nabla^2 f_* \alpha'(0).$$

$$\text{Então } \lim_{k \in \mathbb{K}_2} \frac{\Psi_k(s_k)}{(tk)^2} \leq \frac{1}{2} (\xi_*^t \alpha''(0) + \alpha'(0)^t \nabla^2 f_* \alpha'(0)) < 0.$$

Portanto, para $k \geq k_2 \geq k_1$, $k \in \mathbb{K}_2$ temos

$$\frac{\Psi_k(s_k)}{(tk)^2} \leq \frac{1}{4} (\xi_*^t \alpha''(0) + \alpha'(0)^t \nabla^2 f_* \alpha'(0)) = a < 0$$

$$\text{Logo } \left| \frac{\Psi_k(s_k)}{(tk)^2} \right| \geq |a| \text{ e } \frac{1}{\left| \frac{\Psi_k(s_k)}{(tk)^2} \right|} \leq \frac{1}{|a|}, \quad k \in \mathbb{K}_2, k \geq k_2.$$

Agora,

$$\rho_k = \frac{f(x_k + s_k) - f(x_k)}{\Psi_k(s_k)} = 1 + \frac{f(x_k + s_k) - f(x_k) - \Psi_k(s_k)}{\Psi_k(s_k)}$$

$$\text{Mas } |f(x_k + s_k) - f(x_k) - \Psi_k(s_k)| =$$

$$= \left| f(x_k + s_k) - f(x_k) - \xi_k^t s_k - \frac{1}{2} s_k^t \nabla^2 f_k s_k \right| \leq \frac{M}{3!} \|s_k\|^3$$

(pela Lipschitz continuidade de $\nabla^2 f$)

Assim, para $k \in \mathbb{K}_2$, $k \geq k_2$ temos

$$|\rho_k - 1| = \left| \frac{f(x_k + s_k) - f(x_k) - \Psi_k(s_k)}{\Psi_k(s_k)} \right| \leq \frac{\frac{M}{3!} \frac{\|s_k\|^3}{(tk)^2}}{\left| \frac{\Psi_k(s_k)}{(tk)^2} \right|} \leq$$

$$\leq \frac{M}{3!|a|} \frac{\|s_k\|^3}{(tk)^2} = \frac{M}{3!|a|} \frac{\|\alpha_k(tk) - x_k\|^3}{(tk)^2} = \frac{M}{3!|a|} \|v_k\|^3 tk,$$

onde v_k é obtido pelo teorema do valor médio aplicado a cada componente de $\alpha_k(tk) - x_k$,

$$v_k = \begin{bmatrix} (\alpha_k^1)'(\xi_k^1) \\ \vdots \\ (\alpha_k^n)'(\xi_k^n) \end{bmatrix}, \quad 0 \leq \xi_k^i \leq tk, \quad 1 \leq i \leq n.$$

Então

$$\lim_{k \in K_2} |\rho_k - 1| \leq \lim_{k \in K_2} \frac{M}{3!|a|} \|v_k\|^3 tk = \frac{M}{3!|a|} \|\alpha'(0)\|^3 \lim_{k \in K_2} tk = 0.$$

Logo $\lim_{k \in K_2} \rho_k = 1$, o que contradiz o fato de que $\rho_k \leq 0.75$,

$\forall k \in K_2 \subset K_1$.

Consideremos agora o caso (ii): $\inf_{k \in \mathbb{N}} \Delta_k > 0$.

Seja $K_1 \subset \mathbb{N}$ tal que $\lim_{k \in K_1} \nabla^2 f_k$ existe. Chamemos $\nabla^2 f =$

$$= \lim_{k \in K_1} \nabla^2 f_k.$$

Seja $K_2 \subset K_1$ tal que $\lim_{k \in K_2} x_k$ existe. Chamemos $x^* = \lim_{k \in K_2} x_k$.

$$\text{Temos } f(x_{k+1}) < f(x_k) + 10^{-4} \Psi(s_k), \quad \forall k \in K_2.$$

$$\text{Então } \Psi(s_k) > 10^4 (f(x_{k+1}) - f(x_k)), \quad \forall k \in K_2$$

$$\text{e } |\Psi(s_k)| < 10^4 (f(x_k) - f(x_{k+1})), \quad \forall k \in K_2.$$

Agora, $f(x_k) - f(x_{k+1}) \longrightarrow 0$ (pois caso contrário,

$f(x_k) \longrightarrow -\infty$), portanto $\lim_{k \in K_2} \Psi(s_k) = 0$.

Vamos agora definir $\Psi_*(s) = s^T \xi_* + \frac{1}{2} s^T \nabla^2 f s$.

Como $\inf_{k \in \mathbb{N}} \Delta_k > 0$, temos $\Delta_k \geq \Delta > 0, \forall k \in \mathbb{K}_2$.

Seja $\bar{s}$ solução do problema

$$\begin{array}{l} \text{minimizar } \Psi_*(s) \\ \text{s/a } x^* + s \in B \\ \|s\| \leq \Delta/2 \end{array} \quad (2.4)$$

Existe $\bar{k}$ tal que para $k \geq \bar{k}, k \in \mathbb{K}_2$, temos $\|x^* - x_k\| \leq \Delta/2$.

Definindo $\bar{x} = x^* + \bar{s}$, temos:

$\|\bar{x} - x_k\| = \|x^* + \bar{s} - x_k\| \leq \|x^* - x_k\| + \|\bar{s}\| \leq \Delta \leq \Delta_k, k \in \mathbb{K}_2, k \geq \bar{k}$,
ou seja, $\|\bar{x} - x_k\| \leq \Delta_k$ e como $\bar{x} = x^* + \bar{s} \in B$ segue que $\bar{x} - x_k$ é factível para o subproblema (2.2) e então $\Psi_k(\bar{x} - x_k) \geq \Psi_k(s_k)$.

Agora, no limite para $k \in \mathbb{K}_2, \xi_k \rightarrow \xi_*, \nabla^2 f_k \rightarrow \nabla^2 f$,
 $\Psi_k(s_k) \rightarrow 0$ e $\bar{x} - x_k \rightarrow \bar{s}$. Logo $\Psi_*(\bar{s}) \geq \Psi_*(0) = 0$ e $s = 0$ minimiza (2.4). Portanto vale a CN2 para (2.4). Isto é, para todo arco factível ω ($\omega(0) = 0$) tem-se $\nabla \Psi_*^T(0) \frac{d\omega}{dt}(0) \geq 0$ e para todo ω tal que $\nabla \Psi_*^T(0) \frac{d\omega}{dt}(0) = 0$ tem-se $\frac{d^2 \Psi_*}{dt^2}(0) \geq 0$.

Chamando $\omega(t) = \alpha(t) - x^*$, como $\nabla \Psi_*(0) = \xi_*, \frac{d\omega}{dt}(t) = \frac{d\alpha}{dt}(t)$ e $\frac{d^2 \Psi_*}{dt^2}(0) = \nabla^2 f = \frac{d^2 f}{dt^2}(\alpha(0))$, então para todo α arco factível para (2.1) ($\alpha(0) = x^*$) teremos $\xi_*^T \frac{d\alpha}{dt}(0) \geq 0$ e para α tal que $\xi_*^T \frac{d\alpha}{dt}(0) = 0$ tem-se $\frac{d^2 f}{dt^2}(\alpha(0)) \geq 0$. Ou seja, vale a CN2 para (2.1). ■

Teorema 2.12 (BOA DEFINIÇÃO - algoritmo II - CN1)

Seja $\left\{ x_k \right\}_{k \in \mathbb{N}}$ a sequência gerada pelo algoritmo II.

Se x_k não satisfaz a CN1, então é possível obter x_{k+1} .

Prova:

Como os subproblemas dos algoritmos I e II são os mesmos, a prova deste teorema é análoga à do teorema 2.8. ■

Teorema 2.13 (CONVERGÊNCIA GLOBAL - algoritmo II - CN1)

Seja $\{x_k\}_{k \in \mathbb{N}}$ a sequência gerada pelo algoritmo II.

Se $\|B_k\|$ é uniformemente limitada e $\lim_{k \in K} x_k = x^*$, $K \subseteq \mathbb{N}$
então x^* satisfaz a CN1 para (2.1).

Prova:

Seja $x^* = \lim_{k \in K} x_k$, $K \subseteq \mathbb{N}$.

Existem duas possibilidades para $\{\Delta_k\}_{k \in K}$:

- i) $\inf_{k \in K} \Delta_k = 0$
- ii) $\inf_{k \in K} \Delta_k > 0$.

Consideremos inicialmente o caso (i):

Se $\inf_{k \in K} \Delta_k = 0$ então existe $K_1 \subseteq K$ tal que $\lim_{k \in K_1} \Delta_k = 0$.

Mas, como $\Delta_{k+1} \geq \Delta_{\min} > 0$, $\forall k \in \mathbb{N}$, segue que Δ_k só pode decrescer, indo para zero, se considerarmos a diminuição dos raios de confiança ocorrida durante as iterações da subsequência infinita K_1 . Assim, em cada iteração de K_1 , existe um fracasso na condição de Armijo antes de se mudar de ponto. Ou seja, para todo $k \in K_1$, existe $\bar{s}_k$ tal que $f(x_k + \bar{s}_k) \geq f(x_k) + 10^{-4} \Psi_k(\bar{s}_k)$ e para o subproblema com raio de confiança $\Delta_k \in \left[0.1 \|\bar{s}_k\|, 0.9 \|\bar{s}_k\| \right]$, obtem-se s_k que

satisfaz Armijo, definindo-se $x_{k+1} = x_k + s_k$.

Portanto, $\Delta_k \geq 0.1 \|\bar{s}_k\| \geq 0$, $\forall k \in K_1$, seguindo que

$\lim_{k \in K_1} \|\bar{s}_k\| = 0$. Naturalmente, $\lim_{k \in K_1} x_k = x^*$.

Suponhamos que x^* não satisfaça a CN1 para (2.1).

Então existe α arco factível para (2.1) ($\alpha(0) = x^*$, $\alpha'(0) \neq 0$ e $\alpha(t) \in B$, $\forall t \in [0,1]$) tal que $\xi_*^t \frac{d\alpha}{dt}(0) < 0$. Pela regularidade de B , existe a sequência de arcos factíveis $\{\alpha_k\}_{k \in K_1}$ satisfazendo:

- i) $\alpha_k(0) = x_k$, $\forall k \in K_1$
- ii) $\alpha_k(t) \in B$, $\forall t \in [0,1]$, $\forall k \in K_1$
- iii) $\lim_{k \in K_1} \alpha_k(t) = \alpha(t)$, $\forall t \in [0,1]$
- iv) $\lim_{k \in K_1} \alpha'_k(0) = \alpha'(0)$.

Agora, existe k_1 tal que para todo $k \in K_1$, $k \geq k_1$, o arco α_k atravessa a bola de centro em x_k e raio $\|\bar{s}_k\|$.

De fato, suponhamos que $\|\alpha_k(t) - \alpha_k(0)\| \leq \|\bar{s}_k\|$, $\forall k \in K_2 \subset K_1$, $\forall t \in [0,1]$. Então $\lim_{k \in K_2} \|\alpha_k(t) - \alpha_k(0)\| \leq \lim_{k \in K_2} \|\bar{s}_k\|$, $\forall t \in [0,1] \Rightarrow \|\alpha(t) - \alpha(0)\| \leq 0$, $\forall t \in [0,1]$.

Logo $\alpha(t) = \alpha(0)$, $\forall t \in [0,1]$, o que é absurdo pois $\alpha'(0) \neq 0$.

Assim, existe k_1 tal que para $k \in K_1$, $k \geq k_1$, podemos tomar

$$t_k = \min \left\{ t \in [0,1] : \|\alpha_k(t) - \alpha_k(0)\| = \|\bar{s}_k\| \right\}.$$

Vejamos agora que $\lim_{k \in K_1} t_k = 0$.

$$\text{Temos } \|s_k\| = \|\alpha_k(tk) - x_k\| = \|\alpha_k(tk) - \alpha_k(0)\| = \left\| \begin{bmatrix} \int_0^{tk} (\alpha_k^1)'(t) dt \\ \vdots \\ \int_0^{tk} (\alpha_k^n)'(t) dt \end{bmatrix} \right\|.$$

Pela regularidade de B , como $\lim_{k \in K_1} \alpha_k'(0) = \alpha'(0) \neq 0$, existe

$i \in \{1, \dots, n\}$ tal que $(\alpha_k^i)'(t) \geq \eta > 0$ para todo $t \in [0, 1]$.

$$\text{Então, } \left\| \begin{bmatrix} \int_0^{tk} (\alpha_k^1)'(t) dt \\ \vdots \\ \int_0^{tk} (\alpha_k^n)'(t) dt \end{bmatrix} \right\| \geq \xi \left| \int_0^{tk} (\alpha_k^i)'(t) dt \right| =$$

$$= \xi \int_0^{tk} (\alpha_k^i)'(t) dt \geq \xi \int_0^{tk} \eta dt = \xi \eta tk, \text{ onde } \xi \text{ é uma constante não negativa.}$$

Portanto $\|\bar{s}_k\| \geq \xi \eta tk \geq 0$ e como $\lim_{k \in K_1} \|\bar{s}_k\| = 0$, segue que

$$\lim_{k \in K_1} tk = 0.$$

Pela definição do subproblema, $\Psi_k(\bar{s}_k) \leq \Psi_k(\alpha_k(tk) - x_k)$.

Mas

$$\Psi_k(\alpha_k(tk) - x_k) = g_k^l(\alpha_k(tk) - x_k) + \frac{1}{2} (\alpha_k(tk) - x_k)^t B_k(\alpha_k(tk) - x_k).$$

Como $\|B_k\| \leq M$, $\forall k \in \mathbb{N}$, segue que

$$\Psi_k(\bar{s}_k) \leq \Psi_k(\alpha_k(tk) - x_k) \leq g_k^l(\alpha_k(tk) - x_k) + \frac{M}{2} \|\alpha_k(tk) - x_k\|^2.$$

$$\text{Logo } \frac{\Psi_k(\bar{s}_k)}{tk} \leq g_k^l \frac{[\alpha_k(tk) - \alpha_k(0)]}{tk} + \frac{M}{2tk} \|\alpha_k(tk) - \alpha_k(0)\|^2.$$

Aplicando o teorema do valor médio em cada componente de $\alpha_k(tk) - \alpha_k(0)$ temos que:

$$\frac{\Psi_k(\bar{s}_k)}{t_k} \leq \xi_k^t v_k + M \|v_k\|^2 t_k, \quad v_k = \begin{bmatrix} (\alpha_k^1)'(\xi_k^1) \\ \vdots \\ (\alpha_k^n)'(\xi_k^n) \end{bmatrix},$$

$$0 \leq \xi_k^i \leq t_k, \quad i = 1, \dots, n.$$

$$\text{Logo } \lim_{k \in K_2} \frac{\Psi_k(\bar{s}_k)}{t_k} \leq \xi_*^t \alpha'(0) < 0.$$

$$\text{Então } \frac{\Psi_k(\bar{s}_k)}{t_k} \leq \frac{\xi_*^t \alpha'(0)}{2} = \alpha < 0 \text{ para } \forall k \in K_1, k \geq k_2 \geq k_1$$

$$\text{e } \left| \frac{\Psi_k(\bar{s}_k)}{t_k} \right| \geq |\alpha| \Rightarrow \frac{1}{\left| \frac{\Psi_k(\bar{s}_k)}{t_k} \right|} \leq \frac{1}{|\alpha|}, \forall k \in K_1, k \geq k_2.$$

Agora,

$$\bar{\rho}_k = \frac{f(x_k + \bar{s}_k) - f(x_k)}{\Psi_k(\bar{s}_k)} = 1 + \frac{f(x_k + \bar{s}_k) - f(x_k) - \Psi_k(\bar{s}_k)}{\Psi_k(\bar{s}_k)}$$

$$\text{Mas } |f(x_k + \bar{s}_k) - f(x_k) - \Psi_k(\bar{s}_k)| =$$

$$= |f(x_k + \bar{s}_k) - f(x_k) - \xi_k^t \bar{s}_k - \frac{1}{2} \bar{s}_k^t B_k \bar{s}_k| \leq$$

$$|f(x_k + \bar{s}_k) - f(x_k) - \xi_k^t \bar{s}_k| + \frac{1}{2} |\bar{s}_k^t B_k \bar{s}_k| \leq L \|\bar{s}_k\|^2 + \frac{1}{2} \|B_k\| \|\bar{s}_k\|^2 \leq$$

$$\leq \left(L + \frac{M}{2} \right) \|\bar{s}_k\|^2, \text{ onde } L \text{ é a constante de Lipschitz para } \nabla f \text{ e}$$

usamos o fato de que $\|B_k\| \leq M, \forall k \in \mathbb{N}$.

Dessa forma, temos :

$$|f(x_k + \bar{s}_k) - f(x_k) - \Psi_k(\bar{s}_k)| \leq \left(\frac{2L + M}{2} \right) \|\bar{s}_k\|^2.$$

Assim, para $k \in K_1, k \geq k_2$ temos:

$$|\bar{\rho}_k - 1| = \left| \frac{f(x_k + \bar{s}_k) - f(x_k) - \Psi_k(\bar{s}_k)}{\Psi_k(\bar{s}_k)} \right| \leq \frac{\left(\frac{2L + M}{2} \right) \frac{\|\bar{s}_k\|^2}{t_k}}{\left| \frac{\Psi_k(\bar{s}_k)}{t_k} \right|} \leq$$

$$\leq \left(\frac{2L + M}{2|a|} \right) \frac{\|\bar{s}_k\|^2}{t_k} = \left(\frac{2L + M}{2|a|} \right) \frac{\|a_k(t_k) - x_k\|^2}{t_k} = \frac{2L + M}{2|a|} \|v_k\|^2 t_k \xrightarrow{k \in K_2} 0.$$

Portanto, $\lim_{k \in K_1} \bar{\rho}_k = 1$, contradizendo o fato de que

$\bar{\rho}_k \leq 10^{-4}$ para $\forall k \in K_1$.

Logo, x^* satisfaz a CN1 para (2.1).

Consideremos agora o caso (ii): $\inf_{k \in K} \Delta_k > 0$

Seja $K_1 \subset K$ tal que $\lim_{k \in K_1} B_k$ existe. Chamemos $B = \lim_{k \in K_1} B_k$.

Seja $K_2 \subset K_1$ tal que $\lim_{k \in K_2} x_k$ existe. Então $x^* = \lim_{k \in K_2} x_k$.

Temos $f(x_{k+1}) < f(x_k) + 10^{-4} \Psi_k(s_k)$, $\forall k \in K_2$.

Então $\Psi_k(s_k) > 10^4 (f(x_{k+1}) - f(x_k))$, $\forall k \in K_2$

e $|\Psi_k(s_k)| < 10^4 (f(x_k) - f(x_{k+1}))$, $\forall k \in K_2$.

Agora, $f(x_k) - f(x_{k+1}) \longrightarrow 0$ (pois caso contrário,

$f(x_k) \longrightarrow -\infty$), portanto $\lim_{k \in K_2} \Psi_k(s_k) = 0$.

Vamos agora definir $\Psi_*(s) = s^t \xi_* + \frac{1}{2} s^t B s$.

Como $\inf_{k \in K} \Delta_k > 0$, temos $\Delta_k \geq \Delta > 0$, $\forall k \in K_2$

Seja $\bar{s}$ solução do problema

$$\begin{array}{l}
 \text{minimizar } \Psi_*(s) \\
 \text{s/a } x^* + s \in B \\
 \|s\| \leq \Delta/2
 \end{array} \tag{2.5}$$

Existe $\bar{k}$ tal que para $k \geq \bar{k}$, $k \in \mathbb{K}_2$, temos $\|x^* - x_k\| \leq \Delta/2$

Definindo $\bar{x} = x^* + \bar{s}$, temos:

$\|\bar{x} - x_k\| = \|x^* + \bar{s} - x_k\| \leq \|x^* - x_k\| + \|\bar{s}\| \leq \Delta \leq \Delta_k$, $k \in \mathbb{K}_2$, $k \geq \bar{k}$,
 ou seja, $\|\bar{x} - x_k\| \leq \Delta_k$ e como $\bar{x} = x^* + \bar{s} \in B$, segue que $\bar{x} - x_k$ é factível para o subproblema (2.2) e então $\Psi_k(\bar{x} - x_k) \geq \Psi_k(s_k)$.

Agora, no limite para $k \in \mathbb{K}_2$ temos $s_k \rightarrow s_*$, $B_k \rightarrow B$, $\Psi_k(s_k) \rightarrow 0$ e $\bar{x} - x_k \rightarrow \bar{s}$. Logo $\Psi_*(\bar{s}) \geq \Psi_*(0) = 0$. Portanto $s = 0$ minimiza (2.5), valendo a CN1 para (2.5). Ou seja, para todo arco factível ω ($\omega(0) = 0$) tem-se $\nabla \Psi_*^t(0) \frac{d\omega}{dt}(0) \geq 0$. Chamando $\omega(t) = \alpha(t) - x^*$, como $\nabla \Psi_*(0) = s_*$ e $\frac{d\omega}{dt}(t) = \frac{d\alpha}{dt}(t)$, então para todo α arco factível para (2.1) ($\alpha(0) = x^*$) teremos $s_*^t \alpha'(0) \geq 0$. Ou seja, vale a CN1 para (2.1) ■

Teorema 2.14 (BOA DEFINIÇÃO - algoritmo II - CN2)

Supondo que $f \in C^2(A)$, A aberto, $A \supset B$ e que $\nabla^2 f$ é Lipschitz contínua em A , seja $\{x_k\}_{k \in \mathbb{N}}$ a sequência gerada pelo algoritmo II, com $B_k \equiv \nabla^2 f(x_k)$, $\forall k \in \mathbb{N}$.

Se x_k não satisfaz a CN2, então é possível obter x_{k+1} .

Prova:

Análoga à do teorema 2.10. ■

Teorema 2.15 (CONVERGÊNCIA GLOBAL - algoritmo II - CN2)

Supondo que $f \in C^2(A)$, A aberto, $A \supset B$ e que $\nabla^2 f$ é Lipschitz

contínua em $\mathbb{A}$, seja $\left\{ x_k \right\}_{k \in \mathbb{N}}$ a sequência gerada pelo algoritmo II.

Se $B_k \equiv \nabla^2 f(x_k)$, $\forall k \in \mathbb{N}$ e $\lim_{k \in K} x_k = x^*$, $K \subset \infty \mathbb{N}$

então x^* satisfaz a CN2 para (2.1).

Prova:

Segue combinando-se os argumentos das provas dos teoremas 2.11 e 2.13. ■

CAPÍTULO 3

MÉTODOS DE REGIÃO DE CONFIANÇA PARA MINIMIZAÇÃO EM BOLAS EUCLIDIANAS

3.1 O SUBPROBLEMA

Considerando os algoritmos I e II, o subproblema (2.2) não é sempre fácil de resolver, pois sua região admissível é a interseção de uma bola com um conjunto arbitrário. Para algumas formas de B , no entanto, estes algoritmos tornam-se implementáveis. Quando B é uma bola euclidiana (ou uma esfera, ou o complemento de uma bola, etc), (2.2) pode ser resolvido usando-se a caracterização de minimizadores locais de Martínez [11]. Outro caso implementável ocorre quando B é um polítopo e as regiões de confiança são paralelepípedos. Neste capítulo analisamos os aspectos teóricos do caso em que B é uma bola euclidiana, e então (2.2) reduz-se a minimizar uma quadrática na interseção de duas bolas:

$$\left[\begin{array}{l} \text{minimizar } \Psi_k(s) = s^t g_k + \frac{1}{2} s^t B_k s \\ s/a \quad \|x_k + s\| \leq r \\ \quad \|s\| \leq \Delta_k, \quad s \in \mathbb{R}^n \end{array} \right] \quad (3.1)$$

O conjunto factível descrito pelas restrições de (3.1) pode ser dividido em quatro regiões :

$$\begin{aligned} R_1 &= \left\{ s \in \mathbb{R}^n : \|s\| < \Delta_k, \|x_k + s\| < r \right\} \\ R_2 &= \left\{ s \in \mathbb{R}^n : \|s\| = \Delta_k, \|x_k + s\| < r \right\} \\ R_3 &= \left\{ s \in \mathbb{R}^n : \|s\| < \Delta_k, \|x_k + s\| = r \right\} \\ R_4 &= \left\{ s \in \mathbb{R}^n : \|s\| = \Delta_k, \|x_k + s\| = r \right\} \end{aligned}$$

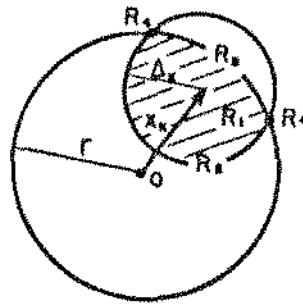


figura 3.1

Chamando de $\hat{s}$ um minimizador global do subproblema (3.1), podemos provar que:

A) Se $\hat{s} \in R_1$, então $\hat{s}$ é minimizador global de

$$\min \Psi_k(s) \quad \text{s/a} \quad s \in \mathbb{R}^n \quad \rightarrow B_k \hat{s} + \xi_k = 0$$

B) Se $\hat{s} \in R_2$, então $\hat{s}$ é minimizador local de

$$\min \Psi_k(s) \quad \text{s/a} \quad \|s\| = \Delta_k$$

C) Se $\hat{s} \in R_3$, então $\hat{s}$ é minimizador local de

$$\min \Psi_k(s) \quad \text{s/a} \quad \|x_k + s\| = r$$

D) Se $\hat{s} \in R_4$, então $\hat{s}$ é minimizador global de

$$\min \Psi_k(s) \quad \text{s/a} \quad \|s\| = \Delta_k, \quad \|x_k + s\| = r$$

Com base na partição determinada pelas regiões R_1 , R_2 , R_3 e R_4 , e considerando ainda as possíveis localizações para $\hat{s}$, estabelecemos o seguinte algoritmo para resolução do subproblema (3.1):

3.2 ALGORITMO III

1) Resolver $\min \Psi_k(s) \quad s/a \quad s \in \mathbb{R}^n$, obtendo SI.

Se SI é factível, i.e., $\|SI\| \leq \Delta_k$ e $\|x_k + SI\| \leq r$, FIM

Senão,

2) resolver $\min \Psi_k(s) \quad s/a \quad \|s\| = \Delta_k$, obtendo SGEC.

Se SGEC é factível, guardar.

Senão, resolver

$\min \text{loc } \Psi_k(s) \quad s/a \quad \|s\| = \Delta_k$, obtendo SLEC.

Se SLEC é factível, guardar.

3) Resolver $\min \Psi_k(s) \quad s/a \quad \|x_k + s\| = r$, obtendo SGEP.

Se SGEP é factível, guardar.

Senão, resolver

$\min \text{loc } \Psi_k(s) \quad s/a \quad \|x_k + s\| = r$, obtendo SLEP.

Se SLEP é factível, guardar.

4) Se nem SGEC nem SGEP forem factíveis, resolver

$\min \Psi_k(s) \quad s/a \quad \|s\| = \Delta_k, \|x_k + s\| = r$, obtendo SG2E

5) Comparar o valor de Ψ_k nas soluções guardadas e decidir qual a solução global do subproblema (3.1)

OBS: SI: solução irrestrita

SGEC: solução global na esfera de confiança
 SLEC: solução local na esfera de confiança
 SGEP: solução global na esfera do problema
 SLEP: solução local na esfera do problema
 SG2E: solução global nas duas esferas

3.3 ANÁLISE DO ALGORITMO III

Observamos inicialmente que o problema irrestrito do passo 1 tem sentido apenas quando $\Psi_k(s)$ for convexa ($B_k \geq 0$). Com relação aos problemas dos passos 2 e 3, podemos tratá-los como variantes de

$$\min \Psi_k(s) \quad s/a \parallel s + v \parallel = R \quad (3.2)$$

e de $\min \text{loc } \Psi_k(s) \quad s/a \parallel s + v \parallel = R \quad (3.3)$

que para serem resolvidos requerem algoritmos eficientes para minimizar quadráticas em esferas, tanto global quanto localmente. Para o problema do passo 4, onde a interseção das duas esferas é uma esfera de dimensão $n-2$, basta fazer uma mudança de variáveis adequada para recair no formato (3.2) acima, com $s \in \mathbb{R}^{n-1}$.

3.3.1 Minimizadores globais de quadráticas em esferas

Vamos analisar o problema (3.2), que é equivalente a :

$$\min \Psi_k(s) \quad s/a (s + v)^t (s + v) = R^2 \quad (3.2.1)$$

Substituímos a resolução de (3.2.1) pela determinação da direção s e do multiplicador μ que zerem o gradiente do Lagrangeano associado a (3.2.1). Desta forma, queremos encontrar $s \in \mathbb{R}^n$ e $\mu \in \mathbb{R}$ tais que

$$\nabla L = B_k s + g_k + \mu (s + v) = 0 \quad (3.2.2)$$

Fazendo a mudança de variáveis $z = s + v$ (3.2.3), temos:

$$B_k(z - v) + \xi_k + \mu z = 0 \Rightarrow (B_k + \mu I) z = B_k v - \xi_k$$

$$\Rightarrow \boxed{z = (B_k + \mu I)^+ (B_k v - \xi_k)} \quad (3.2.4)$$

onde $+$ indica a pseudo-inversa de Moore-Penrose.

Sendo B_k uma matriz simétrica, tomemos sua decomposição espectral $\boxed{B_k = Q_k D_k Q_k^t}$ (3.2.5), onde $Q_k Q_k^t = Q_k^t Q_k = I_n$ e $D_k = \text{diag} (\lambda_1, \dots, \lambda_n)$, com $\lambda_1 \leq \dots \leq \lambda_n$. Na notação de vetores coluna, $Q_k = (q_1, \dots, q_n)$, $q_i \in \mathbb{R}^n$, $i = 1, \dots, n$.

