
Universidade Estadual de Campinas
Instituto de Matemática Estatística e Computação Científica
Departamento de Matemática Aplicada

Desempenho de algoritmos de região de confiança para problemas de empacotamento de cilindros

Larissa Oliveira Xavier*

Mestrado em Matemática Aplicada - Campinas - SP

Orientador: Profa. Dra. Sandra Augusta Santos

Co-Orientador: Prof. Dr. José Mario Martínez

* Este trabalho teve apoio financeiro da CAPES e da FAPESP.

Desempenho de algoritmos de região de confiança para problemas de empacotamento de cilindros

Este exemplar corresponde à redação final da dissertação devidamente corrigida e defendida por **Larissa Oliveira Xavier** e aprovada pela comissão julgadora.

Campinas, 20 de Abril de 2007



Profa. Dra. Sandra A. Santos

Orientadora



Prof. Dr. José Mario Martínez

Co-Orientador

Banca Examinadora

1. Profa. Dra. Elizabeth Wegner Karas (Depto. de Matematica - UFPR)
2. Profa. Dra. Márcia Aparecida Gomes Ruggiero (IMECC - Unicamp)
3. Profa. Dra. Sandra Augusta Santos (IMECC - Unicamp)

Dissertação apresentada ao Instituto de Matemática, Estatística e Computação Científica, Unicamp, para obtenção do Título de MESTRE em Matemática Aplicada.

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Bibliotecária: Miriam Cristina Alves – CRB8a / 5094

Xavier, Larissa Oliveira

X18d Desempenho de algoritmos de região de confiança para problemas de empacotamento de cilindros/Larissa Oliveira Xavier -- Campinas, [S.P. :s.n.], 2007.

Orientadores : Sandra Augusta Santos; José Mario Martínez

Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Algoritmos. 2. Programação (Matemática). 3. Otimização matemática. 4. Empacotamento de esferas. I. Santos, Sandra Augusta. II. Martínez, José Mario. III. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. IV. Título.

Título em inglês: Packing cylinders using trust-region algorithms: a comparative study

Palavras-chave em inglês (Keywords): 1. Algorithms. 2. Programming (Mathematics). 3. Optimization (Mathematics). 4. Packing spheres.

Área de concentração: Otimização

Titulação: Mestre em Matemática Aplicada

Banca examinadora: Profa. Dra. Sandra Augusta Santos (IMECC-Unicamp)
Profa. Dra. Elizabeth Wegner Karas (Depto. de Matematica - UFPR)
Profa. Dra. Márcia Aparecida Gomes Ruggiero (IMECC-Unicamp)

Data da defesa: 20/04/2007

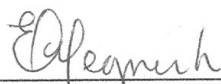
Programa de Pós-Graduação: Mestrado em Matemática Aplicada

Dissertação de Mestrado defendida em 20 de abril de 2007 e aprovada

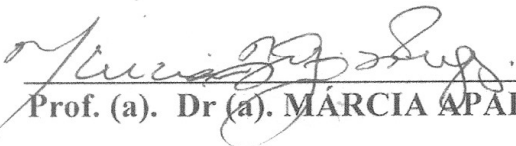
Pela Banca Examinadora composta pelos Profs. Drs.



Prof. (a). Dr (a). SANDRA AUGUSTA SANTOS



Prof. (a). Dr (a). ELIZABETH WEGNER KARAS



Prof. (a). Dr (a). MÁRCIA APARECIDA GOMES RUGGIERO

Agradecimentos

Agradeço:

Aos meus pais e a minha família por sempre me apoiarem em meus estudos e atividades.

À professora Sandra Santos, não apenas por sua orientação desde que eu comecei meus estudos na graduação, mas também pela amizade e confiança em meu trabalho.

Ao professor José Mario Martínez pela co-orientação e pelas importantes sugestões para o desenvolvimento deste projeto.

Aos professores do IMECC com os quais pude aprender a maior parte do que sei hoje e pelas sugestões e incentivos durante estes anos de estudo.

Aos meus colegas e amigos pelo apoio e torcida.

Aos funcionários da secretaria e biblioteca do IMECC que sempre me ajudaram quando precisei.

À FAPESP e à CAPES pelo apoio financeiro.