Substituindo (3.2.5) em (3.2.4) :

$$z = (Q_k D_k Q_k^t + \mu Q_k Q_k^t)^+ (Q_k D_k Q_k^t v - \xi_k)$$

$$\Rightarrow z = [Q_k (D_k + \mu I) Q_k^t]^+ (Q_k D_k Q_k^t v - \xi_k)$$

$$\Rightarrow z = Q_k (D_k + \mu I)^+ Q_k^t (Q_k D_k Q_k^t v - \xi_k)$$

$$\Rightarrow \boxed{z = Q_k (D_k + \mu I)^+ (D_k Q_k^t v - Q_k^t \xi_k)} \quad (3.2.6)$$

Chamando $\boxed{Q_k^t v = u}$ (3.2.7) e $\boxed{Q_k^t \xi_k = c}$ (3.2.8) :

$$\boxed{z = Q_k (D_k + \mu I)^+ (D_k u - c)} \quad (3.2.9)$$

Tomemos o quadrado da norma euclidiana de z , que estabelece a função $\varphi : \mathbb{R} \rightarrow \mathbb{R}^+$:

$$\boxed{\|z\|^2 = \|(D_k + \mu I)^+ (D_k u - c)\|^2 = \varphi(\mu)} \quad (3.2.10)$$

Desta forma, a resolução do sistema não linear (3.2.2) recal na determinação de zero da função real $\boxed{\varphi(\mu) = R^2}$ (3.2.11)

De acordo com a caracterização de Gay [3] e Moré - Sorensen [16], se $\mu \geq -\lambda_1$, então a direção $z = s + v$ correspondente é um minimizador global de (3.2). Se $B_k v - \xi_k \in S_1^\perp$, onde S_1 é o autoespaço

associado ao menor autovalor de B_k , ou se $\varphi(-\lambda_1) > R^2$, existe uma única solução (3.2.9) para $\mu \in]-\lambda_1, \omega)$ e neste caso, $\mu > -\lambda_1$. Agora, se $B_k v - \xi_k \in S_1^\perp$ e $\varphi(-\lambda_1) \leq R^2$, então (3.2.9) é satisfeita para $\mu = -\lambda_1$, para toda direção z tal que $\|z\| = R$ e z pertença à variedade linear :

$$V = \left\{ y \in \mathbb{R}^n : y = (B_k - \lambda_1 I)^+ (B_k v - \xi_k) + \sum_{j=1}^{\dim S_1} \alpha_j q_j \right\},$$

onde $\dim S_1$ é a dimensão de S_1 . Este caso é conhecido como "hard-case", devido à não trivialidade computacional envolvida e será tratado com detalhes na seção 3.4 deste capítulo.

Uma outra maneira de escrever a função $\varphi(\mu)$, que ressalta sua estrutura racional é:

$$\varphi(\mu) = \sum_{i \in I} \left[\frac{\lambda_i u_i - c_i}{\lambda_i + \mu} \right]^2 \quad (3.2.12)$$

onde $I = \{ i \in \{1, \dots, n\}, c_i \neq \lambda_i u_i \}$, $u_i = q_i^t v$ e $c_i = q_i^t \xi_k$

Temos então que:

$$\varphi'(\mu) = -2 \sum_{i \in I} \frac{(\lambda_i u_i - c_i)^2}{(\lambda_i + \mu)^3} \quad (3.2.13)$$

$$\text{e} \quad \varphi''(\mu) = 6 \sum_{i \in I} \frac{(\lambda_i u_i - c_i)^2}{(\lambda_i + \mu)^4} \quad (3.2.14)$$

Usando-se (3.2.5), (3.2.12)-(3.2.14), estabelece-se facilmente as seguintes propriedades para $\varphi(\mu)$, conforme apresenta Martínez [11]:

(a) Seja $\Omega = \mathbb{R} - \{-\lambda_i, i \in I\}$. φ está bem definida para todo $\mu \in \Omega$ e $\varphi \in C^\infty(\Omega)$.

(b) $\varphi''(\mu) \geq 0$ para todo $\mu \in \Omega$. Em consequência, φ é

convexa em qualquer intervalo contido em Ω .

(c) Se $i \in I$, ou seja, para i tal que $-\lambda_i \in \Omega$, temos

$$\lim_{\mu \rightarrow -\lambda_i} \varphi(\mu) = \infty ; \quad \lim_{\mu \rightarrow -\lambda_i^+} \varphi'(\mu) = -\infty \quad \text{e} \quad \lim_{\mu \rightarrow -\lambda_i^-} \varphi'(\mu) = \infty$$

$$(d) \quad \lim_{\mu \rightarrow \infty} \varphi(\mu) = \lim_{\mu \rightarrow -\infty} \varphi(\mu) = 0$$

Devido aos pólos que $\varphi(\mu)$ apresenta no conjunto $\{-\lambda_i, i \in I\}$, Reinsch [20] e Hebden [7] sugerem que ao invés de resolver a equação (3.2.11), utilize-se $\frac{1}{\Phi(\mu)} = \frac{1}{R}$ (3.2.15), onde

$\Phi(\mu) = [\varphi(\mu)]^{1/2}$, para contornar possíveis instabilidades numéricas.

Reinsch estabeleceu as seguintes propriedades para $\frac{1}{\Phi(\mu)}$:

- (a) $\frac{1}{\Phi(\mu)}$ está bem definida para todo $\mu \in \mathbb{R}$
- (b) $\frac{1}{\Phi(\mu)}$ é estritamente crescente em $[-\lambda_1, \infty)$
- (c) $\frac{1}{\Phi(\mu)}$ é côncava em $[-\lambda_1, \infty)$

Na figura 3.2, extraída del [16], podemos visualizar, de modo comparativo, as propriedades de $\Phi(\mu)$ e $\frac{1}{\Phi(\mu)}$.

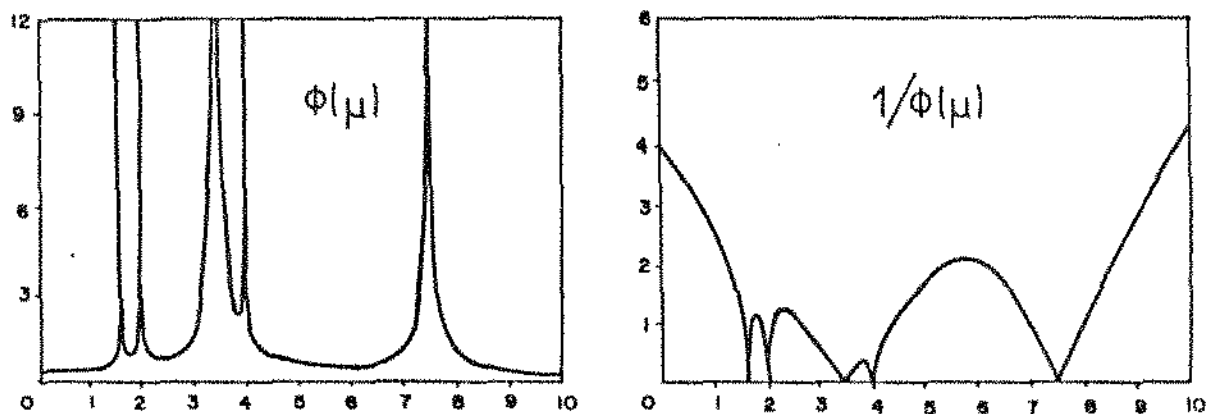


figura 3.2

Pelas características de $\frac{1}{\Phi(\mu)}$, que além de não possuir pólos, tende a ser quase linear em $[-\lambda_1, \infty)$, temos não só garantia de convergência global do método de Newton aplicado à equação (3.2.15), como também um desempenho computacional bastante satisfatório para este método. Usaremos, portanto, o seguinte esquema iterativo:

$$\mu_{k+1} = \mu_k - \frac{\frac{1}{\Phi(\mu_k)} - \frac{1}{R}}{\left[\frac{1}{\Phi(\mu_k)} - \frac{1}{R}\right]}, = \mu_k - \frac{\frac{R - \Phi(\mu_k)}{R \Phi(\mu_k)}}{\frac{\Phi'(\mu_k)}{\Phi(\mu_k)^2}}$$

$$\rightarrow \mu_{k+1} = \mu_k + \frac{R - \Phi(\mu_k)}{\Phi'(\mu_k)} \frac{\Phi(\mu_k)}{R} \quad (3.2.16)$$

onde $\Phi(\mu) = \left[\sum_{i \in I} \left(\frac{\lambda_i u_i - c_i}{\lambda_i + \mu} \right)^2 \right]^{1/2} \quad (3.2.17)$

e $\Phi'(\mu) = \frac{- \sum_{i \in I} \frac{(\lambda_i u_i - c_i)^2}{(\lambda_i + \mu)^3}}{\Phi(\mu)} \quad (3.2.18)$

À medida que $\Phi(\mu)$ se aproxima de R , (3.2.16) comporta-se como o método de Newton aplicado a $\Phi(\mu) = R$, com convergência local q-quadrática para μ^* (conforme [1], p.136).

Sendo $\frac{1}{\Phi(\mu)}$ uma função côncava em $[-\lambda_1, \infty)$, o ponto inicial μ_0 deve ser tal que $\frac{1}{\Phi(\mu_0)} < \frac{1}{R}$, e pode ser obtido por redução do intervalo $[-\lambda_1, -\lambda_1 + 0.5]$.

Assim, pela resolução de (3.2.15) usando-se (3.2.16), uma vez encontrado o multiplicador μ correspondente à solução global, voltamos a (3.2.9) e (3.2.3), obtendo o minimizador global do

problema (3.2):

$$s = Q_k (D_k + \mu I)^+ (D_k u - c) - v \quad (3.2.19)$$

com u e c dados por (3.2.7) e (3.2.8), respectivamente.

3.3.2 Minimizadores locais de quadráticas em esferas

Vamos agora analisar o problema (3.3), onde queremos encontrar minimizador local não global. Martínez [11] apresenta uma teoria para resolver este tipo de problema, que resumiremos no que se segue.

Pela mudança de variáveis (3.2.3), o problema (3.3) pode ser reescrito de modo equivalente como:

$$\begin{aligned} \min \text{ loc } \bar{\Psi}_k(z) &= \xi_k^t (z - v) + \frac{1}{2} (z - v)^t B_k (z - v) \\ \text{s/a } z^t z &= R^2 \end{aligned} \quad (3.3.1)$$

Segundo a teoria clássica dos multiplicadores de Lagrange, se z^* é um minimizador local de (3.3.1), então

$$\begin{aligned} &\left[\begin{array}{l} B_k(z^* - v) + \xi_k + \mu z^* = 0 \\ \| z^* \| = R^2 \end{array} \right] \\ \rightarrow &\left[\begin{array}{l} (B_k + \mu I)z^* = B_k v - \xi_k \\ \| z^* \| = R^2 \end{array} \right] \quad \text{para algum } \mu \in \mathbb{R} \end{aligned} \quad (3.3.2)$$

Martínez caracteriza as soluções de (3.3.1) nos seguintes resultados:

Teorema 3.1

"Se z^* é um minimizador local de (3.3.1), então (3.3.2) vale com $\mu > -\lambda_2$."

Teorema 3.2

(i) " Se $\lambda_1 u_1 = c_1$, isto é, se $(B_k v - g_k) \in S_1^\perp$, então não existem minimizadores locais para (3.3.1) com $\mu \in [-\lambda_2, -\lambda_1]$."

(ii) "Existe no máximo um minimizador local para (3.3.1) com $\mu \in (-\lambda_2, -\lambda_1)$. Para este minimizador, $\varphi(\mu)$ está bem definida e $\varphi'(\mu) \geq 0$."

(iii) "Se (z^*, μ) satisfaz (3.3.2) com $\mu \in (-\lambda_2, -\lambda_1)$ e $\varphi'(\mu) > 0$, então z^* é um minimizador local estrito para (3.3.1)."

Cabe observar que se $\lambda_2 = \lambda_1$, a teoria de Moré - Sorensen e Gay, aliada ao teorema 3.2 de Martínez mostra que não existem minimizadores locais para (3.3.1). Isto justifica a hipótese de dimensão um para S_1 feita no teorema 3.2.

O algoritmo IV a seguir, para calcular o minimizador local não global do problema (3.3.1) foi introduzido por Martínez [11]. Pressupõe o conhecimento de (z_g, μ_g) , solução global de (3.3.2). Se $\mu_g = -\lambda_1$, isto é, se $B_k + \mu_g I$ é singular, então $\lambda_1 u_1 = c_1$ e pelo teorema 3.2 sabemos que não existem minimizadores locais não globais para (3.3.1). Podemos então assumir que $\mu_g > -\lambda_1$. Este algoritmo compreende duas fases. Na primeira busca-se μ tal que $\varphi(\mu) \geq R^2$ e $\varphi'(\mu) \geq 0$ e na segunda, o objetivo é encontrar μ tal que $\varphi(\mu) = R^2$ e $\varphi'(\mu) \geq 0$.

3.3.2.1 Algoritmo IV

fase 1:

0) Sejam $\mu_0 < -\lambda_1$ ($B_k + \mu_0 I$ não é semi-positiva definida),

$k = 0$, tomar $\sigma_1, \sigma_2 \in (0,1)$, $\sigma_1 < \sigma_2$ e $\mu_0^u = \mu_g$

1) Calcular $\varphi(\mu_k)$ e $\varphi'(\mu_k)$.

2) Se $\varphi(\mu_k) \geq R^2$ e $\varphi'(\mu_k) \geq 0$, fim da fase 1

3) Escolher $\tilde{\mu} \in \left[\mu_k + \sigma_1 (\mu_k^u - \mu_k), \mu_k + \sigma_2 (\mu_k^u - \mu_k) \right]$

4) Se $B_k + \tilde{\mu} I \geq 0$ (i.e. $\tilde{\mu} \geq -\lambda_1$)

então definir $\mu_{k+1} = \mu_k$

$$\mu_{k+1}^u = \tilde{\mu}$$

senão definir $\mu_{k+1} = \tilde{\mu}$

$$\mu_{k+1}^u = \mu_k^u$$

5) $k = k + 1$, ir para 1

fase 2:

0) $\mu_k^l = \mu_{k-1}$ ($\varphi(\mu_k^l) < R^2$ ou $\varphi'(\mu_k^l) < 0$)

1) Se $\varphi(\mu_k) = R^2$ e $\varphi'(\mu_k) \geq 0$, fim

2) Calcular $\mu_k^n = \mu_k - \frac{\varphi(\mu_k) - R^2}{\varphi'(\mu_k)}$

Se $\mu_k^n < \mu_k^l$, parar

3) Escolher $\tilde{\mu} \in \left[\mu_k^l + \sigma_1 (\mu_k - \mu_k^l), \mu_k^l + \sigma_2 (\mu_k - \mu_k^l) \right]$

4) Se $\varphi(\tilde{\mu}) \geq R^2$ e $\varphi'(\tilde{\mu}) \geq 0$

então definir $\mu_{k+1} = \tilde{\mu}$

$$\mu_{k+1}^l = \mu_k^l$$

senão definir $\mu_{k+1} = \mu_k$

$$\mu_{k+1}^l = \tilde{\mu}$$

5) $k = k + 1$, ir para 1

Os resultados teóricos de convergência para o algoritmo IV são os seguintes:

Teorema 3.3

"A fase 1 do algoritmo IV termina num número finito de passos, gerando μ_k tal que $\varphi(\mu_k) \geq R^2$ e $\varphi'(\mu_k) \geq 0$ ou então gera uma sequência infinita tal que $\lim_{k \rightarrow \infty} \mu_k = -\lambda_1$. Neste caso, $\lim_{k \rightarrow \infty} \mu_k^u = -\lambda_1$ e $\lambda_1 u_1 = c_1$."

Prova: Ver [11]. ■

Teorema 3.4

Assumindo que no algoritmo IV, μ_0 é escolhido no intervalo $(-\lambda_2, -\lambda_1)$ e que a fase 1 termina após um número finito K de iterações, temos:

(i) "Se a fase 2 do algoritmo IV termina no passo 2, então não existe minimizador local não global para (3.3.1)."

(ii) "Se a fase 2 do algoritmo IV define uma sequência infinita μ_k , então existe $\mu^* \in (-\lambda_2, -\lambda_1)$ tal que $\lim_{k \rightarrow \infty} \mu_k = \lim_{k \rightarrow \infty} \mu_k^l = \mu^*$; $\varphi(\mu^*) \geq R^2$; $\varphi'(\mu^*) \geq 0$ e $\varphi(\mu^*) = R^2$ ou $\varphi'(\mu^*) = 0$."

(iii) "Se $\varphi(\mu^*) > R^2$ e $\varphi'(\mu^*) = 0$, não existe minimizador local não global para (3.3.1). Se $\varphi(\mu^*) = R^2$ e $\varphi'(\mu^*) > 0$, então μ^* define o minimizador local não global de (3.3.1):

$$z^* = (B_k + \mu^* I)^+ (B_k v - g_k)."$$

Prova: Ver [11]. ■

Desta forma, uma vez obtido o par (z^*, μ^*) , solução local não

global para (3.3.1), basta fazermos $s^* = z^* - v$ e então (s^*, μ^*) é a solução local não global para o problema (3.3).

Pela figura 3.3 podemos ter uma idéia geométrica do funcionamento do algoritmo de Martínez:

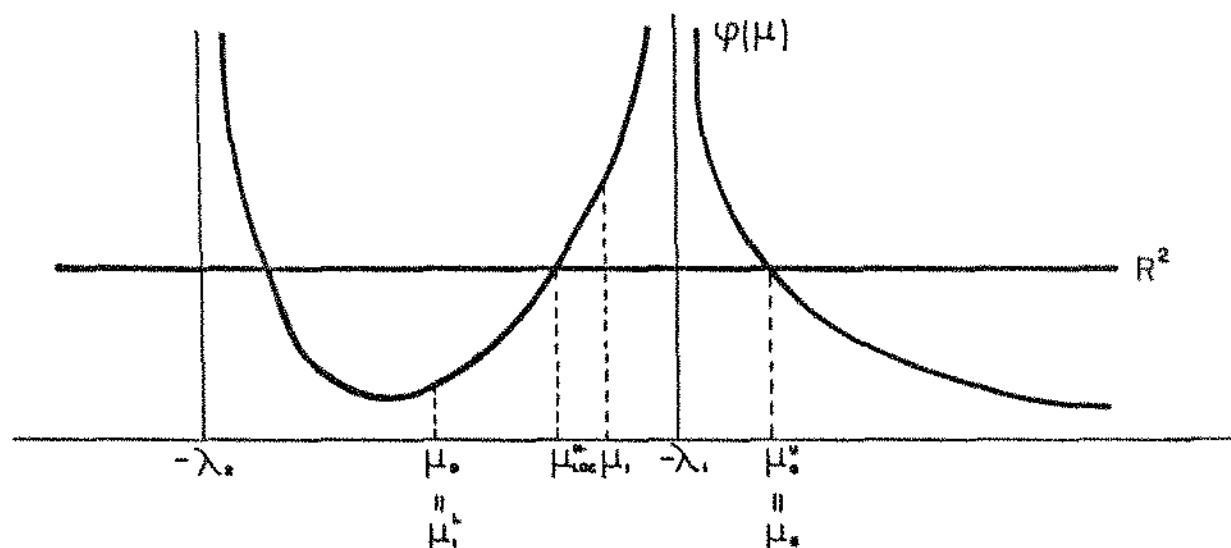


figura 3.3 -Na fase 1, pela redução do intervalo $[\mu_0, \mu_0^u]$, busca-se um ponto μ_k tal que $\varphi(\mu_k) \geq R^2$ e $\varphi'(\mu_k) \geq 0$, o que ocorre nesta figura para $\mu_k = \mu_1$. Passa-se então para a fase 2, onde, pela redução do intervalo inicial $[\mu_1^l, \mu_1]$, busca-se isolar o ponto μ_k tal que $\varphi(\mu_k) = R^2$ e $\varphi'(\mu_k) \geq 0$.

3.3.3 Minimizadores de quadráticas em esferas $(n-2)$ dimensionais

Trataremos agora da mudança de variáveis que permite reduzir o problema do passo 4 do algoritmo III ao formato (3.2) já analisado. Queremos resolver o problema:

$$\min \Psi_k(s) \quad \text{s/a} \quad \|s\| = \Delta_k, \quad \|x_k + s\| = r \quad (3.4)$$

Suponhamos que $x_k \neq 0$ e consideremos a família de direções

$$\mathcal{Y} = \left\{ s \in \mathbb{R}^n : \|s\| = \Delta_k, \|x_k + s\| = r \right\} \quad (3.4.1),$$

o escalar $\alpha \in \mathbb{R}$ tal que

$$\alpha = \frac{r^2 - \Delta_k^2}{2 \|x_k\|^2} + \frac{\|x_k\|}{2} \quad (3.4.2),$$

o vetor

$$c = \frac{\alpha}{\|x_k\|} x_k \quad (3.4.3)$$

e o hiperplano

$$\mathcal{X} = \left\{ w \in \mathbb{R}^n : x_k^t (w - c) = 0 \right\} \quad (3.4.4).$$

Vamos mostrar que para $s \in \mathcal{Y}$, temos $x_k + s \in \mathcal{X}$ e que $\mathcal{Y}$ define o lugar geométrico dos pontos equidistantes a c .

De fato, para $s \in \mathcal{Y}$ temos

$$s^t s = \Delta_k^2 \quad (3.4.5) \quad e$$

$$(x_k + s)^t (x_k + s) = r^2 \quad (3.4.6).$$

Por (3.4.6) temos que $x_k^t x_k + 2 s^t x_k + s^t s = r^2$.

Substituindo (3.4.5) e rearranjando:

$$s^t x_k = \frac{r^2 - \Delta_k^2 - x_k^t x_k}{2}$$

Somando $x_k^t x_k$ em ambos os membros:

$$x_k^t (s + x_k) = \frac{r^2 - \Delta_k^2 + x_k^t x_k}{2} = \left(\frac{r^2 - \Delta_k^2}{2 \|x_k\|^2} + \frac{\|x_k\|}{2} \right) \|x_k\|$$

$$\rightarrow x_k^t (s + x_k) = \alpha \|x_k\| = \frac{\alpha x_k^t x_k}{\|x_k\|} = c^t x_k$$

Logo $x_k^t (s + x_k - c) = 0$ e portanto $x_k + s \in \mathcal{X}$ quando $s \in \mathcal{Y}$.

Vejam os agora que a distância de c a $\mathcal{Y}$ é constante.

$$d^2(c, \mathcal{Y}) = \inf_{s \in \mathcal{Y}} d^2(c, x_k + s)$$

Seja uma direção arbitrária $p \in \mathcal{Y}$

$$\begin{aligned} d^2(c, x_k + p) &= \|c - (x_k + p)\|^2 = (c - x_k - p)^t (c - x_k - p) = \\ &= c^t (c - x_k - p) - x_k^t (c - x_k - p) - p^t (c - x_k - p) = \\ &= c^t c - c^t x_k - c^t p - p^t c + p^t x_k + p^t p = \\ &= c^t c + p^t p - 2 c^t p - c^t x_k + p^t x_k = \\ &= \alpha^2 + \Delta_k^2 - \alpha \|x_k\| + p^t x_k \left(1 - \frac{2\alpha}{\|x_k\|} \right) \end{aligned}$$

Mas

$$\begin{aligned} p^t x_k &= \frac{r^2 - \Delta_k^2 - x_k^t x_k}{2} = \left(\frac{r^2 - \Delta_k^2 - \|x_k\|^2}{2 \|x_k\|} \right) \|x_k\| = \\ &= \left(\frac{r^2 - \Delta_k^2}{2 \|x_k\|} + \frac{\|x_k\|}{2} - \|x_k\| \right) \|x_k\| = (\alpha - \|x_k\|) \|x_k\| \end{aligned}$$

$$\begin{aligned} \rightarrow d^2(c, x_k + p) &= \alpha^2 + \Delta_k^2 - \alpha \|x_k\| + \|x_k\| (\alpha - \|x_k\|) \left(1 - \frac{2\alpha}{\|x_k\|} \right) = \\ &= \alpha^2 + \Delta_k^2 - \alpha \|x_k\| + \alpha \|x_k\| - \|x_k\|^2 - 2\alpha^2 + 2\alpha \|x_k\| = \\ &= -\alpha^2 + \Delta_k^2 - \|x_k\|^2 + 2 \|x_k\| \left(\frac{r^2 - \Delta_k^2}{2 \|x_k\|} + \frac{\|x_k\|}{2} \right) = r^2 - \alpha^2, \end{aligned}$$

que é uma constante independente de p .

Chamando $\boxed{r^2 - \alpha^2 = \rho^2}$ (3.4.7) e tomando o infimo de

$d^2(c, x_k + p)$, $p \in \mathcal{Y}$:

$$d^2(c, \mathcal{Y}) = \inf_{p \in \mathcal{Y}} d^2(c, x_k + p) = \inf_{p \in \mathcal{Y}} \rho^2 = \rho^2$$

OBS: Se $x_k = 0$, $\mathcal{Y} \neq \emptyset$ apenas quando $r = \Delta_k$ e então o problema (3.4.1)

se reduz diretamente ao formato (3.2).

Consideraremos portanto a seguinte caracterização equivalente para a família $\mathfrak{F}$:

$$s \in \mathfrak{F} \Leftrightarrow \boxed{\begin{array}{l} \|x_k + s - c\| = \rho \\ x_k + s \in \mathcal{X} \end{array}} \quad (3.4.8)$$

onde c , $\mathcal{X}$ e ρ são dados respectivamente por (3.4.3), (3.4.4) e (3.4.7)

Chamando de $x_k^\perp$ o subespaço ortogonal a x_k , seja A a matriz cujas colunas formam uma base para $x_k^\perp$ ($A \in \mathbb{R}^{n \times n-1}$)

$$\begin{aligned} x_k + s \in \mathcal{X} &\rightarrow x_k + s - c \in x_k^\perp \\ \rightarrow \boxed{x_k + s - c = A y, \quad y \in \mathbb{R}^{n-1}} &\quad (3.4.9) \end{aligned}$$

Para obtermos a matriz A , vamos tomar a transformação de Householder H tal que $H \hat{x}_k = e_1$, onde $e_1 = (1, 0, \dots, 0) \in \mathbb{R}^n$ e

$\hat{x}_k = \frac{x_k}{\|x_k\|}$ (conforme [6]). Então $H = I - 2 \frac{vv^t}{v^t v}$, $v = \hat{x}_k + \text{sgn}(\hat{x}_k(1)) e_1$. Chamando $H = (h_1, h_2, \dots, h_n)$, $h_i \in \mathbb{R}^n$, $i = 1 \dots n$, temos $\hat{x}_k^t H^t = \hat{x}_k^t H = e_1^t \rightarrow \hat{x}_k^t h_j = 0$, $j = 2 \dots n$. Logo

$$\boxed{A = (h_2, \dots, h_n)} \quad (3.4.10)$$

Como $H^t H = I_n \rightarrow A^t A = I_{n-1}$, seguindo que $\|A y\|^2 = y^t A^t A y = y^t y = \|y\|^2$

Prosseguindo com as equivalências, temos:

$$(3.4.8) \Leftrightarrow \boxed{\begin{array}{l} \|x_k + s - c\| = \rho \\ x_k + s - c = A y \end{array}} \Leftrightarrow \|A y\| = \rho \Leftrightarrow \boxed{\|y\| = \rho} \quad (3.4.11)$$

Tratemos agora da quadrática resultante desta mudança de variáveis:

$$s = A y + c - x_k = A y + \frac{\alpha}{\|x_k\|} x_k - x_k = A y + (\alpha - \|x_k\|) \hat{x}_k$$

$$\rightarrow \boxed{s = Ay + \beta \hat{x}_k} \quad (3.4.12)$$

onde $\beta = \alpha - \|x_k\|$, α dado por (3.4.2) e $\hat{x}_k = \frac{x_k}{\|x_k\|}$

Tínhamos $\Psi_k(s) = \xi_k^t s + \frac{1}{2} s^t B_k s$

Substituindo (3.4.12) :

$$\begin{aligned} \boxed{\Psi_k(s) = \Psi_k(y)} &= \xi_k^t (Ay + \beta \hat{x}_k) + \frac{1}{2} (Ay + \beta \hat{x}_k)^t B_k (Ay + \beta \hat{x}_k) = \\ &= \xi_k^t Ay + \beta \xi_k^t \hat{x}_k + \frac{1}{2} y^t A^t B_k Ay + \beta \hat{x}_k^t B_k Ay + \frac{\beta^2}{2} \hat{x}_k^t B_k \hat{x}_k = \\ &= \boxed{\Psi_k(\beta \hat{x}_k) + (\xi_k^t + \beta \hat{x}_k^t B_k) A y + \frac{1}{2} y^t A^t B_k A y} \end{aligned} \quad (3.4.13)$$

Assim, ao invés de (3.4.1), iremos resolver

$$\boxed{\begin{array}{l} \text{minimizar } \Psi_k(y) \\ \text{s/a } \|y\| = \rho \end{array}} \quad (3.4.14)$$

Seguindo os passos da análise feita para (3.2), temos:

$$\nabla \tilde{L} = A^t (\xi_k + \beta B_k \hat{x}_k) + A^t B_k A y + \mu y = 0$$

$$\text{Chamando } \boxed{\tilde{\xi}_k = A^t (\xi_k + \beta B_k \hat{x}_k)} \quad (3.4.15) \text{ vem}$$

$$\begin{aligned} (A^t B_k A + \mu I) y &= -\tilde{\xi}_k \\ \rightarrow \boxed{y} &= -(A^t B_k A + \mu I)^+ \tilde{\xi}_k \end{aligned} \quad (3.4.16)$$

Fazemos a decomposição espectral: $A^t B_k A = \tilde{Q}_k \tilde{D}_k \tilde{Q}_k^t$, onde $\tilde{Q}_k^t \tilde{Q}_k = \tilde{Q}_k \tilde{Q}_k^t = I_{n-1}$ e $\tilde{D}_k = \text{diag}(\tilde{\lambda}_1, \dots, \tilde{\lambda}_{n-1})$ com $\tilde{\lambda}_1 \leq \dots \leq \tilde{\lambda}_{n-1}$. O multiplicador μ correspondente à solução global de (3.4.14) é obtido resolvendo-se (3.4.15) com $\tilde{g}(\mu) = \|-(\tilde{D}_k + \mu I)^+ \tilde{z}\|$, $\tilde{z} = \tilde{Q}_k \tilde{\xi}_k$ e $R = \rho$. Uma vez estabelecida a solução global de (3.4.14) através de

(3.4.16), retoma-se a variável original por (3.4.12).

3.4 O "HARD-CASE"

Consideremos o problema

$$\min \Psi_k(s) \quad \text{s/a} \quad \|s + v\| = R \quad (3.5)$$

É possível estabelecer o seguinte resultado para as soluções de (3.5):

Teorema 3.5

"Se $B_k v - \xi_k \in S_1^\perp$ e $\varphi(-\lambda_1) \leq R^2$ então o conjunto de minimizadores de (3.5) está contido numa variedade linear."

Prova:

Como $B_k v - \xi_k \in S_1^\perp \Rightarrow B_k v - \xi_k \perp S_1 = \ker(B_k - \lambda_1 I)$, seguindo que $(B_k - \lambda_1 I)z = B_k v - \xi_k$.

Tomemos $w \in \ker(B_k - \lambda_1 I)$, w arbitrário e $\bar{z} = (B_k - \lambda_1 I)^+ (B_k v - \xi_k)$. Consideremos a família de vetores $\bar{z} + tw$, onde $t \in \mathbb{R}$ é tal que $\|\bar{z} + tw\| = R$. Temos:

$$\Psi_k(s) = \xi_k^t s + \frac{1}{2} s^t B_k s$$

$$\Psi_k(z - v) = \Psi_k(z) = (z - v)^t \xi_k + \frac{1}{2} (z - v)^t B_k (z - v)$$

$$\Psi_k(z) = \frac{1}{2} v^t B_k v - v^t \xi_k + (\xi_k - B_k v)^t z + \frac{1}{2} z^t B_k z$$

Como $\xi_k - B_k v = -(B_k - \lambda_1 I) \bar{z}$, chamando $\frac{1}{2} v^t B_k v - v^t \xi_k = k_1$, segue que:

$$\begin{aligned} \Psi_k(\bar{z}) &= k_1 + \frac{1}{2} \bar{z}^t B_k \bar{z} - \bar{z}^t (B_k - \lambda_1 I) \bar{z} = \\ &= k_1 + \frac{1}{2} \left[\bar{z}^t (B_k - \lambda_1 I) \bar{z} + \lambda_1 \bar{z}^t \bar{z} \right] - \bar{z}^t (B_k - \lambda_1 I) \bar{z} = \end{aligned}$$

$$= k_1 - \frac{1}{2} \bar{z}^t (B_k - \lambda_1 I) \bar{z} + \lambda_1 \| \bar{z} \|^2$$

$$\text{Agora, } \Psi_k(\bar{z} + tw) =$$

$$= k_1 - \frac{1}{2} (\bar{z} + tw)^t (B_k - \lambda_1 I) (\bar{z} + tw) + \lambda_1 \| \bar{z} + tw \|^2 =$$

$$= k_1 - \frac{1}{2} \bar{z}^t (B_k - \lambda_1 I) \bar{z} + \lambda_1 \| \bar{z} + tw \|^2$$

e então $\Psi_k(\bar{z} + tw) = \Psi_k(\bar{z})$ para $\bar{z}$ tal que $\| \bar{z} \| = R$, ou seja,

os valores da quadrática ficam constantes na interseção da variedade

$$\text{linear } V = \left\{ y \in \mathbb{R}^n : y = (B_k - \lambda_1 I)^+ (B_k v - \xi_k) + \sum_{j=1}^{\dim 1} \alpha_j q_j \right\},$$

$$\text{com a esfera } \left\{ z \in \mathbb{R}^n : \| z \| = R \right\}. \blacksquare$$

A situação abordada pelo teorema 3.5 é conhecida na literatura como "hard case" ([13]). Neste caso devemos escolher uma solução $s + v$ tal que $\| s + v \| = R$ e $s + v \in V$.

A maneira que escolhemos para decidir sobre a ortogonalidade de um vetor g em relação ao subespaço $S = [v_1, v_2, \dots, v_{\dim}]$ baseia-se no vetor $\cos \Theta = (\cos \theta_1, \dots, \cos \theta_{\dim})^t$, onde $\cos \theta_i = \frac{v_i^t g}{\|v_i\| \|g\|}$, $i = 1, \dots, \dim$.

Se $\| \cos \Theta \| < \varepsilon_{\text{hard}}$, então declaramos $g \perp S \rightarrow g \in S^\perp$.

Observamos que na esfera de confiança: $g = \xi_k$ e $S = S_1 = [q_1, \dots, q_{\dim}]$, na esfera do problema: $g = B_k v - \xi_k$ e $S = S_1$ e na $(n-2)$ -esfera: $g = \tilde{\xi}_k$ e $S = S_2 = [\tilde{q}_1, \dots, \tilde{q}_{\dim 2}]$.

Uma vez detectada a ortogonalidade, como $\rho(-\lambda_1)$ assume um valor finito, a ocorrência do "hard-case" fica condicionada a

$\varphi(-\lambda_1) < \Delta_k^2$, $\varphi(-\lambda_1) < r^2$ ou $\varphi(-\tilde{\lambda}_1) < \rho^2$, conforme o problema esteja na esfera de confiança, do problema ou na $(n-2)$ -esfera. Ilustramos este fato com a figura 3.4.

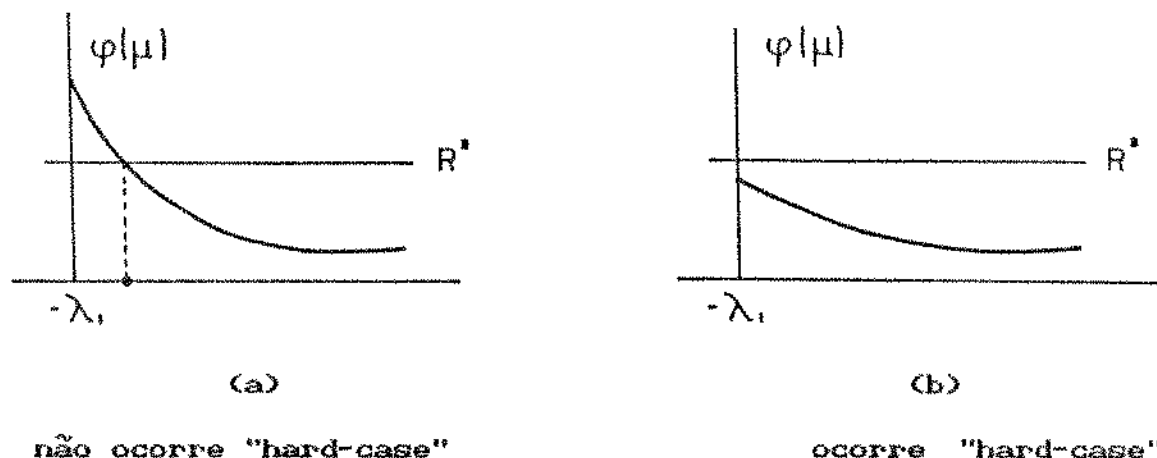


figura 3.4

No que se segue vamos analisar separadamente o "hard-case" na esfera de confiança, na esfera do problema e na $(n-2)$ -esfera.

3.4.1 O "hard case" na esfera de confiança

Neste caso, $v \equiv 0$ e então $(B_k - \lambda_1 I) s = -\xi_k$. Chamemos

$$s_1 = -(B_k - \lambda_1 I)^+ \xi_k \quad (3.6) \text{ e tomemos } s_2 \text{ tal que } s_2 - s_1 \text{ seja a}$$

projeção de $-x_k$ em $\ker(B_k - \lambda_1 I)$. Então

$$s_2 - s_1 = - \left[I - (B_k - \lambda_1 I)^+ (B_k - \lambda_1 I) \right] x_k$$

$$s_2 = s_1 - x_k + (B_k - \lambda_1 I)^+ (B_k - \lambda_1 I) x_k$$

$$s_2 = (B_k - \lambda_1 I)^+ \left[(B_k - \lambda_1 I) x_k - \xi_k \right] - x_k \quad (3.7)$$

Visualizando:

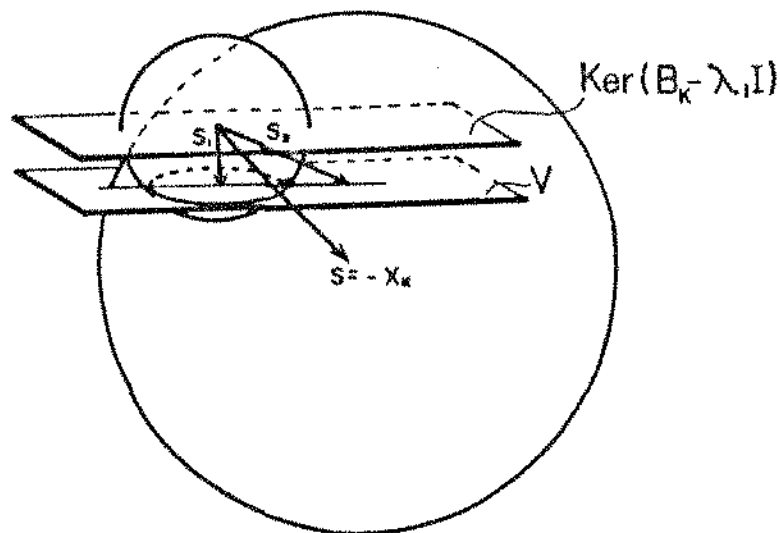


figura 3.5

Tomemos agora a reta $s(\theta) = s_1 + \theta (s_2 - s_1)$, $\theta \in \mathbb{R}$

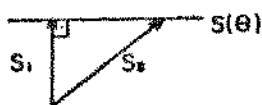


figura 3.6

Queremos a interseção de $s(\theta)$
com a esfera $\|s\| = \Delta_k$,
portanto:

$$\begin{aligned} \left[s_1 + \theta (s_2 - s_1) \right]^t \left[s_1 + \theta (s_2 - s_1) \right] &= \Delta_k^2 \\ s_1^t s_1 + 2 \theta s_1^t (s_2 - s_1) + \theta^2 (s_2 - s_1)^t (s_2 - s_1) &= \Delta_k^2 \end{aligned}$$

$$\theta^2 \|s_2 - s_1\|^2 = \Delta_k^2 - \|s_1\|^2$$

$$\text{Se } s_2 - s_1 = 0 \rightarrow \|s_1\|^2 = \Delta_k^2$$

Supondo $s_2 - s_1 \neq 0$:

$$\theta^2 = \frac{\Delta_k^2 - \|s_1\|^2}{\|s_2 - s_1\|^2}$$

Como o "hard-case" só ocorre quando $\rho(-\lambda_1) \leq \Delta_k^2$, temos

$$\Delta_k^2 \geq \|s_1\|^2 \rightarrow \theta = \pm \sqrt{\frac{\Delta_k^2 - \|s_1\|^2}{\|s_2 - s_1\|^2}}$$

Optaremos pela solução θ tal que $s(\theta)$ esteja mais próxima de

s_2 :

$$\theta = + \sqrt{\frac{\Delta_k^2 - \|s_1\|^2}{\|s_2 - s_1\|^2}} \quad (3.8)$$

$$\begin{aligned} \text{De fato, } \|s_2 - s(\theta-)\| &= \left\| s_2 - \left[s_1 - \theta (s_2 - s_1) \right] \right\| = \\ &= \|s_2 - s_1 + \theta (s_2 - s_1)\| = \|(1 + \theta)(s_2 - s_1)\| \geq \\ &\geq \|(1 - \theta)(s_2 - s_1)\| = \|s_2 - s(\theta+)\| \end{aligned}$$

$$\text{Assim, } s_{a. \text{ coh.}} = s_1 + \theta (s_2 - s_1) \quad (3.9)$$

com s_1 , s_2 e θ dados respectivamente por (3.6), (3.7) e (3.8).

3.4.2 O "hard case" na esfera do problema

Neste caso, $v \equiv x_k$ e então $(B_k - \lambda_1 I)^+ z = B_k x_k - \xi_k$.

Chamemos $z_1 = (B_k - \lambda_1 I)^+ (B_k x_k - \xi_k)$ (3.10) e tomemos z_2 tal

que $z_2 - z_1$ seja a projeção de x_k em $\ker (B_k - \lambda_1 I)$. Então

$$\begin{aligned} z_2 - z_1 &= \left[I - (B_k - \lambda_1 I)^+ (B_k - \lambda_1 I) \right] x_k \\ z_2 &= z_1 + x_k - (B_k - \lambda_1 I)^+ (B_k - \lambda_1 I) x_k \\ z_2 &= (B_k - \lambda_1 I)^+ \left[B_k x_k - \xi_k - B_k x_k + \lambda_1 x_k \right] + x_k \\ z_2 &= (B_k - \lambda_1 I)^+ (\lambda_1 x_k - \xi_k) + x_k \end{aligned} \quad (3.11)$$

Visualizando:

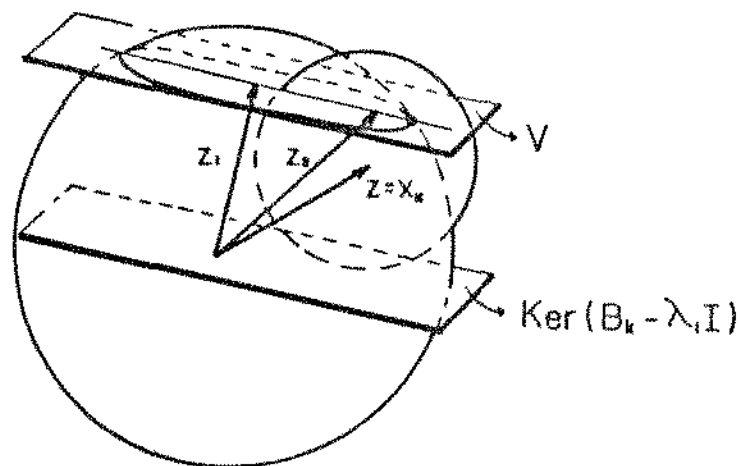


figura 3.7

Tomemos agora a reta $z(\theta) = z_1 + \theta (z_2 - z_1)$, $\theta \in \mathbb{R}$

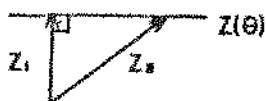


figura 3.8

Queremos a interseção de $z(\theta)$ com a esfera $\|z\| = r$, portanto:

$$\left[z_1 + \theta (z_2 - z_1) \right]^t \left[z_1 + \theta (z_2 - z_1) \right] = r^2$$

$$\theta^2 \|z_2 - z_1\|^2 = r^2 - \|z_1\|^2$$

$$\text{Se } z_2 - z_1 = 0 \rightarrow \|z_1\|^2 = r^2$$

Supondo $z_2 - z_1 \neq 0$:

$$\theta^2 = \frac{r^2 - \|z_1\|^2}{\|z_2 - z_1\|^2}$$

Como o "hard-case" só ocorre quando $\varphi(-\lambda_1) \leq r^2$, temos

$$r^2 \geq \|z_1\|^2 \rightarrow \theta = \pm \sqrt{\frac{r^2 - \|z_1\|^2}{\|z_2 - z_1\|^2}}$$

Analogamente à esfera de confiança, o parâmetro

$$\theta = \pm \sqrt{\frac{r^2 - \|z_1\|^2}{\|z_2 - z_1\|^2}} \quad (3.12) \text{ será o escolhido pois}$$

$$\begin{aligned} \|z_2 - z(\theta+)\| &= \|(1 - \theta)(z_2 - z_1)\| \leq \\ &\leq \|(1 + \theta)(z_2 - z_1)\| = \|z_2 - z(\theta-)\| \end{aligned}$$

$$\text{Logo } \boxed{s_{s. \text{ prob.}} = z_1 + \theta (z_2 - z_1)} \quad (3.13)$$

com z_1 , z_2 e θ dados respectivamente por (3.10), (3.11) e (3.12).

3.4.3 O "hard case" na $(n - 2)$ esfera

Neste caso, chamando de S_2 o autoespaço associado ao menor autovalor $(\tilde{\lambda}_1)$ de $A^t B_k A$ (de dimensão = $\dim 2$), tem-se o "hard-case" sempre que $\tilde{g}_k \in S_2^\perp$, $\rho(-\tilde{\lambda}_1) \leq \rho^2$, com $\tilde{g}_k$ e ρ dados por (3.4.15) e (3.4.7), respectivamente.

$$\text{Chamando } \boxed{y_1 = -(A^t B_k A - \tilde{\lambda}_1 I)^+ \tilde{g}_k} \quad (3.14), \text{ tomamos}$$

$$\begin{aligned} y(\theta) &= y_1 + \theta \tilde{q}_1 \quad (\tilde{q}_1 \in S_2) \text{ e impomos } \|y(\theta)\|^2 = \rho^2 \rightarrow \\ &\rightarrow (y_1 + \theta \tilde{q}_1)^t (y_1 + \theta \tilde{q}_1) = \rho^2 \rightarrow \\ &\rightarrow \|y_1\|^2 + 2\theta \tilde{q}_1^t y_1 + \theta^2 \tilde{q}_1^t \tilde{q}_1 = \rho^2 \rightarrow \theta^2 = \rho^2 - \|y_1\|^2 \end{aligned}$$

Como o "hard-case" só ocorre quando $\rho(-\tilde{\lambda}_1) \leq \rho^2$, temos

$$\rho^2 \geq \|y_1\|^2 \rightarrow \boxed{\theta = \pm \sqrt{\rho^2 - \|y_1\|^2}} \quad (3.15)$$

$$\text{e escolhemos } \theta = \sqrt{\rho^2 - \|y_1\|^2} \rightarrow$$

$$\rightarrow \boxed{s_{(n-2)s.f.} = A(y_1 + \theta \tilde{q}_1) + \beta \tilde{x}_k} \quad (3.16)$$

conforme (3.4.12), com y_1 e θ dados respectivamente por (3.14) e (3.15)

Visualizando:

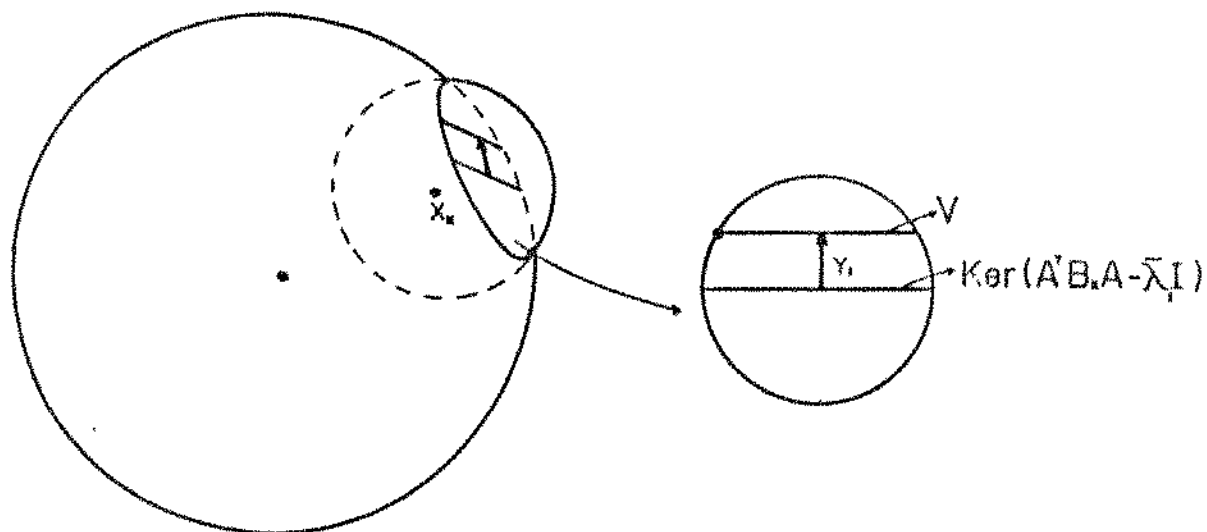


figura 3.9

CAPÍTULO 4

EXPERIMENTOS NUMÉRICOS

4.1 CARACTERÍSTICAS DO PROGRAMA

Para verificarmos o desempenho dos métodos de região de confiança para minimização em bolas, fizemos um programa escrito em FORTRAN 77 e implementado no VAX 11/785 - sistema operacional VMS, da UNICAMP, usando precisão dupla (ver documentação no APÊNDICE).

O algoritmo implementado no programa principal (TRUST) foi:

- 0) $k = 0$, definir Δ_0
- 1) Dados x_k e Δ_k , calcular $g_k = \nabla f(x_k)$ e $B_k \approx \nabla^2 f(x_k)$
- 2) Resolver

$$\begin{array}{l} \text{minimizar } \Psi_k(w) = w^t g_k + \frac{1}{2} w^t B_k w \\ \text{s/a} \quad \quad \quad \| x_k + w \| \leq r \\ \quad \quad \quad \| w \| \leq \Delta_k \end{array} \quad (4.1)$$

obtendo s_k

- 3) Se $f(x_k + s_k) < f(x_k) + 10^{-4} \Psi_k(s_k)$
então definir $x_{k+1} = x_k + s_k$
 $\Delta_{k+1} = \Delta_0$
 $k = k + 1$ e ir para (1)
- senão $\Delta_k = 0.5 \| s_k \|$ e ir para (2)

Escolhemos as matrizes B_k como sendo aproximações para as matrizes Hessianas $\nabla^2 f(x_k)$ via diferenças finitas avançadas aplicadas ao vetor gradiente, conforme sugere [1], p. 102.

Na resolução do subproblema (4.1) através da rotina SUBPRB (implementação do algoritmo III), utilizamos a rotina F02ABF da NAG (precisão dupla) para efetuar a decomposição espectral das matrizes B_k . Esta rotina calcula todos os autovalores e autovetores de matrizes reais simétricas, aplicando o algoritmo QL após tridiagonalizar a matriz via transformações de Householder. Uma vez obtida a decomposição espectral completa de B_k , não precisamos resolver sistemas lineares para encontrar as direções procuradas, mas apenas empregar multiplicações de matrizes por vetores. A rotina SUBPRB foi inicialmente validada com testes utilizando funções quadráticas.

A factibilidade das soluções obtidas em SUBPRB foi verificada através de testes relativos. Por exemplo, após obter uma direção s_k para a qual $x_k + s_k$ é uma solução global na esfera do problema, o teste que permite decidir sobre a factibilidade de s_k na bola de confiança é:

$$\frac{\|s_k\| - \Delta_k}{\|s_k\|} \leq \epsilon_{\text{conf}} \quad (4.2)$$

Os critérios de parada utilizados em TRUST foram:

(i) convergência pela norma da direção:

$$\|s_k\|_2 \leq \epsilon_{\text{conf}} \quad (4.3)$$

(ii) convergência pela norma do gradiente (para solução irrestrita, na esfera de confiança ou quando $B_k \geq 0$):

$$\| \epsilon_k \|_2 \leq \epsilon_{\text{grad}} \quad (4.4)$$

(iii) não há convergência após Kmax iterações .

4.2 ANÁLISE DO DESEMPENHO COMPUTACIONAL

4.2.1 Testes realizados

Trabalhamos com o conjunto de testes a seguir, sugerido em [15] para minimização irrestrita em problemas de quadrados mínimos não lineares, onde a função objetivo é dada por:

$$\begin{aligned} f : \mathbb{R}^N &\longrightarrow \mathbb{R} \\ f(x) &= \sum_{i=1}^M f_i^2(x) \end{aligned} \quad (4.5)$$

(1) Função de Rosenbrock

$$N = 2, M = 2$$

$$f_1(x) = 10 (x_2 - x_1^2)$$

$$f_2(x) = 1 - x_1$$

$$f = 0 \text{ em } (1, 1)^t$$

(2) Função de Freudenstein e Roth

$$N = 2, M = 2$$

$$f_1(x) = -13 + x_1 + ((5 - x_2) x_2 - 2) x_2$$

$$f_2(x) = -29 + x_1 + ((x_2 + 1) x_2 - 14) x_2$$

$$f = 0 \text{ em } (5, 4)^t$$

$$f = 48.9842... \text{ em } (11.41..., -0.8968...)^\top$$

(3) "Helical valley function"

$$N = 3, M = 3$$

$$f_1(x) = 10 \left[x_3 - 10 \theta(x_1, x_2) \right]$$

$$f_2(x) = 10 \left[(x_1^2 + x_2^2)^{1/2} - 1 \right]$$

$$f_3(x) = x_3$$

$$\text{onde } \theta(x_1, x_2) = \begin{cases} \frac{1}{2\pi} \arctan \left(\frac{x_2}{x_1} \right) & , \quad \text{se } x_1 > 0 \\ \frac{1}{2\pi} \arctan \left(\frac{x_2}{x_1} \right) + 0.5 & \text{se } x_1 < 0 \end{cases}$$

$$f = 0 \text{ em } (1, 0, 0)^\top$$

(4) Função EXP6 de Biggs

$$N = 6, M \geq N$$

$$f_i(x) = x_3 \exp[-t_i x_1] - x_4 \exp[-t_i x_2] \\ + x_6 \exp[-t_i x_5] - y_i$$

$$\text{onde } t_i = (0.1)^i$$

$$\text{e } y_i = \exp[-t_i] - 5 \exp[-10 t_i] + 3 \exp[-4 t_i]$$

$$f = 0 \text{ em } (1, 10, 1, 5, 4, 3)^\top,$$

$$(4, 10, 3, 5, 1, 1)^\top \text{ e}$$

$$(4, 1, 3, -1, 10, -5)^\top$$

(5) Função Gaussiana

$$N = 3, M = 15$$

$$f_i(x) = x_1 \exp \left[\frac{-x_2(t_i - x_3)^2}{2} \right] - y_i$$

onde $t_i = (8 - i)/2$

e	i	y_i
	1, 15	0.0009
	2, 14	0.0044
	3, 13	0.0175
	4, 12	0.0540
	5, 11	0.1295
	6, 10	0.2420
	7, 9	0.3521
	8	0.3989

$$f = 1.12793... 10^{-8}$$

(6) Função mal escalada de Powell

$$N = 2, M = 2$$

$$f_1(x) = 10^4 x_1 x_2 - 1$$

$$f_2(x) = \exp[-x_1] + \exp[-x_2] - 1.0001$$

$$f = 0 \quad \text{em} \quad (1.098...10^{-5}, 9.106...)^t$$

(7) Função da caixa tridimensional

$$N = 3, M \geq N$$

$$f_i(x) = \exp[-t_i x_1] - \exp[-t_i x_2] \\ - x_3 (\exp[-t_i] - \exp[-10 t_i])$$

$$\text{onde } t_i = (0.1)^i$$

$$f = 0 \quad \text{em} \quad (1, 10, 1)^t, (10, 1, -1)^t, (k, k, 0)^t$$

(8) Função variavelmente dimensionada

$$N \text{ variável}, M = N + 2$$

$$f_i(x) = x_i - 1, \quad i = 1, \dots, N$$

$$f_{N+1}(x) = \sum_{j=1}^N j (x_j - 1)$$

$$f_{N+2}(x) = \left[\sum_{j=1}^N j (x_j - 1) \right]^2$$

$$f = 0 \text{ em } (1, \dots, 1)^t$$

(9) Função de Watson

$$2 \leq N \leq 31, M = 31$$

$$f_i(x) = \sum_{j=2}^N (j-1) x_j t_i^{j-2} - \left[\sum_{j=1}^N x_j t_i^{j-1} \right]^2 - 1$$

$$\text{onde } t_i = i/29, \quad 1 \leq i \leq 29$$

$$f_{90}(x) = x_1$$

$$f_{91}(x) = x_2 - x_1^2 - 1$$

$$f = 2.28767...10^{-9} \quad \text{se } N = 6$$

$$f = 1.39976...10^{-6} \quad \text{se } N = 9$$

$$f = 4.72238...10^{-10} \quad \text{se } N = 12$$

(10) Função penalidade I

$$N \text{ variável, } M = N + 1$$

$$f_i(x) = 10^{-5/2} (x_i - 1), \quad 1 \leq i \leq N$$

$$f_{N+1}(x) = \left[\sum_{j=1}^N x_j^2 \right] - \frac{1}{4}$$

$$f = 2.24997...10^{-5} \quad \text{se } N = 4$$

$$f = 7.08765...10^{-5} \quad \text{se } N = 10$$

(11) Função penalidade II

$$N \text{ variável, } M = 2N$$

$$f_1(x) = x_1 - 0.2$$

$$f_i(x) = 10^{-5/2} \left[\exp \left[\frac{x_i}{10} \right] + \exp \left[\frac{x_{i-1}}{10} \right] - y_i \right],$$

$$2 \leq i \leq N$$

$$f_i(x) = 10^{-5/2} \left[\exp \left[\frac{x_{i-N+1}}{10} \right] - \exp \left[\frac{-1}{10} \right] \right],$$

$$N \leq i \leq 2N$$

$$f_{2N}(x) = \left[\sum_{j=1}^N (N - j + 1) x_j^2 \right] - 1$$

$$\text{onde } y_i = \exp \left[\frac{i}{10} \right] + \exp \left[\frac{i-1}{10} \right]$$

$$f = 9.37629...10^{-6} \quad \text{se } N = 4$$

$$f = 2.93660...10^{-4} \quad \text{se } N = 10$$

(12) Função mal escalada de Brown

$$N = 2, M = 3$$

$$f_1(x) = x_1 - 10^{-6}$$

$$f_2(x) = x_2 - 2 \cdot 10^{-6}$$

$$f_3(x) = x_1 x_2 - 2$$

$$f = 0 \quad \text{em } (10^6, 2 \cdot 10^{-6})^t$$

(13) Função de Brown e Dennis

$$N = 4, M \geq N$$

$$f_i(x) = \left[x_1 + t_i x_2 - \exp [t_i] \right]^2 +$$

$$+ \left[x_3 + x_4 \sin (t_i) - \cos (t_i) \right]^2$$

$$\text{onde } t_i = i/5$$

$$f = 85822.2... \text{ se } M = 20$$

(14) "Gulf research and development function"

$$N = 3, N \leq M \leq 100$$

$$f_i(x) = \exp \left[- \frac{|y_i M + x_2|^{x_3}}{x_1} \right] - t_i$$

$$\text{onde } t_i = i/100 \quad \text{e} \quad y_i = 25 + \left[-50 \ln(t_i) \right]^{2/3}$$

$$f = 0 \quad \text{em} \quad (50, 25, 1.5)^t$$

(15) Função trigonométrica transladada

$$N \text{ variável}, M = N$$

$$f_i(x) = N - \sum_{j=1}^N \cos(x_j - 1) + \\ + i \left[1 - \cos(x_i - 1) \right] - \sin(x_i - 1)$$

$$f = 0 \quad \text{em} \quad (1, 1, \dots, 1)^t$$

(16) Função de Rosenbrock estendida

$$N \text{ par}, M = N$$

$$f_{2i-1}(x) = 10(x_{2i} - x_{2i-1}^2)$$

$$f_{2i}(x) = 1 - x_{2i-1}$$

$$f = 0 \quad \text{em} \quad (1, \dots, 1)^t$$

(17) Função singular de Powell estendida transladada

$$N \text{ múltiplo de } 4, M = N$$

$$f_{4i-3}(x) = x_{4i-3} + 10x_{4i-2} + 11$$

$$f_{4i-2}(x) = 5^{1/2}(x_{4i-1} - x_{4i})$$

$$f_{4i-1}(x) = (x_{4i-2} - 2x_{4i-1} - 1)^2$$

$$f_{4i}(x) = 10^{1/2} (x_{4i-3} - x_{4i})^2$$

$$f = 0 \quad \text{em} \quad (-1, \dots, -1)^t$$

(18) Função de Beale

$$N = 2, M = 3$$

$$f_i(x) = y_i - x_i (1 - x_2^i)$$

$$\text{onde } y_1 = 1.5, y_2 = 2.25 \quad \text{e} \quad y_3 = 2.625$$

$$f = 0 \quad \text{em} \quad (3, 0.5)^t$$

(19) Função de Wood

$$N = 4, M = 6$$

$$f_1(x) = 10 (x_2 - x_1^2)$$

$$f_2(x) = 1 - x_1$$

$$f_3(x) = 90^{1/2} (x_4 - x_3^2)$$

$$f_4(x) = 1 - x_3$$

$$f_5(x) = 10^{1/2} (x_2 + x_4 - 2)$$

$$f_6(x) = 10^{-1/2} (x_2 - x_4)$$

$$f = 0 \quad \text{em} \quad (1, 1, 1, 1)^t$$

Uma vez definida a função objetivo ((1) a (19)), formulamos dois tipos básicos de problema através de restrições com bolas de raio "grande" ou "pequeno". Como as funções-teste escolhidas têm, em geral, minimizador irrestrito (x^*) conhecido, procuramos escolher o tamanho do raio pelo seguinte princípio:

$$(i) \quad \text{raio grande} = 4 \quad \|x^*\|_2$$

$$(ii) \quad \text{raio pequeno} = 0.25 \quad \|x^*\|_2$$

Para os casos em que x^* não é previamente conhecido, tomamos os valores para os raios conforme apresentado na tabela 4.1.

n ^o do teste	N	M	raio grande	raio pequeno
01	2	2	5.7	0.35
02	2	2	25.6	1.6
03	3	3	4.0	0.25
04	6	13	15.0	0.125
05	3	15	2.0	0.125
06	2	2	10.0	2.5
07	3	10	4.0	0.25
08	10	12	12.65	0.79
09	6	31	4.0	0.25
10	4	5	4.0	0.25
11	4	8	4.0	0.25
12	2	3	10 ⁶	10.0
13	4	20	20.0	4.0
14	3	10	1.0	0.25
15	10	10	12.0	1.0
16	10	10	12.65	0.79
17	12	12	8.0	0.5
18	2	3	12.16	0.76
19	4	6	8.0	0.5

tabela 4.1 - Dimensões e raios efetivamente adotados

Os problemas foram testados tendo-se duas escolhas possíveis para o ponto inicial:

- (i) interior da bola (origem, sempre que possível)
- (ii) fronteira da bola, fixando:

$$x_0 = \frac{r}{\sqrt{N}} \left[1, -1, 1, \dots, (-1)^{N+1} \right] \in \mathbb{R}^N$$

Com exceção dos testes com a função EXP6 de Biggs (4), restrita à bola de raio grande, utilizamos $\Delta_0 = 2r$, isto é,

tomamos o raio de confiança inicial como sendo o diâmetro da bola do problema. Nos testes com (4) tomamos $\Delta_0 = 0.5$.