Resumo

Este trabalho encaminha a investigação de questões relacionadas ao desempenho de algoritmos de região de confiança para problemas de otimização irrestrita de grande porte. O algoritmo clássico de Moré e Sorensen, baseado em fatorações de Cholesky, é comparado com a abordagem de Rojas, Santos e Sorensen (algoritmo **RSS**). Do ponto de vista teórico são estudados os resultados de convergência dos dois algoritmos. Em termos práticos, são resolvidos problemas com a estrutura típica de empacotamento de cilindros. Também são pesquisados o desempenho efetivo do algoritmo **RSS** na solução aproximada dos subproblemas, e a repercussão da precisão com que os subproblemas são resolvidos no esforço global do algoritmo.

Palavras-chave: minimização irrestrita; regiões de confiança; soluções dos subproblemas; fatorações; grande porte; empacotamento de cilindros.

Abstract

This work investigates issues related to the performance of trust-region algorithms for large-scale unconstrained minimization. The classic algorithm of Moré and Sorensen, based on Cholesky's factorizations, is compared with the approach of Rojas, Santos and Sorensen (algorithm RSS). From the theoretic standpoint, the convergence results of both algorithms are compiled. In practical terms, problems with the typical structure of packing of cylinders are solved. The effective performance of the algorithm RSS in the approximate solution of the subproblems is analyzed as well, together with the influence of the inner precision of the subproblems to the global effort of the algorithm.

Keywords: unconstrained minimization; trust regions; solutions of the subproblems; factorizations; large scale; packing of cylinders.

Sumário

Introdução	1
1 Métodos de região de confiança para minimização irrestrita	5
1.1 Propriedades da solução dos subproblemas	7
2 Algoritmo de Moré e Sorensen	9
2.1 Estimando o parâmetro λ	9
2.2 Tratamento do <i>hard case</i>	13
2.3 Detalhamento do Algoritmo	14
2.3.1 Salvaguardas	14
2.3.2 Inicialização dos limitantes λ_L , λ_U e λ_S	15
2.3.3 Atualização dos limitantes	18
2.3.4 Critérios de parada	20
2.3.5 Algoritmo de Moré e Sorensen	21
2.4 Convergência global	22
3 Algoritmo de Rojas, Santos e Sorensen	23
3.1 Um problema de autovalores equivalente	23
3.2 Tratamento do hard case	25
3.3 Ajuste do parâmetro α	30
3.4 Salvaguardas e critérios de parada	32
3.4.1 Inicialização dos parâmetros	32
3.4.2 Atualização de α_L e α_U	33
3.4.3 Critérios de parada	33
3.5 Algoritmo de Rojas, Santos e Sorensen	34
3.6 Convergência do método na resolução dos subproblemas	36
4 Problema de empacotamento de cilindros	37
4.1 Armazenamento da matriz Hessiana e programação	40
4.2 Decomposição de Cholesky para uma matriz esparsa	43
5 Experimentos computacionais	46
5.1 Testes iniciais	47
5.2 Testes da literatura	52
Considerações finais	57

Introdução

Os métodos de região de confiança nasceram de abordagens para o problema de estimativa de parâmetros não-lineares, com os trabalhos de Levenberg [22] e Marquardt [24]. Ambos sugeriram a adição de um múltiplo da identidade à matriz Hessiana como uma forma de estabilização, no contexto da solução de problemas de quadrados mínimos não lineares. Marquardt provou que a minimização do modelo estabilizado corresponde a minimizar o modelo original em uma região restrita. Estas duas abordagens são empregadas na regularização de problemas mal-postos e na suavização de dados experimentais com ruídos (ver [4, cap. 5] e [15, 18, 33, 44]).

Até a década de setenta, utilizavam-se as expressões “método de passo restrito”, “região de validade”, ou ainda, “região de certeza” do modelo. Aparentemente, a terminologia “região de confiança” foi introduzida por Dennis em um curso que ministrou no U. S. National Bureau of Economics Research, inspirado em uma palestra de Powell, que empregou a técnica para assegurar convergência em um método para otimização irrestrita [31], bem como na resolução de sistemas não lineares [32]. O *survey* de Moré [28] certamente contribuiu para sistematizar a nomenclatura nesta área. Recentemente, o livro de Conn, Gould e Toint [7] compila o estado da arte e oferece um excelente panorama do tema nos diversos tópicos da otimização.

Os métodos de região de confiança para minimização irrestrita desfrutam de propriedades fortes de convergência e de uma boa reputação na prática. O principal esforço computacional destes métodos reside na determinação de um minimizador aproximado para um *modelo* da função objetivo restrito a uma *região de confiança*, que consiste na imposição de um tamanho máximo (*raio de confiança*) para o passo, em uma norma conveniente.