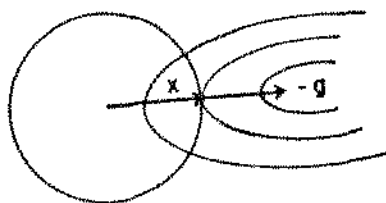
Adotamos as seguintes constantes de tolerância:

ϵ_{conf}	$= 10^{-4}$	(relativa à esfera de confiança)
ϵ_{prob}	$= 10^{-5}$	(relativa à esfera do problema)
$\epsilon_{(n-2)esf}$	$= 10^{-6}$	(relativa à (n-2) esfera)
ϵ_{hard}	$= 10^{-7}$	(relativa ao "hard case")
ϵ_{grad}	$= 10^{-8}$	(relativa ao vetor gradiente)
Kmax	$= 150$	

Obs: Para a função (11) restrita à bola de raio grande, utilizamos $\epsilon_{grad} = 10^{-4}$.

4.2.2. Resultados Numéricos

Setenta (70) dos setenta e seis (76) problemas testados satisfizeram as expectativas iniciais, o que corresponde a um índice de 92%. Dos trinta e oito (38) problemas restritos a bolas de raio grande, obteve-se solução do tipo minimizador irrestrito em trinta e cinco (35) deles. Ou seja, a restrição não interferiu nos resultados obtidos em 92 % dos casos. Para as restrições do tipo bolas de raio pequeno, dos trinta e oito (38) problemas testados, foram encontrados minimizadores na esfera do problema satisfazendo as condições de Kuhn-Tucker (figura 4.1) em trinta e cinco (35) deles.



$$x = -\lambda g, \lambda \geq 0.$$

figura 4.1

Nos gráficos 4.1 e 4.2 podemos ver o tipo de solução obtida, conforme o tipo de restrição do problema e o ponto inicial escolhido.

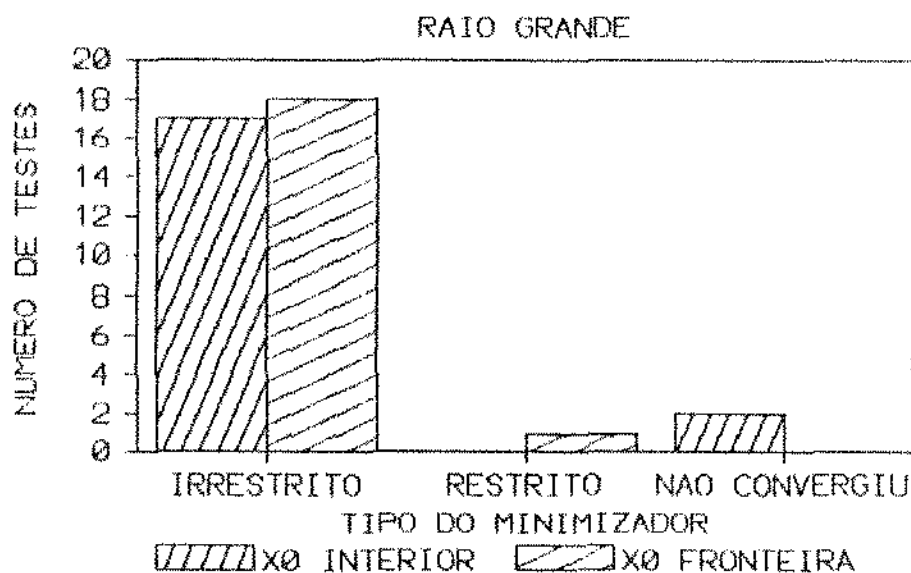


gráfico 4.1

Cabe observar que no problema formado pela função (3) ("helical valley function") restrita à bola de raio pequeno, para os dois tipos de pontos iniciais, a sequência gerada pelo algoritmo implementado convergiu para pontos nos quais não há diferenciabilidade da função objetivo.

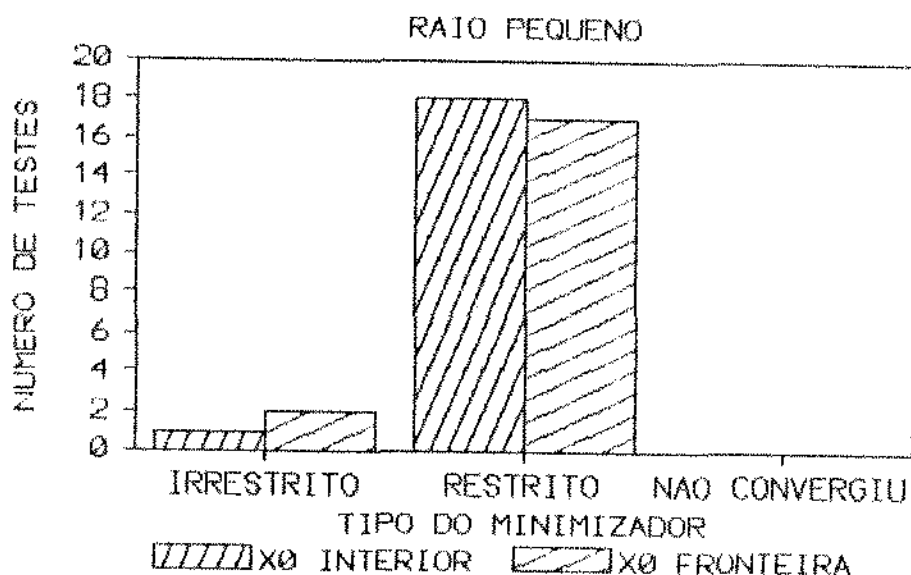


gráfico 4.2

Com relação ao critério de convergência, para as restrições do tipo bola de raio pequeno, o método sempre convergiu pela norma da direção. Para os problemas restritos a bolas de raio grande, o número de iterações atingiu o limite máximo em dois problemas; houve cinco casos de convergência pela norma do gradiente e os demais convergiram pela norma da direção. Além destes resultados, pode-se ver no gráfico 4.3 a influência do ponto inicial escolhido.

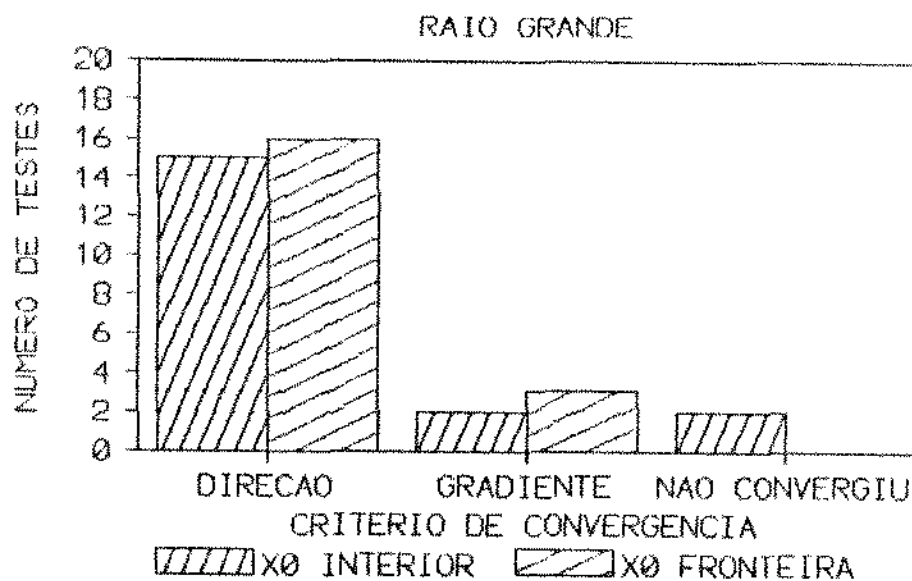


gráfico 4.3

Apresentamos um resumo dos resultados obtidos na tabela 4.2. Para observarmos o comportamento do algoritmo, visualizando a sequência de pontos gerada, escolhemos os problemas com a função de Rosenbrock (1) (figuras 4.2 - 4.6).

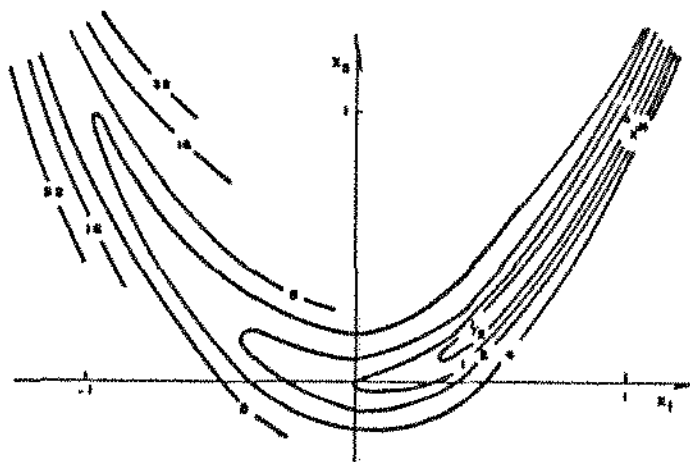


figura 4.2 - Curvas de nível da função de Rosenbrock

FO	raio grande								raio pequeno							
	x_0 interior				x_0 fronteira				x_0 interior				x_0 fronteira			
	TS	CP	IT	NAF	TS	CP	IT	NAF	TS	CP	IT	NAF	TS	CP	IT	NAF
01	I	D	14	20	I	D	19	24	RG	D	5	7	RG	D	4	5
02	I	D	6	13	I	D	16	17	RG	D	5	6	RG	D	4	5
03	I	D	11	26	I	D	11	19	RG	D	2	6	RG ₂	D	5	10
04	#			608	I	D	106	238	RL	D	9	14	RG ₂	D	7	17
05	I	D	8	14	I	D	14	28	RG	D	5	6	RG	D	3	4
06	#			907	I	D	38	54	RG	D	20	56	I	D	34	91
07	I	D	8	25	I	D	19	128	I	D	4	5	I	D	4	5
08	I	D	19	87	I	D	19	76	RG	D	2	3	RG	D	2	3
09	I	D	12	14	I	D	8	11	RG	D	4	5	RG	D	4	5
10	I	D	2	6	I	D	50	73	RG	D	2	3	RG ₂	D	9	18
11	I	G	9	17	I	G	26	27	RG	D	2	3	RG	D	3	4
12	I	D	21	127	I	D	30	119	RG	D	2	3	RG	D	2	3
13	I	D	7	12	I	D	9	10	RG	D	4	5	RG	D	4	5
14	I	D	16	46	I	G	1	2	RG	D	11	16	RG	D	8	12
15	I	D	11	12	RG	D	10	12	RG	D	3	4	RG ₂	D	3	5
16	I	D	14	20	I	D	19	23	RG	D	5	7	RG	D	4	5
17	I	G	23	25	I	G	24	27	RG	D	3	4	RG	D	4	5
18	I	D	8	13	I	D	15	34	RG	D	4	6	RG	D	4	5
19	I	D	8	11	I	D	11	12	RG	D	4	5	RG	D	3	10

tabela 4.2 - Resumo dos resultados obtidos

legenda:

FO - função objetivo

TS - tipo de solução

I - irrestrita

RG - minimizador global na esfera do problema

RL - minimizador local na esfera do problema

RG₂- minimizador global na (n-2) esfera

CP - critério de parada

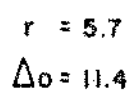
D - norma da direção

G - norma do gradiente

- não convergiu com Kmax iterações

IT - número de iterações efetuadas

NAF - número de vezes que a função foi avaliada



78

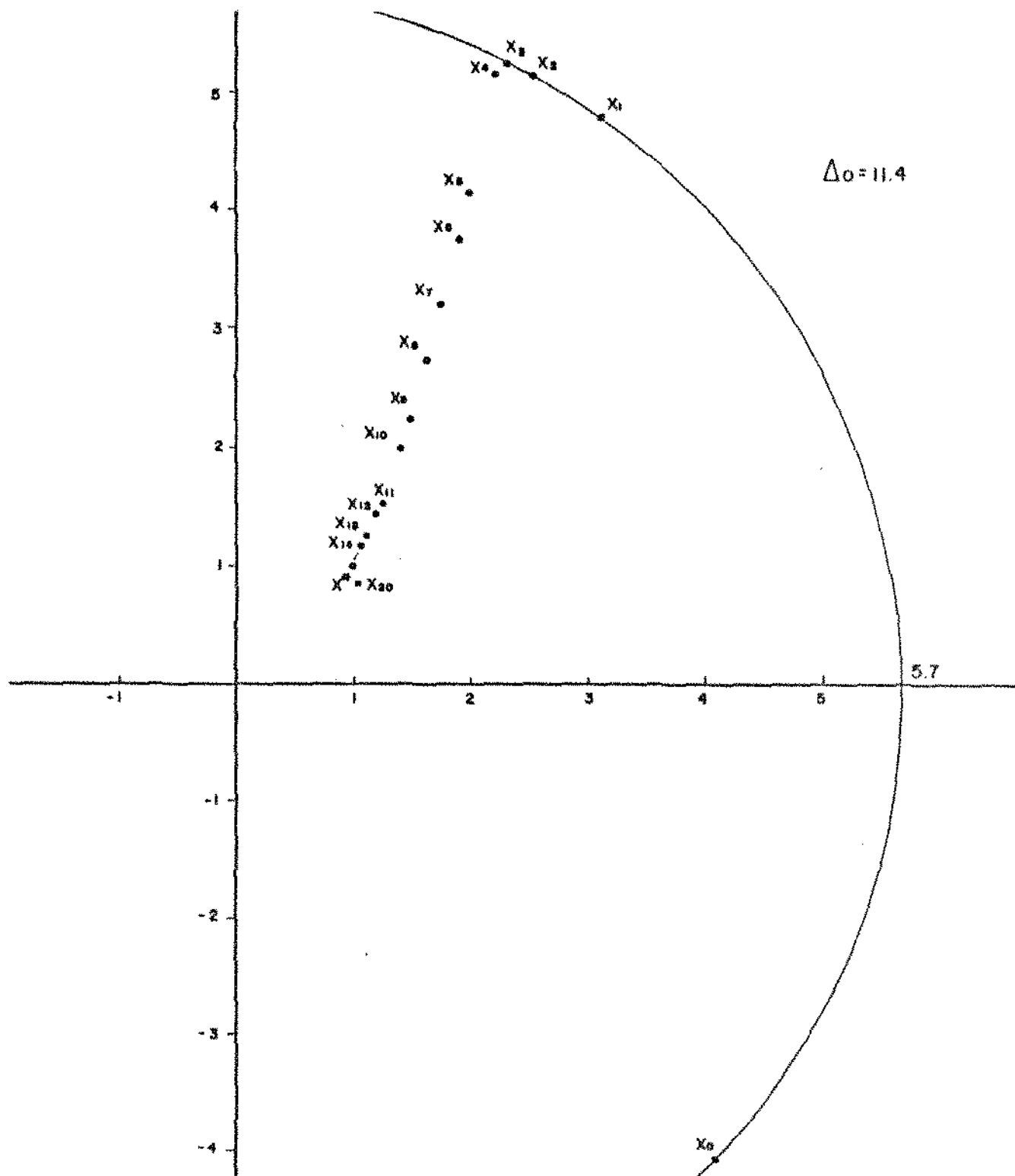


figura 4.4

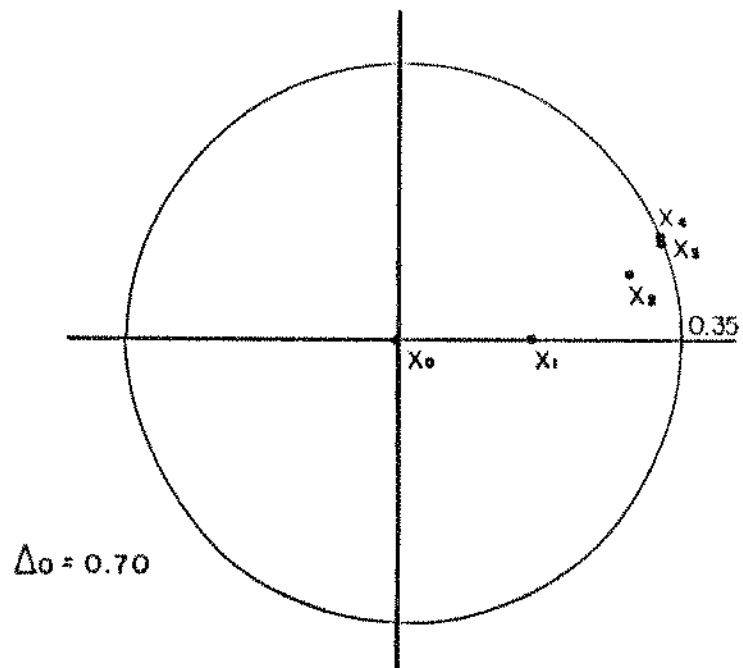


figura 4.5

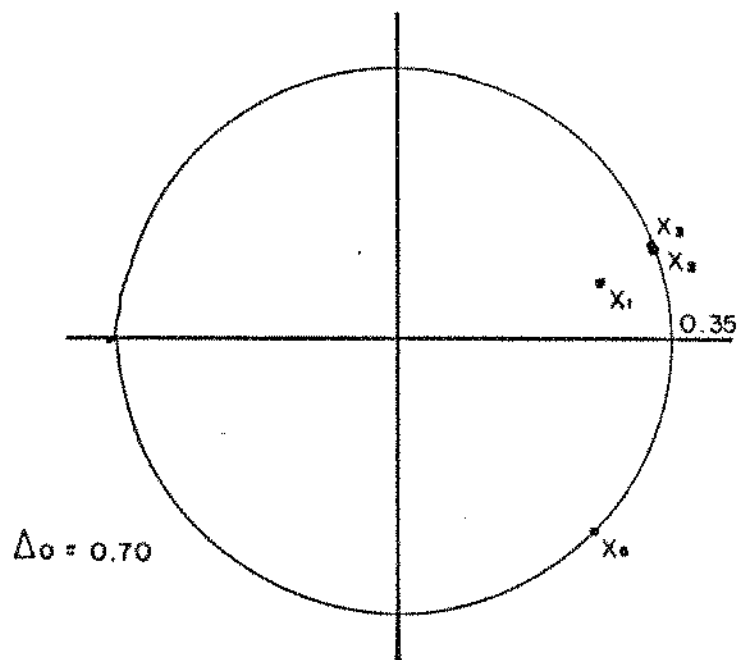


figura 4.6

4.3 CONCLUSÕES E TRABALHOS FUTUROS

Analisando os resultados numéricos obtidos, observamos que o método de região de confiança para minimização em bolas implementado de fato cumpre o papel globalizador para o qual foi proposto. Além disso, acreditamos que o desempenho obtido foi melhor do que se poderia esperar para algoritmos gerais, ainda que não tenhamos feito testes comparativos. O fato da nossa implementação estar baseada na decomposição espectral completa das matrizes B_k inviabiliza a resolução de problemas de maior dimensão. Tal limitação, no entanto, não desmerece nosso trabalho, já que funções caras, tipo ajustes não lineares, com poucos parâmetros, existem na realidade.

Motivados pelo problema de identificação de parâmetros, pretendemos prosseguir trabalhando com métodos deste tipo, com o objetivo de desenvolver algoritmos que sejam viáveis para problemas de grande porte.

APÊNDICE

Apresentamos a seguir a listagem documentada do programa principal e das rotinas que compõem nossa implementação computacional.

```

C
C RESOLUCAO DO PROBLEMA DE MINIMIZACAO RESTRITO A UMA BOLA
C EUCLIDIANA, VIA METODO DE REGIAO DE CONFIANCA, PARA FUNCOES
C TESTE DE MORE, GARBOW & HILLSTROM
C
C
C
C
C REAL*8 RCON, RPRB, RMAX, QSK, FXK, FXK1,
C $ XK(50), XK1(50), SK(50), GK(50), BK(50,50),
C $ E(50), F(50), QK(50,50), RK(50), W(50), C(50), U(50),
C $ A(50,50), B2(50,50), Q2(50,50), R2(50), WC(50), GC(50),
C $ CC(50), UC(50), ENXK, ENG, SI(50), SGEP(50), SLEP(50),
C $ MGEC, MGEP, EPSHRD, EPSCON, EPSPRB, EPS2ES, EMACH,
C $ EPSGRD, A1, A2, H, REPSGRD, ENSK, ENPRB
C
C RCON = raio de confianca
C RPRB = raio do problema
C RMAX = raio de confianca maximo (2.0 * RPRB)
C QSK = valor da quadratica na solucao do k-esimo subproblema
C FXK = valor da funcao objetivo no ponto XK
C FXK1 = valor da funcao objetivo no ponto XK + SK
C XK = k-esimo ponto
C XK1 = ponto XK + SK
C SK = direcao solucao do k-esimo subproblema
C GK = vetor gradiente da funcao objetivo no ponto XK
C BK = aproximacao para a matriz hessiana da funcao objetivo em XK
C E, F = vetores auxiliares
C QK = matriz ortogonal (colunas = autovetores de BK)
C RK = vetor dos autovalores de BK (ordem crescente)
C W = vetor BK XK
C C = vetor (QK)t GK
C U = vetor (QK)t XK
C A = matriz cujas colunas formam base ortonormal
C para o complemento ortogonal de XK
C B2 = produto At * BK * A (hessiana da quadratica na mudanca de
C variaveis para a (n-2) esfera
C Q2 = matriz ortogonal (colunas = autovetores de B2)
C R2 = vetor dos autovalores de B2 (ordem crescente)
C WC = vetor (A)t W
C GC = vetor (A)t GK
C CC = vetor (Q2)t GC
C UC = vetor (Q2)t WC
C ENXK = norma euclidiana do vetor XK
C ENG = norma euclidiana do vetor GK
C SI = solucao irrestrita
C SGEP = solucao global na esfera do problema
C SLEP = solucao local nao global na esfera do problema
C MGEC = multiplicador de Lagrange correspondente a SGEC
C MGEP = idem para SGEP
C EPSHRD = precisao para detectar hard case
C EPSCON = precisao para resolver na esfera de confianca
C EPSPRB = precisao para resolver na esfera do problema
C EPS2ES = precisao para resolver na (n-2)-esfera
C EMACH = precisao da maquina
C EPSGRD = precisao para o vetor gradiente
C A1, A2 = variaveis auxiliares

```

```

C      H = incremento usado para aproximar a hessiana via diferenca
C      finitas avancadas
C      REPSGRD = raiz quadrada de EPSGRD
C      ENSK = norma euclidiana da direcao SK
C      ENPRB = norma do candidato a solucao para testar
C      factibilidade na bola do problema
C
C      INTEGER IPRB, IRAI, IPIN, ISOL, N, NF, I, J, K, NAVAL,
$ DIM, DIM2, MAXNEW, MAXL1, MAXL2, KMAX
C
C      IPRB = indice do problema teste
C      IRAI = codigo para o tipo de raio
C      ( 1 -> grande , 2 -> pequeno )
C      IPIN = codigo para o tipo de ponto inicial
C      ( 1 -> interior , 2 -> fronteira )
C      ISOL = codigo para o tipo de solucao do subproblema
C      0 -> subproblema sem solucao
C      1 -> solucao global na esfera de confianca
C      2 -> solucao local na esfera de confianca
C      3 -> solucao global na esfera do problema
C      4 -> solucao local na esfera do problema
C      5 -> solucao global na (n-2)-esfera
C      6 -> solucao irrestrita
C      N = dimensao do problema teste
C      NF = dimensao do vetor de funcoes cujo quadrado da norma
C      euclidiana constitui a funcao objetivo do problema
C      I, J = indices dos vetores
C      K = contador de iteracoes do algoritmo
C      NAVAL = contador de avaliacoes da funcao
C      DIM = dimensao do autoespaco associado ao menor autovalor de BK
C      DIM2 = idem para o menor autovalor de B2
C      MAXNEW = numero maximo de iteracoes para NEWTON
C      MAXL1 = numero maximo de iteracoes para a fase 1 de MINLOC
C      MAXL2 = idem para a fase 2 de MINLOC
C      KMAX = numero maximo de iteracoes para o algoritmo de regioao
C      de confianca
C
C      LOGICAL SAMEX, FSI, FGEP, FLEP, FMEC, FMEP, DEZE,
$ ORTO1, FALHA, HARDPB
C
C      SAMEX = indica resolucao do subproblema com o mesmo ponto XK
C      e diferente raio de confianca
C      FSI = factibilidade da solucao irrestrita
C      FGEP = factibilidade da sol. glob. na esf. problema
C      FLEP = factibilidade da sol. loc. na esf. problema
C      DEZE = marca decomposicao espectral na (n-2) esfera
C      FMEC = existe multiplicador global na esfera de confianca
C      mas a solucao e infactivel
C      FMEP = idem na esfera do problema
C      ORTO1 = GK ortogonal a S1 (= autoespaco associado ao menor
C      autovalor de BK)
C      FALHA = codigo de erro na avaliacao da funcao ou do gradiente
C      ( .TRUE. -> OK , .FALSE. -> ponto fora do dominio
C      HARDPB = ocorrencia de "hard-case" na esfera do problema
C

```

```

C      FUNCOES UTILIZADAS
C
C      DOUBLE PRECISION EUCNOR
C      DOUBLE PRECISION FUNC
C
C      SUBROTINAS UTILIZADAS
C
C      GKBK
C      SUBPRB
C
C      ENTRADA DE DADOS/INICIALIZACAO
C
      OPEN (UNIT = 8, FILE = 'TRUST.OUT', STATUS = 'NEW')
      WRITE (*,10010)
      WRITE (8,10010)
      WRITE (*,10020)
      ACCEPT *,IPRB
      IF ((IPRB.GE.1).AND.(IPRB.LE.19)) THEN
        WRITE (*,10030)
        ACCEPT *,IRAI
        IF ((IRAI.EQ.1).OR.(IRAI.EQ.2)) THEN
          WRITE (*,10040)
          ACCEPT *,IPIN
          IF ((IPIN.EQ.1).OR.(IPIN.EQ.2)) THEN
            EMACH = 1.0D-16
            EPSGRD = 1.0D-8
            EPSHRD = 1.0D-7
            EPSCON = 1.0D-4
            EPSPRB = 1.0D-5
            EPS2ES = 1.0D-6
            MAXNEW = 50
            MAXL1 = 50
            MAXL2 = 150
            KMAX = 150
            REPSGRD = DSQRT(EPSGRD)
C
C            GOTO (10,20,30,40,50,60,70,80,90,100,110,120,130,140,
C              $      150,160,170,180,190) IPRB
C            GOTO 200
10          CONTINUE
            WRITE (*,10090)
            WRITE (8,10090)
            N = 2
            NF = 2
            GOTO 200
20          CONTINUE
            WRITE (*,10100)
            WRITE (8,10100)
            N = 2
            NF = 2
            GOTO 200
30          CONTINUE
            WRITE (*,10110)
            WRITE (8,10110)
            N = 3

```

```

        NF = 3
        GOTO 200
40      CONTINUE
        WRITE (*,10120)
        WRITE (8,10120)
        N = 6
        NF = 13
        GOTO 200
50      CONTINUE
        WRITE (*,10130)
        WRITE (8,10130)
        N = 3
        NF = 15
        GOTO 200
60      CONTINUE
        WRITE (*,10140)
        WRITE (8,10140)
        N = 2
        NF = 2
        GOTO 200
70      CONTINUE
        WRITE (*,10150)
        WRITE (8,10150)
        N = 3
        NF = 10
        GOTO 200
80      CONTINUE
        WRITE (*,10160)
        WRITE (8,10160)
        N = 10
        NF = N + 2
        GOTO 200
90      CONTINUE
        WRITE (*,10170)
        WRITE (8,10170)
        N = 6
        NF = 31
        GOTO 200
100     CONTINUE
        WRITE (*,10180)
        WRITE (8,10180)
        N = 4
        NF = N + 1
        GOTO 200
110     CONTINUE
        WRITE (*,10190)
        WRITE (8,10190)
        N = 4
        NF = 2 * N
        GOTO 200
120     CONTINUE
        WRITE (*,10200)
        WRITE (8,10200)
        N = 2
        NF = 3

```

```

        GOTO 200
130    CONTINUE
        WRITE (*,10210)
        WRITE (8,10210)
        N = 4
        NF = 20
        GOTO 200
140    CONTINUE
        WRITE (*,10220)
        WRITE (8,10220)
        N = 3
        NF = 10
        GOTO 200
150    CONTINUE
        WRITE (*,10230)
        WRITE (8,10230)
        N = 10
        NF = N
        GOTO 200
160    CONTINUE
        WRITE (*,10240)
        WRITE (8,10240)
        N = 10
        NF = N
        GOTO 200
170    CONTINUE
        WRITE (*,10250)
        WRITE (8,10250)
        N = 12
        NF = N
        GOTO 200
180    CONTINUE
        WRITE (*,10260)
        WRITE (8,10260)
        N = 2
        NF = 3
        GOTO 200
190    CONTINUE
        WRITE (*,10270)
        WRITE (8,10270)
        N = 4
        NF = 6
200    CONTINUE
C
        WRITE(*,*)'RAIO DO PROBLEMA:'
        ACCEPT *,RPRB
C
C    PONTO INICIAL
C
        IF (IPIN.EQ.1) THEN
C
C    PONTO NO INTERIOR DA BOLA
C    (ORIGEM OU PEQUENA VARIACAO DEVIDO AO DOMINIO DA F.O.)
C
        WRITE(*,*)'PONTO INICIAL:'

```

```

      DO I = 1,N
        WRITE(*,*)'XK(',I,') -> '
        ACCEPT *,XK(I)
      ENDDO
    ELSE

```

```

C
C PUNTO NA FRONTEIRA DA BOLA
C

```

```

      A1 = RPRB / DSQRT(1.0D+0 * N)
      DO I = 1,N
        XK(I) = ((-1.0D0) ** (I+1)) * A1
      END DO
    ENDIF
    WRITE (8,10280) RPRB
    IF (IRAI.EQ.1) THEN
      WRITE (8,10290)
    ELSE
      WRITE (8,10300)
    ENDIF
    IF (IFIN.EQ.1) THEN
      WRITE (8,10310)
    ELSE
      WRITE (8,10320)
    ENDIF

```

```

C
C RAIO DE CONFIANCA MAXIMO
C

```

```

      WRITE(*,*)'RAIO DE CONFIANCA MAXIMO -> '
      ACCEPT *,RMAX

```

```

C
C RAIO DE CONFIANCA INICIAL
C

```

```

      RCON = RMAX

```

```

C
C INICIALIZA MARCADOR DE MESMO PONTO NA RESOLUCAO DO SUBPROBLEMA
C

```

```

      SAMEX = .FALSE.