No processo iterativo de resolução do problema, o raio é ajustado de tal forma que os sucessivos modelos reflitam o comportamento da função original na região de confiança. Os modelos quadráticos são os mais utilizados, tanto pela simplicidade quanto por proporcionarem convergência rápida ao método. Do ponto de vista teórico, a escolha da norma que define a região é irrelevante, desde que seja uniformemente relacionada com a norma euclidiana. Tanto a norma euclidiana quanto a norma infinito são escolhas populares,

feitas de acordo com a geometria do problema. Para problemas de grande porte, por exemplo, a escolha de uma norma euclidiana ponderada pode ser essencial para um bom desempenho (cf. [19], ver também [10]).

O subproblema de região de confiança, minimizar uma função quadrática com uma restrição elipsoidal, possui uma estrutura privilegiada e tem sido intensamente explorado sob diversos pontos de vista, conforme revela a literatura na área.

Para o caso denso, no qual a matriz Hessiana utilizada para definir a função quadrática do subproblema de região de confiança não apresenta uma estrutura de esparsidade, o consagrado algoritmo de Moré-Sorensen [29], que se apóia em fatorações de Cholesky, produz uma solução quase-ótima em um número finito de iterações. Este é o método implementado no **NEOS-server** [2, 12] para os subproblemas do algoritmo **NMTR** (*Newton Method Trust Region*).

Nas últimas duas décadas foram propostos vários algoritmos livre de fatorações (ou *matrix free*) para o subproblema de regiões de confiança em problemas de grande porte. Nestes algoritmos, devido ao custo, ou impossibilidade de se construir ou acessar os elementos da matriz Hessiana a informação de segunda ordem do problema é recuperada por meio do produto desta matriz por um vetor.

Steihaug [43] e Toint [45] propuseram, independentemente, que o sistema linear presente nos métodos anteriormente mencionados fossem resolvidos iterativamente via gradientes conjugados, terminando quando o passo deixasse a região de confiança, ou quando uma direção de curvatura negativa fosse encontrada. Buscando melhorar o desempenho do método de Steihaug-Toint para o caso indefinido, Gould, Lucidi, Roma e Toint [19] investigaram a continuação do esquema iterativo via método de Lanczos quando uma direção de curvatura negativa fosse determinada (algoritmo **GLTR**, *Generalized Lanczos Trust Region*).

Outros trabalhos desenvolvidos para o caso esparso foram os de Sorensen [40], e Rojas, Santos e Sorensen [37], nos quais o subproblema de região de confiança é tratado como um problema de autovalores parametrizado. A implementação do algoritmo proposto em [37], e denominado **LSTRS** (*Large Scale Trust Region Subproblem*) está descrita em [38]. A mesma filosofia de autovalores parametrizados permeia o trabalho de Rendl e Wolkowicz [35], que utiliza a linguagem da programação semidefinida e propõe um algoritmo primal-dual para o subproblema de região de confiança. Este trabalho foi revisto e ampliado por Fortin e Wolkowicz [14]. Lucidi, Palagi e Roma [23] identificam propriedades teóricas do subproblema de região de confiança, que é reformulado como um problema irrestrito com função objetivo quártica por partes, e para o qual estratégias que permitem sair de pontos estacionários distintos da solução desejada são delineadas. Um algoritmo baseado na diferença de funções convexas é proposto por Pham Dinh e Hoai An [30], que encontra pontos estacionários para o subproblema de região de confiança, e eventualmente necessita ser

recomeçado. Recentemente, Hager [20] propôs um método de minimização em subespaços sequenciais (**SSM**, do inglês *Sequential Subspace Method*), no qual considera subproblemas restritos aos subespaços de Krylov (precondicionados) gerados por um passo do algoritmo de programação quadrática sequencial aplicado ao subproblema de região de confiança.

Também foram desenvolvidos diversos algoritmos de região de confiança para problemas gerais de otimização. O livro [7], com sua bibliografia comentada, oferece uma extensa cobertura neste sentido, incluindo contribuições do grupo MCDO (Métodos Computacionais de Otimização), do IMECC, Unicamp [11, 13, 26, 27].