```

```

C
C INICIALIZA MARCADORES USADOS EM SUBPRB
C

```

```

      FSI = .FALSE.
      FGEP = .FALSE.
      FLEP = .FALSE.
      FMEC = .FALSE.
      FMEP = .FALSE.
      DEZE = .FALSE.
      ORTO1 = .FALSE.
      HARDPB = .FALSE.

```

```

C
C CONTADOR DE ITERACOES
C

```

```

      K = 0

```

```

C
C NUMERO DE AVALIACOES DA FUNCAO
C

```

```

      NAVAL = 0
C
C      INICIALIZA MARCADOR DE FALHA EM FUNC E GKBK
C
      FALHA = .FALSE.
C
      250 CONTINUE
        IF (.NOT.SAMEX) THEN
          CALL GKBK (XK, GK, BK, IPRB, N, NF, REPSGRD, FALHA)
          K = K + 1
        ENDIF
        IF (.NOT.FALHA) THEN
          CALL SUBPRB (K, N, RPRB, RCON, XK, GK, BK, SK, QSK,
$             ISOL, SAMEX, GK, RK, W, C, U, DIM, ENXK,
$             ENG, A, B2, Q2, R2, WC, GC, CC, UC, DIM2,
$             S1, SGEF, SLEP, FSI, FGEP, FLEP, DE2E,
$             FMEC, MGEC, FMEP, MGEP, ORTO1, HARDFB,
$             EPSHRD, EPSCON, EPSPRB, EPS2ES, EMACH,
$             MAXNEW, MAXL1, MAXL2)
          IF (ISOL.NE.0) THEN
            DO I = 1, N
              XK1(I) = SK(I) + XK(I)
            ENDDO
            ENSK = EUCNOR (SK, N)
            IF (.NOT.SAMEX) THEN
              IF (K.EQ.1) THEN
C
C      PRIMEIRA ITERACAO
C
                FALHA = .FALSE.
                NAVAL = NAVAL + 1
                FXK = FUNC (XK, IPRB, N, NF, FALHA)
                ELSE
                  FXK = FXK1
                ENDIF
                write(*,*)'funcao em xk = ',fxk
                WRITE(8,*)'FUNCAO EM XK = ',FXK
                ENDIF
                IF (.NOT.FALHA) THEN
                  NAVAL = NAVAL + 1
                  FXK1 = FUNC (XK1, IPRB, N, NF, FALHA)
                  IF (.NOT.FALHA) THEN
                    A1 = FXK + 1.0D-4 * QSK
                    A2 = FXK1 - A1
                    IF (FXK1.LE.A1) THEN
C
C      CONDICAO DE ARMIJO SATISFEITA
C
                      SAMEX = .FALSE.
                      DO I = 1, N
                        XK(I) = XK1(I)
                      ENDDO
                      RCON = RMAX
                      FSI = .FALSE.
                      FGEP = .FALSE.
                      FLEP = .FALSE.
                      FMEC = .FALSE.

```



```

        FMEP = .FALSE.
        DEZE = .FALSE.
        ORTO1 = .FALSE.
        HARDPB = .FALSE.
    ELSE
C      FALHA EM ARMIJO
        SAMEX = .TRUE.
        RCON = ENSK / 2.D0
    ENDIF
    ENDIF
    ENDIF
    ENDIF
    ENDIF
    IF ((.NOT.FALHA).AND.(K.LE.KMAX).AND.(ENSK.GT.EPSCON)
$      .AND.((ENG.GT.EPSGRD).OR.(ISOL.NE.6))
$      .AND.((ENG.GT.EPSGRD).OR.(ISOL.NE.1))
$      .AND.((ENG.GT.EPSGRD).OR.(ISOL.NE.2))
$      .AND.((ENG.GT.EPSGRD).OR.(DABS(RK(1)).GT.EPSGRD))
$      .AND.(ISOL.NE.0)) GOTO 250
    WRITE (8,10330)
    IF (.NOT.FALHA) THEN
        IF (ISOL.NE.0) THEN
            IF (K.LE.KMAX) THEN
                IF (ENSK.GT.EPSCON) THEN
C      CONVERGIU (MINIMIZADOR IRRESTRITO)
                    IF (DABS(RK(1)).LE.EPSGRD)
$      THEN
C      HESSIANA SEMI-DEFINIDA POSITIVA
                            WRITE (*,10335)
                            WRITE (8,10335)
                        ENDIF
C      NORMA DO GRADIENTE
                            WRITE (*,10340) ENG
                            WRITE (8,10340) ENG
                            WRITE (*,10352)
                            WRITE (8,10352)
                            DO I = 1,N
                                IF (GK(I).NE.0.D0) THEN
                                    WRITE (*,10355) XK(I), GK(I)
                                    WRITE (8,10355) XK(I), GK(I)
                                ELSE
                                    WRITE (*,10357) XK(I)
                                    WRITE (8,10357) XK(I)
                                ENDIF
                            ENDDO
                            WRITE (*,10390) FXK
                            WRITE (8,10390) FXK
                        ELSE
C      CONVERGIU PELA NORMA DA DIRECAO
                            WRITE (*,10350) ENSK
                            WRITE (8,10350) ENSK
                            WRITE (*,10360)
                            WRITE (8,10360)
C      CALCULO DO GRADIENTE EM XK1
                            CALL GKBK (XK1, GK, BK, IPRB, N, NF, EPSGRD,

```

```

$                                FALHA)
                                IF (.NOT.FALHA) THEN
                                    DO I = 1,N
                                        IF (GK(I).NE.0.0) THEN
                                            WRITE (*,10370) SK(I),XK1(I),GK(I),
$                                             XK1(I)/GK(I)
                                            WRITE (8,10370) SK(I),XK1(I),GK(I),
$                                             XK1(I)/GK(I)
                                        ELSE
                                            WRITE (*,10380) SK(I), XK1(I), GK(I)
                                            WRITE (8,10380) SK(I), XK1(I), GK(I)
                                        ENDIF
                                    ENDDO
                                ENDIF
                                WRITE (*,10390) FXX1
                                WRITE (8,10390) FXX1
                                ENDIF
                                WRITE (*,10400) NAVAL
                                WRITE (8,10400) NAVAL
                                WRITE (*,10405) K
                                WRITE (8,10405) K
                                ELSE
C      NAO CONVERGIU COM KMAX ITERACOES
                                    WRITE (*,10400) NAVAL
                                    WRITE (8,10400) NAVAL
                                    WRITE (*,10410) KMAX
                                    WRITE (8,10410) KMAX
                                ENDIF
                                ELSE
C      SUBPROBLEMA SEM SOLUCAO
                                    WRITE (*,10400) NAVAL
                                    WRITE (8,10400) NAVAL
                                    WRITE (*,10420)
                                    WRITE (8,10420)
                                ENDIF
                                ELSE
C      PONTO FORA DO DOMINIO DA FUNCAO
                                    WRITE (*,10400) NAVAL
                                    WRITE (8,10400) NAVAL
                                    WRITE (*,10430)
                                    WRITE (8,10430)
                                ENDIF
                                ENDIF
                                ENDIF
                                ENDIF
                                CLOSE (8)
C
10010 FORMAT (1X,'RESOLUCAO DE PROBLEMA DE MINIMIZACAO RESTRITO A UMA',
$ /,1X,' BOLA EUCLIDIANA, USANDO REGIAO DE CONFIANCA',/)
10020 FORMAT (1X,'ENTRE COM O NUMERO DO SUBPROBLEMA (1<=IPRB<=19)',
$ /,5X,'IPRB -> ',)$)
10030 FORMAT (1X,'POSSIBILIDADES PARA O RAO DO PROBLEMA : ',
$ /,5X,'1 -> RAO GRANDE ',/,5X,'2 -> RAO PEQUENO ',
$ /,5X,'SUA OPCAO -> ',)$)
10040 FORMAT (1X,'POSSIBILIDADES PARA O PONTO INICIAL : ',

```

```

$ /,5X,'1 -> NO INTERIOR ',/,5X,'2 -> NA FRONTEIRA ',
$ /,5X,'SUA OPCAO -> ',$(
10090 FORMAT (1X,'FUNCAO DE ROSEN BROCK',/)
10100 FORMAT (1X,'FUNCAO DE FREUDENSTEIN E ROTH',/)
10110 FORMAT (1X,'HELICAL VALLEY FUNCTION',/)
10120 FORMAT (1X,'FUNCAO EXP6 DE BIGGS',/)
10130 FORMAT (1X,'FUNCAO GAUSSIANA',/)
10140 FORMAT (1X,'FUNCAO MAL ESCALADA DE POWELL',/)
10150 FORMAT (1X,'FUNCAO DA CAIXA TRIDIMENSIONAL',/)
10160 FORMAT (1X,'FUNCAO VARIAVELMENTE DIMENSIONADA',/)
10170 FORMAT (1X,'FUNCAO DE WATSON',/)
10180 FORMAT (1X,'FUNCAO PENALIDADE I',/)
10190 FORMAT (1X,'FUNCAO PENALIDADE II',/)
10200 FORMAT (1X,'FUNCAO MAL ESCALADA DE BROWN',/)
10210 FORMAT (1X,'FUNCAO DE BROWN E DENNIS',/)
10220 FORMAT (1X,'GULF RESEARCH AND DEVELOPMENT FUNCTION',/)
10230 FORMAT (1X,'FUNCAO TRIGONOMETRICA',/)
10240 FORMAT (1X,'FUNCAO DE ROSEN BROCK ESTENDIDA',/)
10250 FORMAT (1X,'FUNCAO SINGULAR DE POWELL ESTENDIDA TRANSLADADA',
$ ' DE (1,1,...,1)',/)
10260 FORMAT (1X,'FUNCAO DE BEALE',/)
10270 FORMAT (1X,'FUNCAO DE WOOD',/)
10280 FORMAT (1X,'RAIO DO PROBLEMA -> ',1PE14.7)
10290 FORMAT (1X,'(RAIO GRANDE)')
10300 FORMAT (1X,'(RAIO PEQUENO)')
10310 FORMAT (1X,'PONTO INICIAL NO INTERIOR')
10320 FORMAT (1X,'PONTO INICIAL NA FRONTEIRA')
10330 FORMAT (1X,'-----')
$ '-----')
10335 FORMAT (1X,'HESSIANA SEMI-DEFINIDA POSITIVA',/)
10340 FORMAT (1X,'NORMA DO GRADIENTE : ', 1PE14.7,/)
10345 FORMAT (1X,'PONTO ESTRANGULADO PELA SEQUENCIA DE RAIOS DE '
$ 'CONFIANCA',/, '- ULTIMO RAO : ', 1PE14.7,/)
10348 FORMAT (1X,'NUMERO DE AVALIACOES DA FUNCAO ATE O MOMENTO: ',15)
10350 FORMAT (1X,'NORMA DA DIRECAO: ', 1PE14.7,/)
10352 FORMAT (1X,'----- XK(I) ----- GK(I) ----')
10355 FORMAT (1X,2(1PE14.5,2X) )
10357 FORMAT (1X,1PE14.5,2X,' -----')
10360 FORMAT (1X,'----- SK ----- X(K+1) ----- GK+1 -----',
$ '- XK+1(I)/GK+1(I) - ')
10370 FORMAT (1X, 4(1PE14.5,2X) )
10380 FORMAT (1X, 3(1PE14.5,2X), ' -----')
10390 FORMAT (1X,'VALOR DA FUNCAO NO PONTO SOLUCAO : F(X) = ',1PE14.7)
10400 FORMAT (1X,'NUMERO TOTAL DE AVALIACOES DE FUNCAO : ',15)
10405 FORMAT (1X,'NUMERO TOTAL DE ITERACOES : ',15)
10410 FORMAT (1X,'O METODO NAO CONVERGIU EM ',13,' ITERACOES')
10420 FORMAT (1X,'SUBPROBLEMA SEM SOLUCAO -> INTERRUPTAO')
10430 FORMAT (1X,'PONTO FORA DO DOMINIO DA FUNCAO -> INTERRUPTAO ')
C
END

```

```

SUBROUTINE SUBPRB(K, N, RP, RCK, XK, GK, BK, S, QS, ISOL, SAMEX,
$ QK, RK, W, C, U, DIM, ENXK, ENG, A, B2, Q2, R2, WC, GC, CC,
$ UC, DIM2, SI, SGEP, SLEP, FSI, FGEP, FLEP, DE2E, FMEC,
$ MGEC, FMGP, MGEP, ORTO1, HARDFB, EPSHRD, EPSCON, EPSPRB,
$ EPS2ES, EMACH, MAXNEW, MAXL1, MAXL2)

```

```

RESOLUCAO DO K-ESIMO SUBPROBLEMA

```

```

PARAMETROS
-----

```

```

REAL*8 RP, RCK, XK(1), GK(1), BK(50,1), S(1), QS,
$ QK(50,50), RK(50), W(50), C(50), U(50), ENXK, ENG,
$ A(50,50), B2(50,50), Q2(50,50), R2(50), WC(50),
$ GC(50), CC(50), UC(50), SI(50), SGEP(50), SLEP(50),
$ MGEC, MGEP, EPSHRD, EPSCON, EPSPRB, EPS2ES, EMACH

```

```

INTEGER K, N, ISOL, DIM, DIM2, MAXNEW, MAXL1, MAXL2

```

```

LOGICAL SAMEX, FSI, FGEP, FLEP, DE2E, FMEC, FMGP,
$ ORTO1, HARDFB

```

```

RP  = raio do problema
RCK = raio de confianca
XK  = k-esimo ponto
GK  = k-esimo vetor gradiente
BK  = matriz hessiana do k-esimo subproblema, que aproxima
      o hessiano do problema no ponto XK
S   = direcao solucao
QS  = valor da quadratica na solucao
QK  = matriz ortogonal (colunas = autovetores de BK)
RK  = vetor dos autovalores de BK (ordem crescente)
W   = vetor BK XK
C   = vetor (QK)t GK
U   = vetor (QK)t XK
ENXK = norma euclidiana do vetor XK
ENG  = norma euclidiana do vetor GK
A    = matriz cujas colunas formam base ortonormal para
      o complemento ortogonal de XK
B2   = produto  $A^t \cdot BK \cdot A$  (hessiana da quadratica na mudanca de
      variaveis para a (n-2)-esfera)
Q2   = matriz ortogonal (colunas = autovetores de B2)
R2   = vetor dos autovalores de  $(A)^t \cdot BK \cdot A$  (ordem crescente)
WC   = vetor  $(A)^t \cdot W$ 
GC   = vetor  $(A)^t \cdot GK$ 
CC   = vetor  $(Q2)^t \cdot GC$ 
UC   = vetor  $(Q2)^t \cdot WC$ 
SI   = solucao irrestrita
SGEP = solucao global na esfera do problema
SLEP = solucao local nao global na esfera do problema
MGEC = multiplicador de Lagrange correspondente a SGEC
MGEP = idem para SGEP
EPSHRD = precisao para detectar "hard case"
EPSCON = precisao para resolver na esfera de confianca

```

```

C EPSFRB = precisao para resolver na esfera do problema
C EPS2ES = precisao para resolver na (n-2)-esfera
C EMACH = precisao da maquina
C
C K = contador de iteracoes do algoritmo
C N = dimensao do subproblema
C ISOL = codigo do tipo de solucao obtida
C      0 -> subproblema sem solucao
C      1 -> solucao global na esfera de confianca
C      2 -> solucao local na esfera de confianca
C      3 -> solucao global na esfera do problema
C      4 -> solucao local na esfera do problema
C      5 -> solucao global na (n-2)-esfera
C      6 -> solucao irrestrita
C DIM = dimensao do autoespaco associado ao menor autovalor de BK
C DIM2 = idem para o menor autovalor de B2
C MAXNEW = numero maximo de iteracoes para NEWTON
C MAXL1 = numero maximo de iteracoes para a fase 1 de MINLOC
C MAXL2 = idem para a fase 2
C
C SAMEX = indica resolucao do subproblema com o mesmo ponto XK
C         e diferente raio de confianca
C FSI = factibilidade de S1
C FGEP = factibilidade de SGEP
C FLEP = factibilidade de SLEP
C DE2E = marca decomposicao espectral na (n-2) esfera
C FMEC = existe multiplicador global na esfera de confianca, mas
C       a solucao associada e' infactivel
C FMEP = idem na esfera do problema
C ORTO1 = QK ortogonal a S1
C       (S1: autoespaco associado ao menor autovalor de BK)
C HARDPB = ocorrencia de "hard-case" na esfera do problema
C
C
C VARIAVEIS LOCAIS
C -----
C
C REAL*8 XK1(50),E(50), F(50), V(50), Y(50),
C $ SGEC(50), SLEC(50),SG2E(50),
C $ CO1(50), CO2(50), CO3(50), SNM(50), POS(50),
C $ TA, TC, T, MI0, ENCEN, ENV, ENSNM,
C $ ENCO1, ENCO2, ENCO3, ENCON, ENPRB, ENAUX,
C $ AUX, ALFA, BETA, RHO, INFINITO, FI, FILIN,
C $ QSI, QSGEC, QSLEC, QSGEP, QSLEP, QSG2E, QSOL
C
C INTEGER M, IFAIL, KONT, ERRO, I, J, KONT1, KONT2, IDEC
C
C LOGICAL FGEC, FLEC, FG2E, ORTO2, ORTO3, HARDCK, HARD2E
C
C XK1 = k+1-esimo ponto
C E,F = vetores auxiliares
C V    = vetor da transformacao de Householder
C       que leva (1/enxk)*xk em e1 (i.e. vetor elementar)
C Y    = solucao na (n-2)-esfera
C SGEC = solucao global na esfera de confianca

```

C SLEC = solucao local nao global na esfera de confianca
 C SG2E = solucao global nas 2 esferas (esfera N-2 dimensional)
 C CO1 = vetor dos co-senos dos angulos entre GK e os autovetores
 C associados ao menor autovalor de BK
 C CO2 = idem para os angulos entre W - GK e os autovetores
 C associados ao menor autovalor de BK
 C CO3 = idem para os angulos entre (A)t (GK + alfa * W) e os
 C autovetores associados ao menor autovalor de B2
 C SNM = solucao de norma minima
 C POS = projecao da origem do outro sistema de coordenadas
 C TA, TC = coeficientes de equacao de segundo grau
 C T = variavel de equacao de segundo grau
 C MI0 = multiplicador de Lagrange
 C ENCEN = norma euclidiana do vetor centro da (n-2)-esfera
 C ENV = norma euclidiana do vetor V (transf. Householder)
 C ENSNM = norma euclidiana do vetor solucao de norma minima
 C ENCO1 = norma euclidiana do vetor CO1
 C ENCO2 = idem para o vetor CO2
 C ENCO3 = idem para o vetor CO3
 C ENCON = norma do candidato a solucao para testar factibilidade
 C na bola de confianca
 C ENPRB = idem na bola do problema
 C ENAUX = norma do vetor auxiliar
 C AUX = variavel auxiliar
 C ALFA, BETA = escalares utilizados na (n-2)-esfera
 C RHO = raio da (n-2)-esfera
 C INFINITO = auxiliar na obtencao do menor valor da quadratica
 C FI = norma-2 da direcao S, S + XK, ou Y (ou quadrado dela),
 C conforme se esteja resolvendo na esfera de confianca, na
 C esfera do problema ou na (n-2) esfera
 C FILIN = derivada de FI (dFI/dMI)
 C QSI = valor da quadratica em SI
 C QSGEC = valor da quadratica em SGEC
 C QSLEC = valor da quadratica em SLEC
 C QSGEP = valor da quadratica em SGEP
 C QSLEP = valor da quadratica em SLEP
 C QSG2E = valor da quadratica em SG2E
 C QSOL = valor auxiliar na obtencao do minimo
 C
 C M = N - 1
 C IFAIL = indicador de erro para F02ABF (rotina da NAG)
 C KONT = numero de iteracoes efetuado em NEWTON
 C ERRO = codigo do tipo de saida para NEWTON e MINLOC
 C I, J = contadores auxiliares
 C KONT1 = numero de iteracoes efetuado na fase 1 de MINLOC
 C KONT2 = idem para fase 2
 C IDEC = indice declarado para linhas nas matrizes
 C
 C FGEC = factibilidade de SGEC
 C FLEC = factibilidade de SLEC
 C FG2E = factibilidade de SG2E
 C ORTO2 = W - GK ortogonal a S1
 C ORTO3 = (A)t (GK + beta * W) ortogonal a S2
 C (S2: autoespaco associado ao menor autovalor de B2)
 C HARDCK = ocorrencia de "hard-case" na esfera de confianca

```

C      HARD2E = idem na (n-2)-esfera
C
C      FUNCOES UTILIZADAS
C      -----
C
C      double precision DSQRT (LIBRARY/FORTRAN)
C      double precision EUCNOR
C      double precision QUAD
C
C      SUBROTINAS UTILIZADAS
C      -----
C
C      F02ABF (NAG)
C      FIFILIN
C      MINLOC
C      MULTMAT
C      NEWTON
C      PREMUL
C
C      INICIALIZACAO
C      -----
C
C      INFINITO = 1.0D+30
C      IDEC = 50
C      M = N - 1
C      WRITE (*,*) 'ITERACAO ',K
C      WRITE (8,10010)
C      IF (.NOT.SAMEX) THEN
C         WRITE (8,10020) K
C         WRITE (8,10030)
C         DO I = 1,N
C            WRITE (8,10040) I, XK(I), GK(I)
C         ENDDO
C      ENDIF
C      WRITE (8,10050) RCK
C      IF (.NOT.SAMEX) THEN
C
C      DECOMPOSICAO ESPECTRAL DA MATRIZ HESSIANA BK
C      -----
C      UTILIZANDO A ROTINA F02ABF DA NAG
C      -----
C
C      IFAIL = 1
C      CALL F02ABF (BK, IDEC, N, RK, QK, IDEC , E, IFAIL)
C      IF (IFAIL.NE.0) THEN
C         WRITE (8,10060) IFAIL
C         STOP
C      ENDIF
C
C      DIMENSAO DE S1 ( AUTOESPACO ASSOCIADO A RK(1) )
C      -----
C
C      I = 1
C      REPITA
C      1      I = I + 1
C            AUX = DABS(RK(1) - RK(I))
C            IF ((I.LE.N).AND.(AUX.LT.EMACH)) GOTO 1
C      FIM

```

DIM = I - 1

MINIMIZADOR IRRESTRITO (CARACTERIZACAO DO HARD CASE)

CALCULO DO VETOR C = (QK)t GK

CALL PREMULT (QK, N, N, GK, C, .TRUE.)

CALCULO DO VETOR W = (BK) XK

CALL PREMULT (BK, N, N, XK, W, .TRUE.)

CALCULO DO VETOR U = (QK)t XK

CALL PREMULT (QK, N, N, XK, U, .TRUE.)

ENXK = EUCNOR (XK, N)

ENG = EUCNOR (GK, N)

IF (RK(1).GT.0.D0) THEN

HESSIANA POSITIVA DEFINIDA

ORTO1 = .FALSE.

ORTO2 = .FALSE.

CALCULO DA DIRECAO DE NEWTON (SI)

E: VETOR AUXILIAR = - (DK)-1 (QK)t GK

DO I = 1, N

E(I) = - C(I) / RK(I)

END DO

SI: VETOR QK E = -QK (DK)-1 (QK)t GK

CALL PREMULT (QK, N, N, E, SI, .FALSE.)

CALCULO DAS NORMAS PARA TESTAR A FACTIBILIDADE DE SI

ENCON = EUCNOR (SI, N)

DO I = 1, N

XK1(I) = SI(I) + XK(I)

END DO

ENPRB = EUCNOR (XK1, N)

FSI = (((ENCON-RCK)/ENCON).LE.EPSCON).AND.

\$ (((ENPRB-RP)/ENPRB).LE.EPSPRB))

ELSE

HESSIANA NAO POSITIVA DEFINIDA

FSI = .FALSE.

CALCULO DOS VETORES ADIMENSIONAIS CO1 E CO2, QUE CARACTERIZAM A ORTOGONALIDADE DE GK E DE (W - GK), RESPECTIVAMENTE, EM RELACAO


```

C      A S1
C
C      E: VETOR AUXILIAR = W - GK
C

```

```

      DO I = 1,N
        E(I) = W(I) - GK(I)
      END DO
      ENAUX = EUCNOR (E,N)
      DO I = 1,DIM
        IF (ENG.NE.0.D0) CO1(I) = C(I)/ENG
        IF (ENAUX.NE.0.D0) THEN
          AUX = 0.D0
          DO J = 1,N
            AUX = AUX + GK(J,I) * E(J)
          END DO
          CO2(I) = AUX/ENAUX
        ENDIF
      END DO
      IF (ENG.EQ.0.D0) THEN
        ORTO1 = .FALSE.
      ELSE
        ENCO1 = EUCNOR (CO1,DIM)
        ORTO1 = (ENCO1.LE.EPSHRD)
      ENDIF
      IF (ENAUX.EQ.0.D0) THEN
        ORTO2 = .FALSE.
      ELSE
        ENCO2 = EUCNOR (CO2,DIM)
        ORTO2 = (ENCO2.LE.EPSHRD)
      ENDIF
      WRITE (8,10070) RK(1)

```

```

      IF (ORTO1) WRITE (8,10080)
      IF (ORTO2) WRITE (8,10090)

```

```

    ENDIF
  ELSE
    IF (FSI) THEN
      ENCON = EUCNOR (SI,N)
      DO I = 1,N
        XK1(I) = SI(I) + XK(I)
      END DO
      ENPRB = EUCNOR (XK1,N)
      FSI = (((ENCON-RCK)/ENCON).LE.EPSCON).AND.
$      (((ENPRB-RP)/ENPRB).LE.EPSPRB)
    ENDIF
  ENDIF

```

```

C      MINIMIZADOR GLOBAL RESTRITO A ESFERA DE CONFIANCA
C      -----
C

```

```

      IF (.NOT.FSI) THEN
        IF (ORTO1) THEN

```

```

C      DETECTANDO POSSIVEL HARD-CASE
C

```

```

C      AVALIANDO FI(-RK(1)) = (NORMA (S))**2 E FILIN = DFI/DMI(-RK(1))

```

```

C      CALL FIFILIN (C, U, RK, N, DIM, -RK(1), FI, FILIN, 1,
$      .TRUE., 0.10, .TRUE.)
      HARDCK = ((FI - (RCK ** 2)).LT.EPSHRD)
      ELSE
      HARDCK = .FALSE.
      ENDIF
      IF (HARDCK) THEN
C
C      HARD-CASE NA BOLA DE CONFIANCA
C
C      WRITE (8,10100)
C
C      CALCULO DE SNM: SOLUCAO DE NORMA MINIMA PARA O SISTEMA
C      (BK - RK(1) In) S = - GK
C
C      E: VETOR AUXILIAR = - (D - RK(1) In) + C
C
C      DO I = 1, N
C      IF (I.LE.DIM) THEN
C      E(I) = 0.10
C      ELSE
C      E(I) = C(I)/(RK(1) - RK(I))
C      ENDIF
C      END DO
C      CALL PREMULT (QK, N, N, E, SNM, .FALSE.)
C
C      CALCULO DE POS: PROJECAO DE S = - XK NA VARIEDADE
C      ( KER [ BK - RK(1) In ] + SNM )
C
C      E: VETOR AUXILIAR = [ (D - RK(1) In) + (D - RK(1) In) - In ] U
C
C      DO I = 1, N
C      IF (I.LE.DIM) THEN
C      E(I) = - U(I)
C      ELSE
C      E(I) = 0.10
C      ENDIF
C      END DO
C
C      F: VETOR AUXILIAR = (QK) E = PROJECAO DE - XK NO SUBESPACO
C      KER [ BK - RK(1) In ]
C
C      CALL PREMULT (QK, N, N, E, F, .FALSE.)
C      DO I = 1, N
C      POS(I) = F(I) + SNM(I)
C      END DO
C
C      RESOLUCAO DA EQUACAO DE SEGUNDO GRAU
C      TA * T ** 2 + TC = 0, PARA OBTEN A INTERSECAO
C      DA RETA QUE PASSA POR SNM E POS COM A ESFERA DE CONFIANCA
C
C      CALCULO DOS COEFICIENTES
C
C      COEFICIENTE DO TERMO INDEPENDENTE

```

```

C      ENSNM = EUCNOR (SNM, N)
C      TC = (RCK - ENSNM) * (RCK + ENSNM)
C      IF (TC.LT.0.D0) THEN
C          WRITE (8, 10110)
C          FGEC = .FALSE.
C          WRITE (8, 10120) FGEC
C      ELSE IF (TC.EQ.0.D0) THEN
C          T = 0.D0
C      ELSE
C
C      COEFICIENTE DO TERMO T ** 2
C      CASO A PROJECAO DE -XK EM KER [BK - RK(I) In] SEJA NULA,
C      TOMAR DIRECAO DO PRIMEIRO AUTOVETOR DE BK ASSOCIADO A RK(1)
C
C          AUX = EUCNOR (F, N)
C          IF (AUX.EQ.0.D0) THEN
C              T = DSQRT(TC)
C              DO I = 1, N
C                  F(I) = QK(I, 1)
C              ENDDO
C          ELSE
C              T = (DSQRT (TC))/AUX
C          ENDIF
C      ENDIF
C
C      CALCULO DO CANDIDATO A SOLUCAO
C
C      IF (TC.GE.0.D0) THEN
C          DO I = 1, N
C              SGEC(I) = SNM(I) + T * F(I)
C              XK1(I) = SGEC(I) + XK(I)
C          END DO
C
C      TESTE DA FACTIBILIDADE NA BOLA DO PROBLEMA
C
C          ENPRB = EUCNOR (XK1, N)
C          FGEC = (((ENPRB - RF)/ENPRB).LE.EPSPRB)
C          WRITE (8, 10120) FGEC
C      ENDIF
C      FMEC = .FALSE.
C      ELSE
C
C      NAO OCORRE HARD-CASE NA BOLA DE CONFIANCA
C
C          ERRO = 1
C
C      INICIALIZA MULTIPLICADOR DE LAGRANGE
C
C          MI0 = -RK(1) + 0.5D0
C
C      CALCULA RAIZ DA EQUACAO 1/FI(MI) = 1/RCK
C
C          CALL NEWTON (QK, C, XK, U, RK, SGEC, N, DIM, RCK, MI0,
$              EPSCON, MAXNEW, 1, ERRO, KONT, 0.D0 )

```

```

C      IF (ERRO.EQ.1) THEN
C
C      ERRO = 1 -> SUCESSO EM NEWTON
C

```

```

      MGEC = MI0
      WRITE (8,10130) KONT

```

```

C      TESTE DA FACTIBILIDADE DE SGEC
C

```

```

      DO I = 1, N
        XK1(I) = SGEC(I) + XK(I)
      END DO
      ENPRB = EUCNOR (XK1, N)
      FGEC = (((ENPRB - RP)/ENPRB).LE.EPSPRB)
      FMEC = (.NOT.FGEC)
      WRITE (8,10120) FGEC
    ELSE

```

```

C      ERRO = 2 -> KONT > MAXNEW
C

```

```

      WRITE (8,10140) MAXNEW
      FGEC = .FALSE.
      FMEC = .FALSE.
    ENDIF
  ENDIF

```

```

C      MINIMIZADOR LOCAL NAO GLOBAL (ESFERA DE CONFIANCA)
C

```

```

      IF ((-RK(1).GT.0.0D0).AND.(-RK(2).LT.0.0D0)) THEN
        CALL FIFILIN (C, U, RK, N, DIM, 0.0D0, FI, FILIN, 1,
$          .FALSE., 0.0D0, .TRUE.)

```

```

      ELSE
        FI = 0.0D0
      ENDIF

```

```

      IF ((ORTO1).OR.(DIM.GT.1).OR.(FI.GT.(RCK**2))) THEN

```

```

C      NAO EXISTE MINIMIZADOR LOCAL NAO GLOBAL
C

```

```

      FLEC = .FALSE.
    ELSE
      IF (FMEC) THEN

```

```

C      EXISTE SOLUCAO GLOBAL, MAS INFACTIVEL
C

```

```

      ERRO = 1

```

```

C      CALCULO DE SLEC PELO ALGORITMO DE MARTINEZ
C

```

```

      CALL MINLOC (QK, C, XK, U, RK, SLEC, N, RCK, MGEC, EPSCON,
$        1, ERRO, DIM, MAXL1, MAXL2, KONT1, KONT2, EHACH)

```

```

C      INICIO DO CASE
C

```

```

      GOTO (10, 20, 30, 40) ERRO

```

```

      GOTO 50
10    CONTINUE
C
C    ERRO = 1 -> SUCESSO EM MINLOC
C
      WRITE(8,10150) KONT1, KONT2
C
C    TESTE DA FACTIBILIDADE DE SLEC
C
      DO I = 1, N
        XK1(I) = SLEC(I) + XK(I)
      END DO
      ENPRB = EUCNOR (XK1, N)
      FLEC = (((ENPRB - RP)/ENPRB).LE.EPSPRB)
      WRITE (8,10160) FLEC
      GOTO 50
20    CONTINUE
C
C    ERRO = 2 -> NAO EXISTE MINIMIZADOR LOCAL NAO GLOBAL
C          NA ESFERA DE CONFIANCA
C
      FLEC = .FALSE.
      WRITE (8,10170)
      GOTO 50
30    CONTINUE
C
C    ERRO = 3 -> KONT1 > MAXL1
C
      WRITE (8,10180) MAXL1
      FLEC = .FALSE.
      GOTO 50
40    CONTINUE
C
C    ERRO = 4 -> KONT2 > MAXL2
C
      WRITE (8, 10190) MAXL2
      FLEC = .FALSE.
50    CONTINUE
C
C    FIM DO CASE
C
      ELSE
        FLEC = .FALSE.
        WRITE (8, 10160) FLEC
      ENDIF
    ENDIF
C
C    MINIMIZADOR GLOBAL RESTRITO A ESFERA DO PROBLEMA
C    -----
      IF (SAMEX) THEN
        IF (FGEP) THEN
          ENCON = EUCNOR (SGEP,N)
          FGEP = (((ENCON - RCK)/ENCON).LE.EPSCON)
          FMEP = ((.NOT.FGEP).AND.(.NOT.HARDPB))
        ENDIF

```

```
WRITE (8, 10200) FGEP  
ELSE  
IF (ORT02) THEN
```

DETECTANDO POSSIVEL HARD-CASE

$$\text{AVALIANDO FI}(-RK(1)) = (\text{NORMA}(S+XK))^{**2} \text{ E FILIN} = \text{DFI}/\text{DMI}(-RK(1))$$

```
CALL FIFILIN (C, U, RK, N, DIM, -RK(1), FI, FILIN, 2,  
$              .TRUE., 0.00 ,.TRUE.)
```

```
HARDFB = ((FI - (RP ** 2)),LT,EPSHRD)
```

ELSE

HARDPE = FALSE.

END

IF (HARDPB) THEN

HARD-CASE NA BOLA DO PROBLEMA

WRITE (8, 10210)

CALCULO DE SNM: SOLUCAO DE NORMA MINIMA PARA O SISTEMA
 $(BK - RK(1) \cdot In) \cdot Z = (BK) \cdot XK - GK$

F: VETOR AUXILIAR = (QK)t (BK) XK = (QK)t W

E: VETOR AUXILIAR = (D - RK(1) In)+ (F - C)

CALL PREMULY (QK, N, N, W, F, .TRUE.)

$$D(I) = \{1, \dots, N\}$$

IF (I.LE.DIM) THEN

$$E(I) = 0.10$$

ELSE

$$E(I) = (F(I) - C(I)) / (RK(I) - RK(1))$$

END

END DO

CALL PREMUL (QK, N, N, E, SNH, .FALSE.)

CALCULO DE POS: PROJECAO DE $Z = X_K$ NA VARIEDADE

```

( KER [ BK - BK(1) In J + SNM ]

```

$$E: \text{VETOR AUXILIAR} = [I_n - (D - RK(1) I_n) + (D - RK(1) I_n)] U$$

DO I = 1, N

IF C1 LE DEDS THEN

$$E(I) = U(I)$$

ELSE

$$E(I) = 0,10$$

END IF

END DO

F: VETOR AUXILIAR = (QK) E = PROJECAO DE XK NO SUBESPACO

$$\ker C \cap K = RK(1) \text{ in } J$$

CALL PREMUL (OK, N, N, E, F, FALSE.)

DO NOT WRITE IN THESE SPACES

$$FOS(I) = F(I) + SNH(I)$$

```

C      END DO
C
C      RESOLUCAO DA EQUACAO DE SEGUNDO GRAU
C       $TA * T ** 2 + TC = 0$ , PARA OBTENIR A INTERSECAO
C      DA RETA QUE PASSA POR SNM E POS COM A ESFERA DO PROBLEMA
C
C      CALCULO DOS COEFICIENTES
C
C      COEFICIENTE DO TERMO INDEPENDENTE
C
C      ENSNM = EUCNOR (SNM, N)
C      TC = (RP + ENSNM) * (RP - ENSNM)
C      IF (TC.LT.0.D0) THEN
C          WRITE (8,10220)
C          FGEP = .FALSE.
C          WRITE (8,10200) FGEP
C      ELSE IF (TC.EQ.0.D0) THEN
C          T = 0.D0
C      ELSE
C
C      COEF. DO TERMO  $T ** 2$ 
C      CASO A PROJECAO DE XK EM KER [BK - RK(1) InJ SEJA NULA
C      TOMAR DIRECAO DO PRIMEIRO AUTOVETOR DE BK ASSOCIADO A RK(1)
C
C          AUX = EUCNOR (F,N)
C          IF (AUX.EQ.0.D0) THEN
C              T = DSQRT(TC)
C              DO I = 1, N
C                  F(I) = QK(I,1)
C              ENDDO
C          ELSE
C              TA = AUX ** 2
C              T = (DSQRT (TC))/AUX
C          ENDIF
C      ENDIF
C
C      CALCULO DO CANDIDATO A SOLUCAO
C
C      IF (TC.GE.0.D0) THEN
C          DO I = 1, N
C              SGEP(I) = SNM(I) + T * F(I) - XK(I)
C          END DO
C
C      TESTE DA FACTIBILIDADE NA BOLA DE CONFIANCA
C
C          ENCON = EUCNOR (SGEP, N)
C          FGEP = (((ENCON - RCK)/ENCON).LE.EPSCON)
C          WRITE (8,10200) FGEP
C      ENDIF
C      FMEP = .FALSE.
C      ELSE
C
C      NAO OCORRE HARD-CASE NA BOLA DO PROBLEMA
C
C          ERRO = 1

```

```

      MI0 = -RK(1) + 0.5D0
C
C  CALCULA RAIZ DA EQUACAO 1/FI(MI) = 1/RF
C
      CALL NEWTON (QK, C, XK, U, RK, SGEP, N, DIM, RF, MI0,
$          EPSPRB, MAXNEW, 2, ERRO, KONT, BETA)
C
      IF (ERRO.EQ.1) THEN
C
C  ERRO = 1 -> SUCESSO EM NEWTON
C
      MGEP = MI0
      WRITE (8,10230) KONT
C
C  TESTE DA FACTIBILIDADE DE SGEP
C
      ENCON = EUCNOR (SGEP, N)
      FGEP = (((ENCON - RCK)/ENCON).LE.EPSCON)
      FMEP = .NOT.FGEP
      WRITE (8,10200) FGEP
      ELSE
C
C  ERRO = 2 -> KONT > MAXNEW
C
      WRITE (8, 10240) MAXNEW
      FGEP = .FALSE.
      FMEP = .FALSE.
      ENDIF
      ENDIF
      ENDIF
C
C  MINIMIZADOR LOCAL NAO GLOBAL (ESFERA DO PROBLEMA)
C  -----
C
      IF ((-RK(1).GT.0.0D0).AND.(~RK(2).LT.0.0D0)) THEN
      CALL FIFILIN (C, U, RK, N, DIM, 0.0D0, FI, FILIN, 2,
$          .FALSE., 0.0D0, .TRUE.)
      ELSE
      FI = 0.0D0
      ENDIF
      IF ((ORT02).OR.(DIM.GT.1).OR.(FI.GT.(RF**2))) THEN
C
C  NAO EXISTE MINIMIZADOR LOCAL NAO GLOBAL
C
      FLEP = .FALSE.
      ELSE
      IF (SAMEX.AND.FLEP) THEN
      ENCON = EUCNOR (SLEP,N)
      FLEP = (((ENCON - RCK)/ENCON).LE.EPSCON)
      WRITE (8,10250) FLEP
      ELSE
      IF (FMEP) THEN
C
C  EXISTE SOLUCAO GLOBAL, MAS INFACTIVEL
C

```



```

C          ERRO = 1
C
C          CALCULO DE SLEP FELO ALGORITMO DE MARTINEZ
C
C          CALL MINLOC (OK, C, XK, U, RK, SLEP, N, RP, HGEF,
$              EPSPRB, 2, ERRO, DIM, MAXL1, MAXL2,
$              KONT1, KONT2, EMACH)
C
C          INICIO DO CASE
C
C              GOTO (110, 120, 130, 140) ERRO
C              GOTO 150
110          CONTINUE
C
C          ERRO = 1 -> SUCESSO EM MINLOC
C
C              WRITE (8,10260) KONT1, KONT2
C
C          TESTE DA FACTIBILIDADE DE SLEP
C
C              ENCON = EUCNOR (SLEP, N)
C              FLEP = ((ENCON - RCK)/ENCON).LE.EPSCON)
C              WRITE (8,10250) FLEP
C              GOTO 150
120          CONTINUE
C
C          ERRO = 2 -> NAO EXISTE MINIMIZADOR LOCAL NAO GLOBAL
C              NA ESFERA DO PROBLEMA
C              FLEP = .FALSE.
C              WRITE (8,10270)
C              GOTO 150
130          CONTINUE
C
C          ERRO = 3 -> KONT1 > MAXL1
C
C              WRITE (8,10280) MAXL1
C              FLEP = .FALSE.
C              GOTO 150
140          CONTINUE
C
C          ERRO = 4 -> KONT2 > MAXL2
C
C              WRITE (8,10290) MAXL2
C              FLEP = .FALSE.
150          CONTINUE
C
C          FIM DO CASE
C
C              ELSE
C              FLEP = .FALSE.
C              WRITE (8, 10250) FLEP
C              ENDIF
C          ENDIF
C      ENDIF
C

```

```

C      MINIMIZADOR GLOBAL RESTRITO AS DUAS ESFERAS - (N-2)-ESFERA
C      -----
C      SO PODE HAVER SG2E SE NENHUM DOS MIN. GLOBAIS TIVER SIDO OBTIDO
C
C      IF ((.NOT.FGEC).AND.(.NOT.FGEP)) THEN
C          IF (ENXK.EQ.0.D0) THEN
C              FG2E = .FALSE.
C              WRITE (8,10300) FG2E
C          ELSE
C
C      NORMA DO VETOR CENTRO DA (N-2)-ESFERA
C
C          ALFA = ((RP+RCK)*(RP-RCK))/(2.D0 * ENXK) + 0.5D0 * ENXK
C          ENCEN = DABS(ALFA)
C          IF (ENCEN.EQ.RP) THEN
C
C      (N-2)-ESFERA E' UM UNICO PONTO
C
C          WRITE (8,10310)
C          AUX = RCK / ENXK
C          DO I = 1,N
C              IF (RP.GT.RCK) THEN
C                  SG2E(I) = AUX * XK(I)
C              ELSE
C                  SG2E(I) = - AUX * XK(I)
C              ENDIF
C          END DO
C          FG2E = .TRUE.
C          WRITE (8,10300) FG2E
C          ELSE IF (ENCEN.GT.RP) THEN
C
C      NAO EXISTE (N-2) ESFERA
C
C          FG2E = .FALSE.
C          WRITE (8,10320)
C          IF ((RP.LT.RCK).AND.(RP.LT.(RCK-ENXK))) WRITE (8,10330)
C          IF ((RP.GT.RCK).AND.(RP.GT.(RCK+ENXK))) WRITE (8,10340)
C          ELSE
C
C      RAO DA (N-2)-ESFERA
C
C          RHO = DSQRT ((RP + ENCEN) * (RP - ENCEN))
C          IF ((.NOT.SAMEX).OR.(.NOT.DE2E)) THEN
C
C      MONTAGEM DA MATRIZ A, CUJAS COLUNAS FORMAM BASE PARA
C      O COMPLEMENTO ORTOGONAL DE XK
C      V: VETOR DA TRANSFORMACAO DE HOUSEHOLDER H TAL QUE
C          H(XK) = ESCALAR * E1
C      AUX CONTEM O SINAL DE XK(1)
C
C          AUX = 1.D0
C          IF (XK(1).LT.0.D0) AUX = - AUX
C          V(1) = (XK(1)/ENXK) + AUX
C          DO I = 2,N
C              V(I) = XK(I)/ENXK

```

```

END DO
ENV = EUCNOR(V,N)
ENV = ENV ** 2
C
C COLUNAS DE A: (N-1) ULTIMAS COLUNAS DE H (A: N x (N-1) )
C
DO J = 2,N
DO I = 1,N
IF (I.EQ.J) THEN
A(I,(J-1)) = 1.D0 - 2.D0 * ( V(I) ** 2 ) / ENV
ELSE
A(I,(J-1)) = - 2.D0 * V(I) * V(J) / ENV
ENDIF
END DO
END DO
C
C MONTAR MATRIZ (A)t BK A = B2
C (USO DE Q2 COMO MATRIZ AUXILIAR)
C
CALL MULTMAT (BK, A, Q2, N, N, M, .FALSE.)
CALL MULTMAT (A, Q2, B2, N, M, M, .TRUE.)
C
C DECOMPOSICAO ESPECTRAL DE B2 = (A)t BK A (USANDO F02ABF)
C
IFAIL = 1
CALL F02ABF (B2, IDEC, M, R2, Q2, IDEC, E, IFAIL)
IF (IFAIL.NE.0) THEN
WRITE (8,10060) IFAIL
STOP
ENDIF
C
C DIMENSAO DE S2 ( AUTOESPACO ASSOCIADO A R2(1) )
C
IF (M.EQ.1) THEN
DIM2 = 1
ELSE
I = 1
REFITA
160 I = I + 1
AUX = DABS(R2(I) - R2(1))
IF ((I.LE.M).AND.(AUX.LT.EMACH)) GOTO 160
C
FIM
DIM2 = I - 1
ENDIF
C
C CALCULO DOS VETORES AUXILIARES
C
CALL PREMULT (A, N, M, W, WC, .TRUE.)
CALL PREMULT (A, N, M, GK, GC, .TRUE.)
CALL PREMULT (Q2, M, M, WC, UC, .TRUE.)
CALL PREMULT (Q2, M, M, GC, CC, .TRUE.)
C
C ATUALIZA MARCADOR DA MONTAGEM E DECOMPOSICAO ESPECTRAL DE B2
C
DEZE = .TRUE.

```

```

      ENDIF
C
C   CALCULO DO ESCALAR BETA
C
      BETA = (ALFA/ENXK) - 1.D0
C
C   CARACTERIZACAO DO HARD-CASE NA (N-2)-ESFERA
C
      IF (R2(1).GT.0.D0) THEN
C
C   HESSIANA [(A)t BK A] DEFINIDA POSITIVA
C
          ORTO3 = .FALSE.
          HARD2E = .FALSE.
      ELSE
C
C   HESSIANA [(A)t BK A] NAO DEFINIDA POSITIVA
C
C   CALCULO DO VETOR ADIMENSIONAL CO3, QUE CARACTERIZA A
C   ORTOGONALIDADE DE (A)t (GK + BETA * BK * XK) EM REL. A S2
C
C   E: VETOR AUXILIAR = (A)t (GK + BETA * W)
C
          DO I = 1,M
              E(I) = GC(I) + BETA * WC(I)
          END DO
          ENAUX = EUCNOR (E,M)
          IF (ENAUX.NE.0.D0) THEN
              DO I = 1, DIM2
                  AUX = 0.D0
                  DO J = 1,M
                      AUX = AUX + Q2(J,I) * E(J)
                  END DO
                  CO3(I) = AUX/ENAUX
              END DO
              ENCO3 = EUCNOR (CO3, DIM2)
              ORTO3 = (ENCO3.LE.EPSHRD)
          ELSE
              ORTO3 = .FALSE.
          ENDIF
          IF (ORTO3) THEN
              WRITE (8,10350)
              CALL FIFILIN (CC, UC, R2, M, DIM2, -R2(1), FI,
%                   FILIN, 3, .TRUE., BETA, .TRUE.)
              HARD2E = ((FI - (RHO ** 2)).LT.EPSHRD)
          ELSE
              HARD2E = .FALSE.
          ENDIF
          ENDIF
          IF (.NOT.HARD2E) THEN
C
C   NAO OCORRE HARD-CASE NA (N-2)-ESFERA
C
              ERRO = 1
              MI0 = -R2(1) + 0.5D0

```

```

C
C  CALCULA RAIZ DA EQUACAO  $1/FI(MI) = 1/RHO$ 
C

```

```

$      CALL NEWTON (Q2, CC, XK, UC, R2, Y, M, DIM2, RHO, K10,
      EPS2ES, MAXNEW, 3, ERRO, KONT, BETA)

```

```

C      IF (ERRO.EQ.1) THEN

```

```

C      ERRO = 1 -> SUCESSO EM NEWTON
C

```

```

      WRITE (8, 10360) KONT
      CALL PREMULT (A, N, M, Y, E, .FALSE.)
      DO I = 1, N
        SG2E(I) = BETA * XK(I) + E(I)
      END DO
      FG2E = .TRUE.
    ELSE

```

```

C      ERRO = 2 -> KONT > MAXNEW
C

```

```

      WRITE (8, 10370) MAXNEW
      FG2E = .FALSE.
    ENDIF
  ELSE

```

```

C      HARD-CASE NA (N-2) ESFERA
C

```

```

      WRITE (8, 10380)

```

```

C      CALCULO DE SNM: SOLUCAO DE NORMA MINIMA PARA O SISTEMA
C      ( (A)t B A - R2(1) Im ) Y = - (A)t ( GK + BETA (BK) XK )
C

```

```

C      E: VETOR AUXILIAR = - (D2 - R2(1) Im)+ (CC + BETA UC)
C

```

```

      DO I = 1, M
        IF (I.LE.DIM2) THEN
          E(I) = 0.D0
        ELSE
          E(I) = - (CC(I) + BETA * UC(I))/(R2(I) - R2(1))
        ENDIF
      END DO
      CALL PREMULT (Q2, N, M, E, SNM, .FALSE.)

```

```

C      RESOLUCAO DA EQUACAO DE SEGUNDO GRAU
C       $T ** 2 + TC = 0$ , PARA OBTEN A INTERSECAO DA RETA
C      QUE PASSA POR SNM E TEM A DIRECAO DO PRIMEIRO AUTOVETOR
C      DE B2, ASSOCIADO A R2(1), COM A (N-2)-ESFERA
C

```

```

C      CALCULO DO COEF. DO TERMO INDEPENDENTE
C

```

```

      ENSNM = EUCNOR (SNM, M)
      TC = (RHO + ENSNM) * (RHO - ENSNM)
      IF (TC.LT.0.D0) THEN
        WRITE (8, 10390)
        FG2E = .FALSE.
      ENDIF

```

```

        WRITE (8,10300) FG2E
    ELSE IF (TC.EQ.0.D0) THEN
        T = 0.D0
    ELSE
        T = DSQRT (TC)
    ENDIF

```

C
C
C CALCULO DA SOLUCAO

```

        IF (TC.GE.0.D0) THEN
            DO I = 1,M
                Y(I) = SNM(I) + T * Q2(I,1)
            END DO
            CALL PREMULT (A, N, M, Y, E, .FALSE.)
            DO I = 1,N
                SG2E(I) = E(I) + BETA * XK(I)
            END DO
            FG2E = .TRUE.
        ENDIF
    ENDIF
ENDIF
ELSE
    FG2E = .FALSE.
ENDIF

```

C
C
C DECISAO DO MINIMIZADOR GLOBAL DO SUBPROBLEMA

C
C -----
C CALCULO DO MENOR DOS VALORES DA QUADRATICA
C PARA AS DIRECOES FACTIVEIS OBTIDAS
C

```

    ISOL = 0
    QSOL = INFINITO
    IF (FGEC) THEN
        QSGEC = QUAD (BK, GK, SGEC, N)
        IF (QSGEC.LT.QSOL) THEN
            QSOL = QSGEC
            ISOL = 1
        ENDIF
    ENDIF
    IF (FLEC) THEN
        QSLEC = QUAD(BK, GK, SLEC, N)
        IF (QSLEC.LT.QSOL) THEN
            QSOL = QSLEC
            ISOL = 2
        ENDIF
    ENDIF
    IF (FGEP) THEN
        QSGEP = QUAD(BK, GK, SGEP, N)
        IF (QSGEP.LT.QSOL) THEN
            QSOL = QSGEP
            ISOL = 3
        ENDIF
    ENDIF
    IF (FLEP) THEN

```

```

      QSLEP = QUAD(BK, GK, SLEP, N)
      IF (QSLEP.LT.QSOL) THEN
        QSOL = QSLEP
        ISOL = 4
      ENDIF
    ENDIF
  IF (FG2E) THEN
    QSG2E = QUAD(BK, GK, SG2E, N)
    IF (QSG2E.LT.QSOL) THEN
      QSOL = QSG2E
      ISOL = 5
    ENDIF
  ENDIF
ELSE
  ISOL = 6
  QSI = QUAD(BK, GK, SI, N)
ENDIF
C
C  SAIDA DA SOLUCAO
C  -----
      GOTO (210, 220, 230, 240, 250, 260) ISOL
      GOTO 270
210  CONTINUE
      WRITE (8,10400)
      WRITE (*,10400)
      DO I = 1, N
        S(I) = SGEC(I)
        WRITE (8, 10410) I, SGEC(I)
      END DO
      WRITE (8,10420) QSGEC
      QS = QSGEC
      GOTO 270
220  CONTINUE
      WRITE (8,10430)
      WRITE (*,10430)
      DO I = 1, N
        S(I) = SLEC(I)
        WRITE (8,10410) I, SLEC(I)
      END DO
      WRITE (8,10420) QSLEC
      QS = QSLEC
      GOTO 270
230  CONTINUE
      WRITE (8,10440)
      WRITE (*,10440)
      DO I = 1, N
        S(I) = SGEP(I)
        WRITE (8,10410) I, SGEP(I)
      END DO
      WRITE (8,10420) QSGEP
      QS = QSGEP
      GOTO 270
240  CONTINUE
      WRITE (8,10450)
      WRITE (*,10450)

```

```

      DO I = 1, N
        S(I) = SLEP(I)
        WRITE (8,10410) I, SLEP(I)
      END DO
      WRITE (8,10420) QSLEP
      QS = QSLEP
      GOTO 270
250  CONTINUE
      WRITE (8,10460)
      WRITE (*,10460)
      DO I = 1, N
        S(I) = SG2E(I)
        WRITE (8,10410) I, SG2E(I)
      END DO
      WRITE (8,10420) QSG2E
      QS = QSG2E
      GOTO 270
260  CONTINUE
      WRITE (8,10470)
      WRITE (*,10470)
      DO I = 1, N
        S(I) = SI(I)
        WRITE (8,10410) I, SI(I)
      END DO
      WRITE (8,10420) QSI
      QS = QSI
270  CONTINUE
      RETURN
C
10010 FORMAT (1X, '*****',
$ '*****')
10020 FORMAT (1X, 'RESOLUCAO DO SUBPROBLEMA - ITERACAO K -> ', I3)
10030 FORMAT (/, ' -- I ----- XK(I) ----- GK(I)-----')
10040 FORMAT (2X, I3, 5X, 1PE14.7, 5X, 1PE14.7)
10050 FORMAT (1X, 'RAIO DE CONFIANCA -> ', 1PE14.7)
10060 FORMAT (/, 1X, 'ERRO NA ROTINA F02ABF: IFAIL = ', I3)
10070 FORMAT(/, 1X, 'MENOR AUTOVALOR DA HESSIANA BK: ', 1PE14.7, /)
10080 FORMAT (/, 1X, '***** GK ORTOGONAL A S1 ***** ', /)
10090 FORMAT (/, 1X, '***** (B * XK - GK) ORTOGONAL A S1 ***** ', /)
10100 FORMAT(/, 1X, '***** HARD CASE - ESFERA DE CONFIANCA ***** ', /)
10110 FORMAT (/, 1X, 'NAO EXISTE INTERSECAO ENTRE A VARIEDADE E A '
$ ' ESFERA DE CONFIANCA ')
10120 FORMAT (1X, 'FACTIBILIDADE DE SSEC : ', L7)
10130 FORMAT (1X, 'NEWTON (ESFERA DE CONFIANCA) CONVERGIU COM ', I3,
$ ' ITERACOES ')
10140 FORMAT (1X, 'NEWTON (ESFERA DE CONFIANCA) NAO CONVERGIU COM ',
$ ' I4, ' ITERACOES')
10150 FORMAT (1X, 'ITERACOES PARA OBTER LOCAL NAO GLOBAL (ESFERA ',
$ ' DE CONFIANCA)', /, 1X, 'FASE 1: ', I3, /, 1X, 'FASE 2: ', I3, /)
10160 FORMAT(1X, 'FACTIBILIDADE DE SLEC: ', L7)
10170 FORMAT (1X, 'NAO EXISTE MINIMIZADOR LOCAL NAO GLOBAL', /, 5X,
$ '(ESFERA DE CONFIANCA)')
10180 FORMAT (1X, 'A FASE 1 DE MINLOC (ESFERA DE CONFIANCA) NAO ',
$ ' CONVERGIU COM ', I4, ' ITERACOES ')
10190 FORMAT (1X, 'A FASE 2 DE MINLOC (ESFERA DE CONFIANCA) NAO ',

```



```

$ ' CONVERGIU COM ', I4, ' ITERACOES ' )
10200 FORMAT (1X, 'FACTIBILIDADE DE SGEP : ', L7)
10210 FORMAT(/, 1X, '***** HARD CASE - ESFERA DO PROBLEMA ***** ', /)
10220 FORMAT (/, 1X, 'NAO EXISTE INTERSECAO ENTRE A VARIEDADE E A '
$ ' ESFERA DO PROBLEMA ' )
10230 FORMAT (1X, 'NEWTON (ESFERA DO PROBLEMA) CONVERGIU COM ', I3,
$ ' ITERACOES ' )
10240 FORMAT (1X, 'NEWTON (ESFERA DO PROBLEMA) NAO CONVERGIU COM ',
$ I4, ' ITERACOES ' )
10250 FORMAT(1X, 'FACTIBILIDADE DE SLEP: ', L7)
10260 FORMAT (1X, 'ITERACOES PARA OBTER LOCAL NAO GLOBAL (ESFERA ',
$ ' DO PROBLEMA)', //, 1X, 'FASE 1: ', I4, //, 1X, 'FASE 2: ', I4)
10270 FORMAT (1X, 'NAO EXISTE MINIMIZADOR LOCAL NAO GLOBAL', //, 5X,
$ ' (ESFERA DO PROBLEMA) ' )
10280 FORMAT (1X, 'A FASE 1 DE MINLOC (ESFERA DO PROBLEMA) NAO ',
$ ' CONVERGIU COM ', I4, ' ITERACOES ' )
10290 FORMAT(1X, 'A FASE 2 DE MINLOC (ESFERA DO PROBLEMA) NAO ',
$ ' CONVERGIU COM ', I4, ' ITERACOES ' )
10300 FORMAT(1X, 'FACTIBILIDADE DE SG2E : ', L7)
10310 FORMAT (/, 1X, '(N-2)-ESFERA = UM UNICO PONTO')
10320 FORMAT(/, 1X, 'NAO EXISTE (N-2)-ESFERA')
10330 FORMAT (/, 1X, 'ESF. DE CONF. EXTERNA A ESF. DO PROB.')
10340 FORMAT (/, 1X, 'ESF. DO PROB. EXTERNA A ESF. DE CONF.')
10350 FORMAT(/, 1X, '***** (A)t (GK + BETA * B * XK) ',
$ ' ORTOGONAL A S2 *****', /)
10360 FORMAT (1X, 'NEWTON ((N-2) ESFERA) CONVERGIU COM ', I3,
$ ' ITERACOES ' )
10370 FORMAT (1X, 'NEWTON ((N-2)-ESFERA) NAO CONVERGIU COM ', I3,
$ ' ITERACOES ' )
10380 FORMAT (/, 1X, '***** HARD CASE - (N-2)-ESFERA ***** ' )
10390 FORMAT (/, 1X, 'NAO EXISTE INTERSECAO ENTRE A VARIEDADE E A '
$ ' (N-2)-ESFERA ' )
10400 FORMAT (1X, 'SOLUCAO: MINIMIZADOR GLOBAL DA ESFERA DE '
$ ' CONFIANCA', /)
10410 FORMAT (1X, 'S ( ', I3, ' ) = ', 1PE14.7)
10420 FORMAT (1X, 'VALOR MINIMO DA QUADRATICA: Q(S) = ', 1PE14.7)
10430 FORMAT(1X, 'SOLUCAO: MINIMIZADOR LOCAL DA ESFERA DE ',
$ ' CONFIANCA', /)
10440 FORMAT (1X, 'SOLUCAO: MINIMIZADOR GLOBAL DA ESFERA DO ',
$ ' PROBLEMA', /)
10450 FORMAT (1X, 'SOLUCAO: MINIMIZADOR LOCAL DA ESFERA DO ',
$ ' PROBLEMA', /)
10460 FORMAT (1X, 'SOLUCAO: MINIMIZADOR GLOBAL NA (N-2) - ESFERA ', /)
10470 FORMAT (1X, 'SOLUCAO: MINIMIZADOR IRRESTRITO')
C
END

```

```

SUBROUTINE PREMULT (A, ILIN, ICOL, V, U, TRANSP)
C
C EFETUA A MULTIPLICACAO A * V = U OU At * V = U
C
C PARAMETROS
C
C REAL*8 A(50,1), V(1), U(1)
C INTEGER ILIN, ICOL
C LOGICAL TRANSP
C
C A = matriz premultiplicadora
C V = vetor a ser multiplicado por A ou At
C U = resultado da multiplicacao
C ILIN = indice das linhas de A
C ICOL = indice das colunas de A
C TRANSP = indica se a multiplicacao sera feita com A
C           (TRANSP = .false.) ou com At (TRANSP = .true.)
C
C VARIAVEIS LOCAIS
C
C REAL*8 AUX
C INTEGER I, J
C
C AUX = acumulador na multiplicacao
C I = variacao nas linhas de A
C J = variacao nas colunas de A
C
C IF (.NOT.TRANSP)
C $ THEN
C     DO 20 I = 1, ILIN
C       AUX = 0.D0
C       DO 10 J = 1, ICOL
C         AUX = AUX + A(I,J) * V(J)
10      CONTINUE
C       U(I) = AUX
20      CONTINUE
C     ELSE
C       DO 40 J = 1, ICOL
C         AUX = 0.D0
C         DO 30 I = 1, ILIN
C           AUX = AUX + A(I,J) * V(I)
30        CONTINUE
C         U(J) = AUX
40        CONTINUE
C       ENDIF
C     RETURN
C   END
C
C
C //////////////////////////////////////
C
C
C SUBROUTINE MULTMAT (A, B, C, ILINA, ICOLA, ICOLB, ATR)
C

```

```

C      EFETUA A MULTIPLICACAO A * B = C OU (A)t * B = C
C
C      PARAMETROS
C
C      REAL*8 A(50,1), B(50,1), C(50,1)
C      INTEGER ILINA, ICOLA, ICOLB
C      LOGICAL ATR
C
C      A = matriz premultiplicadora
C      B = matriz a ser multiplicada por A ou (A)t
C      C = resultado da multiplicacao
C      ILINA = indice das linhas de A
C      ICOLA = indice das colunas de A
C      ICOLB = indice das colunas de B
C      ATR = indica se a multiplicacao sera feita com A (ATR = .false.)
C            ou com (A)t (ATR = .true.)
C
C      VARIAVEIS LOCAIS
C
C      REAL*8 SOM
C      INTEGER I, J, K
C
C      SOM = acumulador na multiplicacao
C      I,J,K = variacao nos elementos das matrizes
C
C      IF (.NOT.ATR)
C      $ THEN
C          DO 10 I = 1, ILINA
C              DO 10 J = 1, ICOLB
C                  SOM = 0.D0
C                  DO 5 K = 1, ICOLA
C                      SOM = SOM + A(I,K) * B(K,J)
C                  CONTINUE
C                  C(I,J) = SOM
C          10 CONTINUE
C      ELSE
C          DO 20 I = 1, ICOLA
C              DO 20 J = 1, ICOLB
C                  SOM = 0.D0
C                  DO 15 K = 1, ILINA
C                      SOM = SOM + A(K,I) * B(K,J)
C                  CONTINUE
C                  C(I,J) = SOM
C          20 CONTINUE
C      ENDIF
C      RETURN
C      END
C
C
C      C/////////////////////////////////////////////////////////////////
C
C      SUBROUTINE GKBK (XK, GK, BK, IPRB, N, NF, EPS, FALHA)
C
C      CALCULO DO VETOR GRADIENTE GK, NO PONTO XK, E MONTAGEM

```

```

C   DA APROXIMACAO BK PARA A MATRIZ HESSIANA, NO PONTO XK,
C   VIA DIFERENCAS FINITAS AVANÇADAS (DENNIS E SCHNABEL P.103)
C
C   PARAMETROS
C
C   REAL*8 XK(1), GK(1), BK(50,1), EPS
C   INTEGER IPRB, N, NF
C   LOGICAL FALHA
C
C   XK = K-ESIMO PONTO
C   GK = VETOR GRADIENTE DA FUNCAO OBJETIVO NO PONTO XK
C   BK = APROXIMACAO PARA A MATRIZ HESSIANA DA F.O. EM XK
C   EPS = PRECISAO PARA DIFERENCAS FINITAS
C   IPRB = INDICE DO PROBLEMA TESTE
C   N = DIMENSAO DO PROBLEMA TESTE
C   NF = DIMENSAO DO VETOR DE FUNCOES CUJO QUADRADO DA NORMA
C       EUCLIDIANA CONSTITUI A F.O. DO PROBLEMA
C   FALHA = CODIGO DE ERRO NA AVALIACAO DO GRADIENTE
C           .TRUE. -> OK      .FALSE. -> PONTO FORA DO DOMINIO
C
C   VARIAVEIS LOCAIS
C
C   REAL*8 E(50), F(50), A(50,50), H, AUX
C
C   E,F = VETORES AUXILIARES
C   A = MATRIZ AUXILIAR
C   H = INCREMENTO USADO PARA A APROXIMACAO
C   AUX = VARIAVEL AUXILIAR
C
C   SUBROTINA UTILIZADA: GRAD
C
C   CALCULO DO GRADIENTE NO PONTO XK
C
C   FALHA = .FALSE.
C   CALL GRAD (XK, GK, IPRB, N, NF, FALHA)
C   J = 1
C
C   MONTAGEM DA APROXIMACAO PARA A MATRIZ HESSIANA
C
C   DO WHILE ((J.LE.N).AND.(.NOT.FALHA))
C       IF (XK(J).EQ.0.D0) THEN
C           H = EPS
C       ELSE
C           H = EPS * XK(J)
C       ENDIF
C       DO I = 1,N
C           IF (I.EQ.J) THEN
C               E(I) = XK(I) + H
C           ELSE
C               E(I) = XK(I)
C           ENDIF
C       ENDDO
C       CALL GRAD (E, F, IPRB, N, NF, FALHA)
C       IF (.NOT.FALHA) THEN
C           DO I = 1,N

```

```

                A(I,J) = (F(I) - GK(I)) / H
            ENDDO
        ENDIF
        J = J + 1
    ENDDO
    IF (.NOT.FALHA) THEN
        DO 200 I = 1,N
            DO 200 J = 1,N
                BK(I,J) = 0.5D0 * ( A(I,J) + A(J,I) )
200          CONTINUE
            ENDIF
        RETURN
    END

C
C
C ///////////////////////////////////////////////////////////////////
C
C
C      DOUBLE PRECISION FUNCTION EUCNOR (V, IV)
C
C      CALCULA A NORMA EUCLIDIANA DO VETOR V
C
C      PARAMETROS
C
C      REAL*8 V(50)
C      INTEGER IV
C
C      V = vetor cuja norma euclidiana sera calculada
C      IV = dimensao de V
C
C      VARIAVEIS LOCAIS
C
C      REAL*8 AUX
C      INTEGER I
C
C      AUX = acumulador
C      I = variacao dos termos de V
C
C      AUX = 0.D0
C      DO I = 1, IV
C          AUX = AUX + V(I) ** 2
C      ENDDO
C      EUCNOR = DSQRT (AUX)
C      RETURN
C      END
C
C
C ///////////////////////////////////////////////////////////////////
C
C
C      DOUBLE PRECISION FUNCTION QUAD( B, G, S, N)
C
C      CALCULA O VALOR DA QUADRATICA NO PONTO S
C
C      PARAMETROS

```

```

C
REAL*8 B(50,1), G(1), S(1)
INTEGER N

C
C   R = matriz hessiana da quadratica
C   G = gradiente da funcao
C   S = direcao para a qual a quadratica sera avaliada
C   N = dimensao do subproblema
C
C   VARIAVEIS LOCAIS
C
REAL*8 A, V(50)
INTEGER I

C
C   A = acumulador
C   V = vetor auxiliar : V = BS
C   I = indice da soma
C
CALL PREMULT (B, N, N, S, V, .FALSE.)
A = 0.D0
DO 10 I = 1, N
    A = A + S(I) * ( (V(I)/2.D0) + G(I) )
10 CONTINUE
QUAD = A
RETURN
END

C
C
C //////////////////////////////////////
C
C
SUBROUTINE FIFILIN (C, U, R, N, DIM, MI, FI, FILIN, ESFERA,
$ AUTO, BETA, QUADRADO)

C
C   CALCULA AS FUNCOES FI(MI) E FILIN(MI)
C
C   PARAMETROS
C
REAL*8 C(1), U(1), R(1), MI, FI, FILIN, BETA
INTEGER N, DIM, ESFERA
LOGICAL AUTO, QUADRADO

C
C   C = vetor (QK)t GK (esferas de confianca ou do problema) ou
C       vetor (Q2 A)t GK ( (n-2)-esfera )
C   U = vetor (QK)t XK (esferas de confianca ou do problema) ou
C       vetor (Q2 A)t (BK) XK
C   R = vetor dos autovalores da matriz hessiana
C   MI = multiplicador de Lagrange
C   FI = norma-2 ou norma-2 ao quadrado da direcao S, S+XK ou Y,
C       conforme a esfera
C   FILIN = derivada de FI = d(FI)/d(MI)
C   N = dimensao do subproblema
C   DIM = dimensao do autoespaco associado ao menor autovalor da
C       hessiana
C   ESFERA = codigo do tipo de esfera:

```

```

C          1: esfera de confianca
C          2: esfera do problema
C          3: (n-2) - esfera
C      AUTO = indicador se MI = - R(1)
C      BETA = escalar utilizado na (n-2)-esfera
C      QUADRADO = indicador se FI = norma-2 ao quadrado ou nao
C
C      VARIAVEIS LOCAIS
C
C      REAL*8 S1, S2, A1, A2, A3
C      INTEGER I, J
C
C      S1, S2 = acumuladores das somas
C      A1, A2, A3 = variaveis auxiliares
C      I, J = indices das somas
C
C      S1 = 0.D0
C      S2 = 0.D0
C      IF (AUTO)
C      $ THEN
C          J = DIM + 1
C      ELSE
C          J = 1
C      ENDIF
C      DO I = J, N
C          A1 = R(I) + MI
C          GOTO (2, 4, 6) ESFERA
C          GOTO 8
C      2      CONTINUE
C          A2 = 0.D0
C          GOTO 8
C      4      CONTINUE
C          A2 = - R(I) * U(I)
C          GOTO 8
C      6      CONTINUE
C          A2 = BETA * U(I)
C      8      CONTINUE
C          A3 = ( ( C(I) + A2) / A1 ) ** 2
C          S1 = S1 + A3
C          S2 = S2 + A3 * (1.D0 / A1)
C      END DO
C      IF (QUADRADO) THEN
C          FI = S1
C          FILIN = - 2.D0 * S2
C      ELSE
C          FI = DSQRT(S1)
C          IF (FI.NE.0.D0) THEN
C              FILIN = - S2 / FI
C          ELSE
C              FILIN = 0.D0
C          ENDIF
C      ENDIF
C      RETURN
C      END
C

```

```

C
C//////////
C
C
C      SUBROUTINE NEWTON (Q, C, X, U, R, S, N, DIM, RAIO, MI0, EPS,
$ ITMAX, ESFERA, ERRO, KONT, BETA)
C
C      CALCULA A RAIZ DA EQUACAO 1/FI (MI0) = 1/RAIO , MI0 > -R(1),
C      ONDE FI (MI0) = NORMA-2 ( DIRECAO ) ,
C      USANDO O METODO DE NEWTON COM A MODIFICACAO DE REINSCH-HEBDEN
C
C      PARAMETROS
C
C      REAL*8 Q(50,1), C(1), X(1), U(1), R(1), S(1),
$ RAIO, MI0, EPS, BETA
C      INTEGER N, DIM, ITMAX, ESFERA, ERRO, KONT
C
C      Q = base ortonormal dos autovetores de BK ou B2
C      C = vetor Qt GK ou (Q2 A)t GK
C      X = vetor da mudanca de variaveis (Z = S + X) na esfera do
C      problema (X = 0 na esfera de confianca)
C      U = vetor Qt X ou (Q2 A)t (BK) X
C      R = vetor dos autovalores de BK ou B2 ( resp. RK ou R2)
C      S = direcao associada ao multiplicador MI0, obtido como solucao
C      RAIO = raio da esfera (de confianca, do problema, da (n-2)-esf.)
C      MI0 = multiplicador inicial/ retorna com mult. da solucao global
C      EPS = precisao com que se obtem a raiz
C      BETA = escalar utilizado na (n-2)-esfera
C      N = dimensao da direcao ( N ou N-1 )
C      DIM = dimensao do autoespaco associado ao menor autovalor do
C      hessiano BK ou B2
C      ITMAX = numero maximo de iteracoes
C      ESFERA = codigo do tipo de esfera:
C              1: esfera de confianca
C              2: esfera do problema
C              3: (n-2) - esfera
C      ERRO = codigo de saida da rotina NEWTON
C              1: sucesso
C              2: KONT > MAXNEW
C      KONT = contador de iteracoes / retorna com o numero de iteracoes
C      efetuado
C
C      VARIAVEIS LOCAIS
C
C      REAL*8 FIT, FI, FILIN, VC(50), AUX
C      INTEGER I
C
C      FIT = funcao de iteracao no metodo de Newton/modificacao
C      de Reinsch-Hebden
C      FI = norma-2 da direcao S, S + X ou Y, conforme a esfera
C      FILIN = derivada de FI
C      VC = vetor auxiliar no calculo da direcao S
C      AUX = variavel auxiliar
C      I = indices dos vetores
C

```



```

C
C      INICIALIZA CONTADOR
C
C      KONT = 0
C
C      OBTENCAO DO PONTO INICIAL MI0 TAL QUE 1/FI(MI0) (= 1/RAIO,
C      VIA BISSECAO
C
C      INTERVALO INICIAL
C
C      AUX = MI0 + R(1)
C
C      CALCULO INICIAL DA FUNCAO FI
C
C      CALL FIFILIN (C, U, R, N, DIM, MI0, FI, FILIN, ESFERA, .FALSE.,
$      BETA, .FALSE.)
C
C      LOOP DA BISSECAO
C
C      DO WHILE (FI.LT.RAIO)
C          AUX = AUX / 2.D0
C          MI0 = -R(1) + AUX
C          CALL FIFILIN (C, U, R, N, DIM, MI0, FI, FILIN, ESFERA,
$          .FALSE., BETA, .FALSE.)
C      END DO
C
C      FUNCAO DE ITERACAO INICIAL (REINSCH-HEBDEN)
C
C      FIT = ((FI - RAIO) * FI) / (RAIO * FILIN)
C
C      LOOP DE NEWTON
C
100    IF((DABS(FI-RAIO).LT.EPS).OR.(KONT.GE.ITMAX)) GOTO 120
C          AUX = DABS(FI - RAIO)
C          MI0 = MI0 - FIT
C          CALL FIFILIN (C, U, R, N, DIM, MI0, FI, FILIN, ESFERA,
$          .FALSE., BETA, .FALSE.)
C          FIT = ((FI - RAIO) * FI) / (RAIO * FILIN)
C          KONT = KONT + 1
C          GOTO 100
120    CONTINUE
C
C      IF (KONT.LT.ITMAX) THEN
C
C          NEWTON CONVERGIU / CALCULO DA DIRECAO
C
C          DO I = 1, N
C              GOTO (10, 20, 30) ESFERA
C              GOTO 40
10          CONTINUE
C              AUX = 0.D0
C              GOTO 40
20          CONTINUE
C              AUX = R(I) * U(I)
C              GOTO 40

```

```

30      CONTINUE
      AUX = - BETA * U(I)
40      CONTINUE
      VC(I) = (AUX - C(I)) / ( R(I) + MI0 )
      END DO

C
      CALL PREMULT (Q, N, N, VC, S, .FALSE.)

C
C     NESTE MOMENTO, QUANDO NEWTON E' CHAMADO NA ESFERA DO PROBLEMA,
C     S = S + X ;
C     RECUPERANDO A DIRECAO PURA S, QUE RETORNA PARA O PROG. PRINCIPAL:
C
      IF (ESFERA.EQ.2) THEN
        DO I = 1, N
          S(I) = S(I) - X(I)
        END DO
      ENDIF

C
      ELSE

C
C     NEWTON NAO CONVERGIU
C
      ERRO = 2
      ENDIF
      RETURN
      END

C
C
C //////////////////////////////////////
C
C
      SUBROUTINE MINLOC (Q, C, X, U, R, S, N, RAIO, MI, EPS, ESFERA,
$   ERRO, DIM, ITM1, ITM2, KF1, KF2, EMACH)

C
C     CALCULA A RAIZ DA EQUACAO FI (MI) = RAIO ** 2 ,
C     PARA MI EM [-R(2), -R(1)] E TAL QUE FILIN (MI) >= EPS,
C     ONDE FI (MI) = (NORMA-2 ( DIRECAO )) ** 2,
C     USANDO O METODO DE MARTINEZ [1989]
C
C     PARAMETROS
C
C     REAL*8 Q(50,1), C(1), X(1), U(1), R(1), S(1), RAIO, MI,
$   EPS, EMACH
C     INTEGER N, ESFERA, ERRO, DIM, ITM1, ITM2, KF1, KF2

C
C     Q = base ortonormal dos autovetores de BK
C     C = vetor Qt GK
C     X = vetor da mudanca de variaveis (Z = S + X) na esfera do
C         problema (X = 0 na esfera de confianca)
C     U = vetor Qt X
C     R = vetor dos autovalores de BK (ordem crescente)
C     S = direcao associada ao multiplicador MI obtido como solucao
C     RAIO = raio da esfera (de confianca, do problema, da (n-2)-esf.)
C     MI = multiplicador inicial(correspondente a sol. global)
C         ( retorna com multiplicador da solucao S )

```

```

C     EPS = precisao com que se obtem a raiz
C     EMACH = precisao da maquina
C     N = dimensao do subproblema
C     ESFERA = codigo do tipo de esfera:
C             1: esfera de confianca
C             2: esfera do problema
C     ERKO = codigo de saida da rotina MINLOC
C             1: sucesso em MINLOC
C             2: nao existe minimizador local nao global na esfera
C             3: KF1 > ITM1
C             4: KF2 > ITM2
C     DIM = dimensao do autoespaco associado ao menor autovalor de BK
C     ITM1 = numero maximo de iteracoes para a fase 1
C     ITM2 = idem para a fase 2
C     KF1 = retorna com o numero de iteracoes efetuado na fase 1
C     KF2 = idem para a fase 2
C
C     VARIAVEIS LOCAIS
C
C     REAL*8 VC(50), Miant, MIUPP, MILOW, MIAUX, MINEW, CCUPP, CLOW,
$     T, FI, FILIN, QR, INF, SUP, AUX, INTERV, FAUX, FLAUX
C     INTEGER I
C
C     VC = vetor auxiliar no calculo da direcao S
C     Miant = multiplicador da etapa anterior (= MI (KF1 - 1))
C     MIUPP = multiplicador que limita superiormente o intervalo
C     MILOW = multiplicador que limita inferiormente o intervalo
C     MIAUX = multiplicador auxiliar
C     MINEW = multiplicador obtido pela funcao de Newton para
C             determinar a raiz de FI ( MI ) = QR
C     CCUPP = combinacao convexa para limite superior
C     CLOW = combinacao convexa para limite inferior
C     T = parametro da combinacao convexa
C     FI = quadrado da norma-2 da direcao S ou S + X, conforme a esfera
C     FILIN = derivada de FI em relacao a MI
C     QR = quadrado do raio
C     INF = extremo inferior de intervalo
C     SUP = extremo superior de intervalo
C     AUX = variavel auxiliar
C     INTERV = comprimento do intervalo
C     FAUX = valor da funcao FI(MIAUX)
C     FLAUX = valor da funcao FILIN(MIAUX)
C     I = indices dos vetores
C
C     COMBINACOES CONVEXAS (FUNCOES DE COMANDO)
C
C     CLOW(T,MI,MILOW) = MILOW + T * (MI - MILOW)
C     CCUPP(T,MI,MIUPP) = MI + T * (MIUPP - MI)
C
C     INICIALIZA CONTADORES
C
C     KF1 = 0
C     KF2 = 0
C
C     EXTREMO SUPERIOR DO INTERVALO INICIAL: MULTIPLICADOR

```

```

C      CORRESPONDENTE A SOLUCAO GLOBAL
C
C      MIUPP = MI
C
C      INICIALIZA MULTIPLICADOR ANTERIOR
C      (CANDIDATO A EXTREMO INFERIOR DO INTERVALO NA FASE 2)
C
C      MIAUT = - 0.75D0 * R(2) - 0.25 * R(1)
C
C      MULTIPLICADOR INICIAL: PONTO MEDIO DO INTERVALO [-R(2),-R(1)]
C
C      MI = - ( R(1) + R(2) ) / 2.D0
C
C      QUADRADO DO RAIO
C
C      QR = RAO ** 2
C
C      COMPRIMENTO INICIAL DO INTERVALO
C
C      INTERV = MIUPP - MI
C
C      CALCULO INICIAL DAS FUNCOES FI E FILIN
C
C      CALL FIFILIN (C, U, R, N, DIM, MI, FI, FILIN, ESFERA, .FALSE.,
$      0.D0, .TRUE.)
C
C      FASE 1
C      OBTENCAO DE MI TAL QUE FI(MI) >= QR E FILIN(MI) >= 0
C
C      DO WHILE (((FI.LT.QR).OR.(FILIN.LT.0.D0)).AND.(INTERV.GT.EHACH)
$      .AND.(KF1.LT.ITM1))
C          MIAUX = CCUPP (5.0D-1, MI, MIUPP)
C          IF ((R(1)+MIAUX).GE.0.D0) THEN
C              MIUPP = MIAUX
C          ELSE
C              MIAUT = MI
C              MI = MIAUX
C              CALL FIFILIN (C, U, R, N, DIM, MI, FI, FILIN, ESFERA,
$              .FALSE., 0.D0, .TRUE.)
C          ENDIF
C          INTERV = MIUPP - MI
C          KF1 = KF1 + 1
C      END DO
C
C      IF (KF1.EQ.ITM1) THEN
C          ERRO = 3
C      ELSE
C
C          FASE 2
C          EXTREMO INFERIOR DO INTERVALO INICIAL
C
C          MILOW = MIAUT
C
C          COMPRIMENTO INICIAL DO INTERVALO

```

INTERV = MI - MILOW

OBTENCAO DE MI TAL QUE FI(MI) = QR E FILIN(MI) >= 0,
VIA BISSECAO

```
C
C
C
C
DO WHILE ((ERRO.EQ.1).AND.((DABS(FI-QR).GT.EPS).OR.
$ (FILIN.LT.0.D0)).AND.(INTERV.GT.EMACH).AND.(KF2.LT.ITM2))
  IF (FILIN.EQ.0.D0) THEN
    ERRO = 2
  ELSE
    MINEW = MI - (FI - QR)/FILIN
    IF (MINEW.LT.MILOW) THEN
      ERRO = 2
    ELSE
      INF = CCLOW (1.0D-1, MI, MILOW)
      SUP = CCLOW (9.0D-1, MI, MILOW)
      IF ((MINEW.GE.INF).AND.(MINEW.LE.SUP)) THEN
        MIAUX = MINEW
      ELSE
        MIAUX = CCLOW (5.0D-1, MI, MILOW)
      ENDIF
      CALL FIFILIN (C, U, R, N, DIM, MIAUX, FAUX, FLAUX,
$     ESFERA, .FALSE., 0.D0, .TRUE.)
      IF ((FAUX.GE.QR).AND.(FLAUX.GE.0.D0)) THEN
        MI = MIAUX
        FI = FAUX
        FILIN = FLAUX
      ELSE
        MILOW = MIAUX
      ENDIF
      INTERV = MI - MILOW
      KF2 = KF2 + 1
    ENDIF
  ENDIF
END DO
```

```
C
IF (ERRO.NE.2) THEN
  IF (KF2.EQ.ITM2) THEN
    ERRO = 4
  ELSE
```

CALCULO DA DIRECAO S

```
C
C
C
DO I = 1, N
  IF (ESFERA.EQ.1) AUX = 0.D0
  IF (ESFERA.EQ.2) AUX = R(I) * U(I)
  VC(I) = (AUX - C(I)) / (R(I) + MI)
END DO
CALL PREMULT ( Q, N, N, VC, S, .FALSE.)
```

C
C
C
C
C
C
NESTE MOMENTO, QUANDO MINLOC E' CHAMADA NA ESFERA DO
PROBLEMA, S = S + X; RECUPERANDO A DIRECAO FURA S,
QUE RETORNA PARA O PROG. PRINCIPAL:

```
IF (ESFERA.EQ.2) THEN
```

```
      DO I = 1, N
        S(I) = S(I) - X(I)
      END DO
    ENDIF
  ENDIF
ENDIF
RETURN
END
```

```
C
C
C////////////////////////////////////
```

```

SUBROUTINE GRAD (X , G, IPRB, N, NF, ERRO)
C
C  CALCULO DO VETOR GRADIENTE (G) NO PONTO X
C  PARA O PROBLEMA TESTE IPRB
C
C  PARAMETROS
C
C  REAL*8 X(1), G(1)
C  INTEGER IPRB, N, NF
C  LOGICAL ERRO
C
C  X = PONTO NO QUAL O GRADIENTE SERA AVALIADO
C  G = VETOR GRADIENTE
C  IPRB = INDICE DO PROBLEMA TESTE
C  N = DIMENSAO DO PROBLEMA TESTE
C  NF = DIMENSAO DO VETOR DE FUNCOES CUJO QUADRADO DA NORMA
C      EUCLIDIANA CONSTITUI A FUNCAO OBJETIVO DO PROBLEMA TESTE
C  ERRO = INDICA FALHA NA EXECUCAO (PONTO FORA DO DOMINIO)
C
C  VARIAVEIS LOCAIS
C
C  REAL*8 F(100), Y(50), DF(100,50),
C  $  YI, TI, A1, A2, A3, A4, A5, A6
C  INTEGER I, J, K
C
C  F = VETOR DAS FUNCOES
C  Y = VETOR DE CONSTANTES
C  DF = MATRIZ JACOBIANA
C  YI, TI = CONSTANTES DAS FUNCOES
C  A1,...,A6 = CONSTANTES AUXILIARES
C  I,J,K = INDICES DOS VETORES
C
C  GOTO (100, 200, 300, 400, 500, 600, 700, 800, 900, 1000,1100,
C  $  1200, 1300, 1400, 1500, 1600, 1700, 1800, 1900) IPRB
C  GOTO 9999
100  CONTINUE
C  FUNCAO DE ROSENBROCK
C  F(1) = 10.D0 * (X(2) - X(1)**2)
C  DF(1,1) = -20.D0 * X(1)
C  DF(1,2) = 10.D0
C  F(2) = 1.D0 - X(1)
C  DF(2,1) = -1.D0
C  DF(2,2) = 0.D0
C  GOTO 8000
C-----
200  CONTINUE
C  FUNCAO DE FREUDENSTEIN E ROTH
C  F(1) = -13.D0 + X(1) + ((5.D0 - X(2))*X(2) - 2.D0)*X(2)
C  DF(1,1) = 1.D0
C  DF(1,2) = X(2)*(10.D0 - 3.D0*X(2)) - 2.D0
C  F(2) = -29.D0 + X(1) + ((X(2) + 1.D0)*X(2) - 14.D0)*X(2)
C  DF(2,1) = 1.D0
C  DF(2,2) = X(2)*(3.D0*X(2) + 2.D0) - 14.D0
C  GOTO 8000

```

```

C
300  CONTINUE
C      HELICAL VALLEY FUNCTION
      A1 = 50.D0/3.1415926D0
      IF (X(1).LT.0.D0) THEN
        F(1) = 10.D0 * X(3) - A1 * DATAN (X(2)/X(1)) - 50.D0
      ELSE IF (X(1).GT.0.D0) THEN
        F(1) = 10.D0 * X(3) - A1 * DATAN (X(2)/X(1))
      ELSE
        ERRO = .TRUE.
      ENDIF
      IF (.NOT.ERRO) THEN
        A2 = X(1) ** 2 + X(2) ** 2
        A3 = DSQRT(A2)
        F(2) = 10.D0 * A3 - 10.D0
        F(3) = X(3)
        IF (A2.NE.0.D0) THEN
          DF(1,1) = A1 * X(2) / A2
          DF(1,2) = - A1 * X(1) / A2
          DF(1,3) = 10.D0
          DF(2,1) = 10.D0 * X(1) / A3
          DF(2,2) = 10.D0 * X(2) / A3
          DF(2,3) = 0.D0
          DF(3,1) = 0.D0
          DF(3,2) = 0.D0
          DF(3,3) = 1.D0
        ELSE
          ERRO = .TRUE.
        ENDIF
      ENDIF
      GOTO 8000
C-----
400  CONTINUE
C      FUNCAO EXP6 DE BIGGS
      DO 450 I = 1, NF
        TI = 1.D-1 * I
        YI = DEXP(-TI) - 5.D0*DEXP(-10.D0*TI) + 3.D0*DEXP(-4.D0*TI)
        A1 = DEXP (-TI*X(1))
        A2 = DEXP (-TI*X(2))
        A3 = DEXP (-TI*X(5))
        A4 = X(3) * A1
        A5 = X(4) * A2
        A6 = X(6) * A3
        F(I) = A4 - A5 + A6 - YI
        DF(I,1) = -TI * A4
        DF(I,2) = TI * A5
        DF(I,3) = A1
        DF(I,4) = - A2
        DF(I,5) = - TI * A6
        DF(I,6) = A3
      450  CONTINUE
      GOTO 8000
C-----
500  CONTINUE
C      FUNCAO GAUSSIANA

```



```

Y(1) = 0.0009D0
Y(2) = 0.0044D0
Y(3) = 0.0175D0
Y(4) = 0.0540D0
Y(5) = 0.1295D0
Y(6) = 0.2420D0
Y(7) = 0.3521D0
Y(8) = 0.3989D0
Y(9) = Y(7)
Y(10) = Y(6)
Y(11) = Y(5)
Y(12) = Y(4)
Y(13) = Y(3)
Y(14) = Y(2)
Y(15) = Y(1)
DO 550 I = 1,NF
  TI = (8.D0 - I)/2.D0
  A1 = - ((TI - X(3)) ** 2)/2.D0
  A2 = DEXP (X(2) * A1)
  A3 = X(1) * A2
  F(I) = A3 - Y(I)
  DF(I,1) = A2
  DF(I,2) = A3 * A1
  DF(I,3) = A3 * X(2) * (TI - X(3))
550  CONTINUE
     GOTO 8000

```

```

600  CONTINUE
C    FUNCAO HAL ESCALADA DE POWELL
  A1 = DEXP(-X(1))
  A2 = DEXP(-X(2))
  A3 = 1.0D+4
  F(1) = A3 * X(1) * X(2) - 1.D0
  F(2) = A1 + A2 - 1.0001D0
  DF(1,1) = A3 * X(2)
  DF(1,2) = A3 * X(1)
  DF(2,1) = - A1
  DF(2,2) = - A2
  GOTO 8000

```

```

700  CONTINUE
C    FUNCAO DA CAIXA TRIDIMENSIONAL
  DO 750 I = 1,NF
    TI = 1.D-1 * I
    A1 = DEXP (-TI * X(1))
    A2 = DEXP (-TI * X(2))
    A3 = DEXP (-TI) - DEXP (-10.D0 * TI)
    F(I) = A1 - A2 - X(3) * A3
    DF(I,1) = - TI * A1
    DF(I,2) = TI * A2
    DF(I,3) = - A3
  750  CONTINUE
     GOTO 8000

```

```

800  CONTINUE

```

```

C      FUNCAO VARIAVELMENTE DIMENSIONADA
      DO 820 I = 1,N
        F(I) = X(I) - 1.D0
        A1 = 0.D0
        DO 820 J = 1,N
          DF(I,J) = 1.D0
          A1 = A1 + J * (X(J) - 1.D0)
820    CONTINUE
      F(N+1) = A1
      F(N+2) = A1 ** 2
      DO 850 J = 1,N
        DF((N+1),J) = J * 1.D0
        DF((N+2),J) = 2.D0 * J * A1
850    CONTINUE
      GOTO 8000

C-----
900  CONTINUE
C      FUNCAO DE WATSON
      DO 930 I = 1,29
        TI = I/29.D0
        A1 = 0.D0
        A2 = 0.D0
        DO 920 J = 1,N
          A3 = TI ** (J-2)
          A1 = A1 + (J-1) * X(J) * A3
          A2 = A2 + X(J) * A3 * TI
920    CONTINUE
        F(I) = A1 - A2 ** 2 - 1.D0
        DO 930 J = 1,N
          A3 = TI ** (J-2)
          DF(I,J) = (J-1) * A3 - 2.D0 * A3 * TI * A2
930    CONTINUE
        F(30) = X(1)
        F(31) = X(2) - X(1) ** 2 - 1.D0
        DF(30,1) = 1.D0
        DF(31,1) = -2.D0 * X(1)
        DF(31,2) = 1.D0
        DO 950 J = 2,N
          DF(30,J) = 0.D0
          IF((J+1).LE.N) DF(31,(J+1)) = 0.D0
950    CONTINUE
      GOTO 8000

C-----
1000 CONTINUE
C      FUNCAO PENALIDADE I
      A1 = DSQRT(1.0D-5)
      DO 1020 I = 1,N
        F(I) = A1 * (X(I) - 1.D0)
        DO 1020 J = 1,N
          DF(I,J) = A1
1020  CONTINUE
      A2 = 0.D0
      DO 1050 J = 1,N
        A2 = A2 + X(J) ** 2
        DF((N+1),J) = 2.D0 * X(J)

```

```

1050    CONTINUE
      F(N+1) = A2 - 0.25D0
      GOTO 8000

```

```

1100    CONTINUE

```

```

C      FUNCAO PENALIDADE II

```

```

      A1 = DSQRT(1.0D-5)

```

```

      F(1) = X(1) - 0.2D0

```

```

      DF(1,1) = 1.0D0

```

```

      DO 1120 J = 2,N

```

```

        DF(1,J) = 0.0D0

```

```

1120    CONTINUE

```

```

      DO 1130 I = 2,N

```

```

        A2 = DEXP (1.0D-1 * X(I))

```

```

        A3 = DEXP (1.0D-1 * X(I-1))

```

```

        YI = DEXP (1.0D-1 * I) + DEXP (1.0D-1 * (I-1))

```

```

        F(I) = A1 * (A2 + A3 - YI)

```

```

        DO 1130 J = 1,N

```

```

          IF ((J.NE.(I-1)).AND.(J.NE.I))

```

```

$         THEN

```

```

            DF(I,J) = 0.0D0

```

```

          ELSE

```

```

            IF (J.EQ.(I-1)) DF(I,J) = 1.0D-1 * A1 * A3

```

```

            IF (J.EQ.I) DF(I,J) = 1.0D-1 * A1 * A2

```

```

          ENDIF

```

```

1130    CONTINUE

```

```

      A2 = DEXP (-1.0D-1)

```

```

      DO 1140 I = (N+1), (2*N-1)

```

```

        A3 = DEXP (1.0D-1 * X(I-N+1))

```

```

        F(I) = A1 * (A3 - A2)

```

```

        DO 1140 J = 1,N

```

```

          IF (J.EQ.(I-N+1))

```

```

$         THEN

```

```

            DF(I,J) = 1.0D-1 * A1 * A3

```

```

          ELSE

```

```

            DF(I,J) = 0.0D0

```

```

          ENDIF

```

```

1140    CONTINUE

```

```

      A2 = 0.0D0

```

```

      DO 1150 J = 1,N

```

```

        DF(2*N,J) = 2.0D0*(N-J+1)*X(J)

```

```

        A2 = A2 + (N-J+1) * (X(J) ** 2)

```

```

1150    CONTINUE

```

```

      F(2*N) = A2 - 1.0D0

```

```

      GOTO 8000

```

```

1200    CONTINUE

```

```

C      FUNCAO MAL ESCALADA DE BROWN

```

```

      F(1) = X(1) - 1.0D+6

```

```

      F(2) = X(2) - 2.0D-6

```

```

      F(3) = X(1) * X(2) - 2.0D0

```

```

      DF(1,1) = 1.0D0

```

```

      DF(1,2) = 0.0D0

```

```

      DF(2,1) = 0.0D0

```

```

      DF(2,2) = 1.0D0

```

```

DF(3,1) = X(2)
DF(3,2) = X(1)
GOTO 8000

```

C-----

1300 CONTINUE

C FUNCAO DE BROWN E DENNIS

```

DO 1350 I = 1,NF
  TI = 1/5.D0
  A1 = DEXP (TI)
  A2 = DSIN (TI)
  A3 = DCOS (TI)
  A4 = X(1) + TI * X(2) - A1
  A5 = X(3) + X(4) * A2 - A3
  F(I) = A4 ** 2 + A5 ** 2
  DF(I,1) = 2.D0 * A4
  DF(I,2) = 2.D0 * TI * A4
  DF(I,3) = 2.D0 * A5
  DF(I,4) = 2.D0 * A2 * A5

```

1350 CONTINUE
GOTO 8000

C-----

1400 CONTINUE

C GULF RESEARCH AND DEVELOPMENT FUNCTION

```

IF ((X(1).NE.0.D0).AND.(X(2).NE.0.D0)) THEN
  DO I = 1,NF
    TI = 1.D-2 * I
    YI = (-50.D0 * DLOG(TI)) ** (2.D0/3.D0) + 25.D0
    A1 = DABS (YI * NF * I * X(2))
    A2 = A1 ** X(3)
    A3 = -A2/X(1)
    A4 = DEXP(A3)
    F(I) = A4 - TI
    DF(I,1) = A4 * (-A3/X(1))
    DF(I,2) = A4 * A3 * X(3) / X(2)
    DF(I,3) = A4 * A3 * DLOG(A1)
  END DO
ELSE
  ERRO = .TRUE.
ENDIF
GOTO 8000

```

C-----

1500 CONTINUE

C FUNCAO TRIGONOMETRICA TRANSLADADA DE (1,1,...,1)

```

A1 = 0.D0
DO 1520 J = 1,N
  Y(J) = DSIN(X(J) - 1.0D0)
  A1 = A1 + DCOS(X(J) - 1.0D0)

```

1520 CONTINUE

```

DO 1550 I = 1,NF
  A2 = DCOS(X(I)- 1.0D0)
  F(I) = N - A1 + I * (1.D0 - A2) - Y(I)
  DO 1550 J = 1,N
    IF (J.EQ.I)
      THEN
        DF(I,J) = (1.D0 + I) * Y(I) - A2

```

\$

```

        ELSE
            DF(I,J) = Y(J)
        ENDIF
1550    CONTINUE
        GOTO 8000
C-----
1600    CONTINUE
C      FUNCAO DE ROSENBROCK ESTENDIDA
        DO 1650 I = 1, (NF/2)
            K = 2 * I - 1
            F(K) = 10.D0 * (X(K+1) - X(K) ** 2)
            F(K+1) = 1 - X(K)
            DO 1650 J = 1, N
                IF ((J.NE.K).AND.(J.NE.(K+1)))
$                 THEN
                    DF(K,J) = 0.D0
                    DF((K+1),J) = 0.D0
                ELSE
                    IF (J.EQ.K)
$                     THEN
                        DF(K,J) = -20.D0 * X(J)
                        DF((K+1),J) = -1.D0
                    ENDIF
                    IF (J.EQ.(K+1))
$                     THEN
                        DF(K,J) = 10.D0
                        DF((K+1),J) = 0.D0
                    ENDIF
                ENDIF
            ENDIF
1650    CONTINUE
        GOTO 8000
C-----
1700    CONTINUE
C      FUNCAO SINGULAR DE POWELL ESTENDIDA TRANSLADADA DE (1,1,...,1)
        A1 = DSQRT(5.0D+0)
        A2 = DSQRT(1.0D+1)
        DO 1750 I = 1, (N/4)
            K = 4 * I - 3
            F(K) = X(K) + 10.D0 * X(K+1) + 11.D0
            F(K+1) = A1 * (X(K+2) - X(K+3))
            F(K+2) = (X(K+1) - 2.D0 * X(K+2) - 1.D0) ** 2
            F(K+3) = A2 * (X(K) - X(K+3)) ** 2
            DO 1750 J = 1, N
                IF (J.EQ.K) THEN
                    DF(K,J) = 1.D0
                    DF((K+1),J) = 0.D0
                    DF((K+2),J) = 0.D0
                    DF((K+3),J) = 2.D0 * A2 * (X(K) - X(K+3))
                ELSE IF (J.EQ.(K+1)) THEN
                    DF(K,J) = 10.D0
                    DF((K+1),J) = 0.D0
                    DF((K+2),J) = 2.D0 * (X(K+1) - 2.D0 * X(K+2) - 1.D0)
                    DF((K+3),J) = 0.D0
                ELSE IF (J.EQ.(K+2)) THEN
                    DF(K,J) = 0.D0

```

```

      DF((K+1),J) = A1
      DF((K+2),J) = -4.D0*(X(K+1)-2.D0*X(K+2)-1.D0)
      DF((K+3),J) = 0.D0
      ELSE IF (J.EQ.(K+3)) THEN
        DF(K,J) = 0.D0
        DF((K+1),J) = - A1
        DF((K+2),J) = 0.D0
        DF((K+3),J) = -2.D0*A2*(X(K) - X(K+3))
      ELSE
        DF(K,J) = 0.D0
        DF((K+1),J) = 0.D0
        DF((K+2),J) = 0.D0
        DF((K+3),J) = 0.D0

```

```

      ENDIF

```

```

1750  CONTINUE
      GOTO 8000

```

```

C-----
1800  CONTINUE

```

```

C      FUNCAO DE BEALE
      IF (X(2).NE.0.D0) THEN
        Y(1) = 1.5D0
        Y(2) = 2.25D0
        Y(3) = 2.625D0
        DO I = 1,NF
          A1 = X(2) ** (I-1)
          A2 = A1 * X(2)
          F(I) = Y(I) - X(1) * (1.D0 - A2)
          DF(I,1) = A2 - 1.D0
          DF(I,2) = I * X(1) * A1
        END DO
      ELSE
        ERRO = .TRUE.
      ENDIF
      GOTO 8000

```

```

C-----
1900  CONTINUE

```

```

C      FUNCAO DE WOOD
      A1 = DSQRT(9.0D+1)
      A2 = DSQRT(1.0D+1)
      A3 = 1.D0/A2
      F(1) = 10.D0 * (X(2) - X(1) ** 2)
      F(2) = 1.D0 - X(1)
      F(3) = A1 * (X(4) - X(3) ** 2)
      F(4) = 1.D0 - X(3)
      F(5) = A2 * (X(2) + X(4) - 2.D0)
      F(6) = A3 * (X(2) - X(4))
      DF(1,1) = -20.D0 * X(1)
      DF(1,2) = 10.D0
      DF(1,3) = 0.D0
      DF(1,4) = 0.D0
      DF(2,1) = -1.D0
      DF(2,2) = 0.D0
      DF(2,3) = 0.D0
      DF(2,4) = 0.D0
      DF(3,1) = 0.D0

```

```

DF(3,2) = 0.D0
DF(3,3) = -2.D0 * A1 * X(3)
DF(3,4) = A1
DF(4,1) = 0.D0
DF(4,2) = 0.D0
DF(4,3) = -1.D0
DF(4,4) = 0.D0
DF(5,1) = 0.D0
DF(5,2) = A2
DF(5,3) = 0.D0
DF(5,4) = A2
DF(6,1) = 0.D0
DF(6,2) = A3
DF(6,3) = 0.D0
DF(6,4) = -A3
GOTO 8000

```

```

C-----
8000  CONTINUE
      DO 8200 J = 1,N
        A1 = 0.D0
        DO 8100 I = 1, NF
          A1 = A1 + F(I) * DF(I,J)
6100   CONTINUE
        G(J) = 2.D0 * A1
8200  CONTINUE
9999  RETURN
      END

```

```

C
C////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
C

```

```

      DOUBLE PRECISION FUNCTION FUNC (X, IPRB, N, NF, ERRO)

```

```

C
C      CALCULO DO VALOR DA FUNCAO OBJETIVO DO PROBLEMA TESTE IPRB
C      NO PONTO X
C

```

```

C      PARAMETROS
C

```

```

      REAL*8 X(1)
      INTEGER IPRB, N, NF
      LOGICAL ERRO

```

```

C
C      X = PONTO NO QUAL A FUNCAO SERA AVALIADA
C      IPRB = INDICE DO PROBLEMA TESTE
C      N = DIMENSAO DO PROBLEMA TESTE
C      NF = DIMENSAO DO VETOR DE FUNCOES CUJO QUADRADO DA NORMA
C           EUCLIDIANA CONSTITUI A FUNCAO OBJETIVO DO PROBLEMA TESTE
C      ERRO = INDICA FALHA NA EXECUCAO (PONTO FORA DO DOMINIO)
C

```

```

C      VARIAVEIS LOCAIS
C

```

```

      REAL*8 F(100), Y(50), Y1, T1, A1, A2, A3, A4, A5, A6
      INTEGER I, J, K

```

```

C
C      F = VETOR DAS FUNCOES
C      Y = VETOR DE CONSTANTES

```

```

C      Y1, I1 = CONSTANTES DAS FUNCOES
C      A1,...,A6 = CONSTANTES AUXILIARES
C      I,J,K = INDICES DOS VETORES
C
C      GOTO (100, 200, 300, 400, 500, 600, 700, 800, 900, 1000,1100,
$      1200, 1300, 1400, 1500, 1600, 1700, 1800, 1900) IPRB
      GOTO 9000
100  CONTINUE
C      FUNCAO DE ROSENBROCK
      F(1) = 10.D0 * (X(2) - X(1)**2)
      F(2) = 1.D0 - X(1)
      GOTO 8000
C-----
200  CONTINUE
C      FUNCAO DE FREUDENSTEIN E ROTH
      F(1) = -13.D0 + X(1) + ((5.D0 - X(2))*X(2) - 2.D0)*X(2)
      F(2) = -29.D0 + X(1) + ((X(2) + 1.D0)*X(2) - 14.D0)*X(2)
      GOTO 8000
C-----
300  CONTINUE
C      HELICAL VALLEY FUNCTION
      A1 = 50.D0/3.1415926D0
      IF (X(1).LT.0.D0) THEN
          F(1) = 10.D0 * X(3) - A1 * DATAN (X(2)/X(1)) - 50.D0
      ELSE IF (X(1).GT.0.D0) THEN
          F(1) = 10.D0 * X(3) - A1 * DATAN (X(2)/X(1))
      ELSE
          ERRO = .TRUE.
      ENDIF
      IF (.NOT.ERRO) THEN
          A2 = X(1) ** 2 + X(2) ** 2
          A3 = DSQRT(A2)
          F(2) = 10.D0 * A3 - 10.D0
          F(3) = X(3)
      ENDIF
      GOTO 8000
C-----
400  CONTINUE
C      FUNCAO EXP6 DE BIGGS
      DO 450 I = 1, NF
          TI = 1.D-1 * I
          YI = DEXP(-TI) - 5.D0*DEXP(-10.D0*TI) + 3.D0*DEXP(-4.D0*TI)
          A1 = DEXP (-TI*X(1))
          A2 = DEXP (-TI*X(2))
          A3 = DEXP (-TI*X(5))
          A4 = X(3) * A1
          A5 = X(4) * A2
          A6 = X(6) * A3
          F(I) = A4 - A5 + A6 - YI
      450  CONTINUE
      GOTO 8000
C-----
500  CONTINUE
C      FUNCAO GAUSSIANA

```



```

Y(1) = 0.0009D0
Y(2) = 0.0044D0
Y(3) = 0.0175D0
Y(4) = 0.0540D0
Y(5) = 0.1295D0
Y(6) = 0.2420D0
Y(7) = 0.3521D0
Y(8) = 0.3989D0
Y(9) = Y(7)
Y(10) = Y(6)
Y(11) = Y(5)
Y(12) = Y(4)
Y(13) = Y(3)
Y(14) = Y(2)
Y(15) = Y(1)
DO 550 I = 1,NF
  TI = (0.D0 - I)/2.D0
  A1 = ((TI - X(3)) ** 2)/2.D0
  A2 = - X(2) * A1
  A3 = DEXP (A2)
  A4 = X(1) * A3
  F(I) = A4 - Y(I)

```

```

550  CONTINUE
     GOTO 8000

```

```

600  CONTINUE
C    FUNCAO MAL ESCALADA DE POWELL
  A1 = DEXP(-X(1))
  A2 = DEXP(-X(2))
  A3 = 1.0D+4
  F(1) = A3 * X(1) * X(2) - 1.D0
  F(2) = A1 + A2 - 1.0001D0
  GOTO 8000

```

```

700  CONTINUE
C    FUNCAO DA CAIXA TRIDIMENSIONAL
  DO 750 I = 1,NF
    TI = 1.D-1 * I
    A1 = DEXP (-TI * X(1))
    A2 = DEXP (-TI * X(2))
    A3 = DEXP (-TI) - DEXP (-10.D0 * TI)
    F(I) = A1 - A2 - X(3) * A3
  750  CONTINUE
     GOTO 8000

```

```

800  CONTINUE
C    FUNCAO VARIAVELMENTE DIMENSIONADA
  DO 820 I = 1,N
    F(I) = X(I) - 1.D0
    A1 = 0.D0
    DO 820 J = 1,N
      A1 = A1 + J * (X(J) - 1.D0)
  820  CONTINUE
  F(N+1) = A1
  F(N+2) = A1 ** 2

```

GOTO 8000

C-----

900 CONTINUE

C FUNCAO DE WATSON

DO 930 I = 1,29

TI = 1/29.D0

A1 = 0.D0

A2 = 0.D0

DO 920 J = 1,N

A3 = TI ** (J-2)

A1 = A1 + (J-1) * X(J) * A3

A2 = A2 + X(J) * A3 * TI

920 CONTINUE

F(I) = A1 - A2 ** 2 - 1.D0

930 CONTINUE

F(30) = X(1)

F(31) = X(2) - X(1) ** 2 - 1.D0

GOTO 8000

C-----

1000 CONTINUE

C FUNCAO PENALIDADE I

A1 = DSQRT(1.0D-5)

DO 1020 I = 1,N

F(I) = A1 * (X(I) - 1.D0)

1020 CONTINUE

A2 = 0.D0

DO 1050 J = 1,N

A2 = A2 + X(J) ** 2

1050 CONTINUE

F(N+1) = A2 - 0.25D0

GOTO 8000

C-----

1100 CONTINUE

C FUNCAO PENALIDADE II

A1 = DSQRT(1.0D-5)

F(1) = X(1) - 0.2D0

DO 1130 I = 2,N

A2 = DEXP (1.D-1 * X(I))

A3 = DEXP (1.D-1 * X(I-1))

YI = DEXP (1.D-1 * I) + DEXP (1.D-1 * (I-1))

F(I) = A1 * (A2 + A3 - YI)

1130 CONTINUE

A2 = DEXP (-1.D-1)

DO 1140 I = (N+1), (2*N-1)

A3 = DEXP (1.D-1 * X(I-N+1))

F(I) = A1 * (A3 - A2)

1140 CONTINUE

A2 = 0.D0

DO 1150 J = 1,N

A2 = A2 + (N-J+1) * (X(J) ** 2)

1150 CONTINUE

F(2*N) = A2 - 1.D0

GOTO 8000

C-----

1200 CONTINUE

```

C      FUNCAO HAL ESCALADA DE BROWN
      F(1) = X(1) - 1.0D+6
      F(2) = X(2) - 2.0D-6
      F(3) = X(1) * X(2) - 2.D0
      GOTO 8000

C-----
1300  CONTINUE
C      FUNCAO DE BROWN E DENNIS
      DO 1350 I = 1,NF
        TI = 1/5.D0
        A1 = DEXP (TI)
        A2 = DSIN (TI)
        A3 = DCOS (TI)
        A4 = X(1) + TI * X(2) - A1
        A5 = X(3) + X(4) * A2 - A3
        F(I) = A4 ** 2 + A5 ** 2
1350  CONTINUE
      GOTO 8000

C-----
1400  CONTINUE
C      GULF RESEARCH AND DEVELOPMENT FUNCTION
      IF ((X(1).NE.0.D0).AND.(X(2).NE.0.D0)) THEN
        DO I = 1,NF
          TI = 1.D-2 * I
          YI = (-50.D0 * DLOG(TI)) ** (2.D0/3.D0) + 25.D0
          A1 = DABS (YI * NF * I * X(2))
          A2 = A1 ** X(3)
          A3 = -A2/X(1)
          A4 = DEXP(A3)
          F(I) = A4 - TI
        END DO
      ELSE
        ERRO = .TRUE.
      ENDIF
      GOTO 8000

C-----
1500  CONTINUE
C      FUNCAO TRIGONOMETRICA TRANSLADADA DE (1,1,...,1)
      A1 = 0.D0
      DO 1520 J = 1,N
        Y(J) = DSIN(X(J) - 1.0D0)
        A1 = A1 + DCOS(X(J) - 1.0D0)
1520  CONTINUE
      DO 1550 I = 1,NF
        A2 = DCOS(X(I) - 1.0D0)
        F(I) = N - A1 + I * (1.D0 - A2) - Y(I)
1550  CONTINUE
      GOTO 8000

C-----
1600  CONTINUE
C      FUNCAO DE ROSENBROCK ESTENDIDA
      DO 1650 I = 1, (NF/2)
        K = 2 * I - 1
        F(K) = 10.D0 * (X(K+1) - X(K) ** 2)
        F(K+1) = 1.D0 - X(K)

```

```

1650    CONTINUE
      GOTO 8000
C-----
1700    CONTINUE
C      FUNCAO SINGULAR DE POWELL ESTENDIDA TRANSLADADA DE (1,1,...,1)
      A1 = DSQRT(5.0D+0)
      A2 = DSQRT(1.0D+1)
      DO 1750 I = 1, (N/4)
        K = 4 * I - 3
        F(K) = X(K) + 10.D0 * X(K+1) + 11.D0
        F(K+1) = A1 * (X(K+2) - X(K+3))
        F(K+2) = (X(K+1) - 2.D0 * X(K+2) - 1.D0) ** 2
        F(K+3) = A2 * (X(K) - X(K+3)) ** 2
1750    CONTINUE
      GOTO 8000
C-----
1800    CONTINUE
C      FUNCAO DE BEALE
      IF (X(2).EQ.0.D0) THEN
        ERRO = .TRUE.
      ELSE
        Y(1) = 1.5D0
        Y(2) = 2.25D0
        Y(3) = 2.625D0
        DO I = 1,NF
          A1 = X(2) ** I
          F(I) = Y(I) - X(1) * (1.D0 - A1)
        END DO
      ENDIF
      GOTO 8000
C-----
1900    CONTINUE
C      FUNCAO DE WOOD
      A1 = DSQRT(9.0D+1)
      A2 = DSQRT(1.0D+1)
      A3 = 1.D0/A2
      F(1) = 10.D0 * (X(2) - X(1) ** 2)
      F(2) = 1.D0 - X(1)
      F(3) = A1 * (X(4) - X(3) ** 2)
      F(4) = 1.D0 - X(3)
      F(5) = A2 * (X(2) + X(4) - 2.D0)
      F(6) = A3 * (X(2) - X(4))
      GOTO 8000
C-----
8000    CONTINUE
      IF (ERRO) THEN
        FUNC = 0.D0
      ELSE
        A1 = 0.D0
        DO I = 1, NF
          A1 = A1 + F(I) ** 2
        END DO
        FUNC = A1
      ENDIF
9000    RETURN
      END

```

REFERÊNCIAS

- [1] DENNIS, J.E. and SCHNABEL, R.B., "Numerical Methods for Unconstrained Optimization and Nonlinear Equations", New Jersey, Prentice-Hall, Englewood Cliffs, 1983.
- [2] FLETCHER, R., "Practical Methods of Optimization", 2 ed., New York, John Wiley & Sons, 1987.
- [3] GAY, D.M., "Computing Optimal Locally Constrained Steps", SIAM J. Sci. Stat. Comput., v.2, no.2, 186-197, 1981.
- [4] GILL, P.E., MURRAY, W. and WRIGHT, M.H., "Practical Optimization", London, Academic Press, 1981.
- [5] GOLDFELD, S.M., QUANDT, R.E. and TROTTER, H.F., "Maximization by Quadratic Hill-Climbing", Econometrica, 34, 541-551, 1966.
- [6] GOLUB, G.H. and VAN LOAN, C., "Matrix Computations", Baltimore, The Johns Hopkins University Press, 1983.
- [7] HEBDEN, M.D. , "An Algorithm for Minimization Using Exact Second Derivatives", Atomic Energy Research Establishment, Report T.P.515, Harwell, England, 1973.
- [8] LEVENBERG, K. "A Method for the Solution of Certain Nonlinear Problems in Least Squares", Quart. Appl. Math., 2, 164-168, 1944.
- [9] LUENBERGER, D.G., "Linear and Nonlinear Programming", 2 ed., California, Addison-Wesley Publishing Co., 1984.

- [10] MARQUARDT, D.W., "An Algorithm for Least Squares Estimation of Nonlinear Parameters", SIAM J. Appl. Math., 11, 431-441, 1963.
- [11] MARTINEZ, J.M., "Local Minimizers of a Quadratic Function with a Spherical Constraint", RT 16/89, IMECC-UNICAMP, Campinas, 1989.
- [12] MORE, J.J., "The Levenberg-Marquardt Algorithm: Implementation and Theory", Lecture Notes in Mathematics 630, G.A.Watson, ed., Springer-Verlag, Berlin - Heidelberg - New York, 105-116, 1978.
- [13] MORE, J.J., "Recent Developments in Algorithms and Software for Trust Region Methods", Argonne National Laboratory, Mathematics and Computer Science Division, Report ANL/MCS-TM-2, Argonne, Illinois, 1982.
- [14] MORE, J.J., "Trust Regions and Projected Gradients", Argonne National Laboratory, Mathematics and Computer Science Division, Report ANL/MCS-TM-107, Argonne, Illinois, 1988.
- [15] MORE, J.J., GARBOW, B.S. and HILLSTROM, K.E., "Testing Unconstrained Optimization Software", ACM Trans. Math. Soft., v.7, no.1, 17-41, 1981.
- [16] MORE, J.J. and SORENSEN, D.C., "Newton's Method", Argonne National Laboratory, Applied Mathematics Division, Report ANL-82-2, Argonne, Illinois, 1982.
- [17] MORE, J.J. and SORENSEN, D.C., "Computing a Trust Region Step", SIAM J. Numer. Anal., v.4, no.3, 553-572, 1983.

- [18] POWELL, M.J.D., "A Hybrid Method for Nonlinear Equations", in Numerical Methods for Nonlinear Algebraic Equations, P.Rabinowitz, ed., Gordon and Breach, London, 87-114, 1970.
- [19] POWELL, M.J.D., "Convergence Properties of a Class of Minimization Algorithms". In: MANGASARIAN, O.L. et alii, "Nonlinear Programming 2", New York, Academic Press, 1-27, 1975.
- [20] REINSCH, C.H., "Smoothing by spline functions II", Numer. Math. 16, 451-454, 1971.
- [21] SHULTZ, G.A., SCHNABEL, R.B. and BYRD, R.H., "A Family of Trust-Region-Based Algorithms for Unconstrained Minimization with Strong Global Convergence Properties", SIAM J. Numer. Anal., v.22, no.1, 47-67, 1985.
- [22] SORENSEN, D.C., "Newton's Method with a Model Trust Region Modification", SIAM J. Numer. Anal. v.19, no.2, 409-426, 1982.
- [23] THOMAS, S.W., "Sequential Estimation Techniques for Quasi Newton Algorithms", Ph. D. dissertation, Cornell University, Ithaca, New York, 1976.
- [24] TOINT, Ph.L., "Global Convergence of a Class of Trust Region Methods for Nonconvex Minimization in Hilbert Space", IMA J. Numer. Anal., 8, 231-252, 1988.