As ferramentas de derivação simbólica (*softwares Mathematica, Matlab, Mupad*, entre outros), e de derivação automática (por exemplo **ADIFOR** [1]) certamente favoreceram as investidas no cálculo de vetores gradiente e montagem efetiva de matrizes Hessianas no contexto da otimização computacional. Com um desempenho médio inferior a duas iterações por subproblema (cf. [29]), o algoritmo de Moré-Sorensen é clássico para resolver aproximadamente os subproblemas em algoritmos de região de confiança para minimização irrestrita. O uso de fatorações de Cholesky é a principal crítica para não se adotar este algoritmo na solução de problemas de grande porte. Nas décadas de 80 e 90, esta talvez fosse uma real fraqueza desta abordagem. Com o advento de máquinas mais velozes e com crescente disponibilidade de memória, nos perguntamos se os sofisticados algoritmos livres de fatoração são de fato a melhor opção para problemas de grande porte. Não haveria classes de problemas com muitas variáveis, mas para os quais fatorações de Cholesky (possivelmente esparsas e com salvaguardas, ver p.ex. [17]) são uma alternativa competitiva?

Mais concretamente, neste trabalho investigamos como se comparam os algoritmos de Moré-Sorensen e **LSTRS** para problemas com estruturas típicas. Consideramos famílias de problemas de empacotamento de cilindros [5, 8] como ponto de partida para os desafiantes problemas de configuração molecular [3, 25].

Um outro aspecto que pesquisamos é o desempenho efetivo do algoritmo **RSS** na solução aproximada dos subproblemas em uma implementação via regiões de confiança para minimização irrestrita. Embora este algoritmo tenha mostrado suas potencialidades em diversos casos, especialmente na regularização de problemas mal postos [36, 39], ainda não foi testado como *solver* interno para minimização sem restrições. Analisamos se o algoritmo **RSS** permite, da mesma maneira que o algoritmo de Moré-Sorensen, a obtenção de soluções de baixo custo e úteis no contexto dos algoritmos de região de confiança. A esta questão da sensibilidade está naturalmente relacionada a precisão demandada na resolução dos subproblemas e sua repercussão no esforço global do algoritmo.

Este texto está organizado da seguinte maneira. No Capítulo 1 apresentamos brevemente os métodos de região de confiança para minimização irrestrita, com ênfase nas propriedades teóricas dos subproblemas que servem de subsídio para os algoritmos de-

envolvidos para abordá-los. Os Capítulos 2 e 3 detalham os algoritmos escolhidos para encontrar soluções aproximadas dos subproblemas neste trabalho, respectivamente o de Moré e Sorensen e o de Rojas, Santos e Sorensen. O Capítulo 4 descreve o problema de empacotamento de cilindros, detalhando os aspectos computacionais relativos ao armazenamento da matriz Hessiana e à implementação da decomposição de Cholesky para o armazenamento proposto. Os experimentos numéricos são descritos e analisados no Capítulo 5. O texto é concluído com algumas considerações finais e perspectivas de trabalhos futuros.

Capítulo 1

Métodos de região de confiança para minimização irrestrita

O problema de otimização abordado neste trabalho é a minimização irrestrita, isto é,

$$\begin{array}{ll} \min & f(x) \\ \text{s.a} & x \in \mathbb{R}^n \end{array} \quad (1.1)$$

onde a função objetivo $f : \mathbb{R}^n \rightarrow \mathbb{R}$ é suave.

A resolução de (1.1) é encaminhada por métodos de região de confiança, que podem ser vistos essencialmente como uma estratégia de globalização para o método de Newton.

A idéia principal dos métodos de região de confiança é aproximar a cada iteração a função objetivo original por uma quadrática construída em torno do iterando x_k da k -ésima iteração do algoritmo. Equivalentemente, pode-se trabalhar com a quadrática que modela o decréscimo da função objetivo.

A cada iteração é resolvido então um subproblema de minimizar a quadrática em questão sujeita a uma região de confiança $\|x - x_k\| \leq \Delta$ com Δ o raio da região de confiança. A escolha de Δ é importante para garantir que o minimizador da quadrática aproxima bem o minimizador da função objetivo restrita à região de confiança.

Na Figura 1 temos um exemplo ilustrativo com as curvas de nível da função objetivo e as curvas de nível da quadrática no primeiro e segundo gráficos, respectivamente. O ponto preto é o iterando x_k e o ponto vermelho é o minimizador da quadrática sujeita à região de confiança.

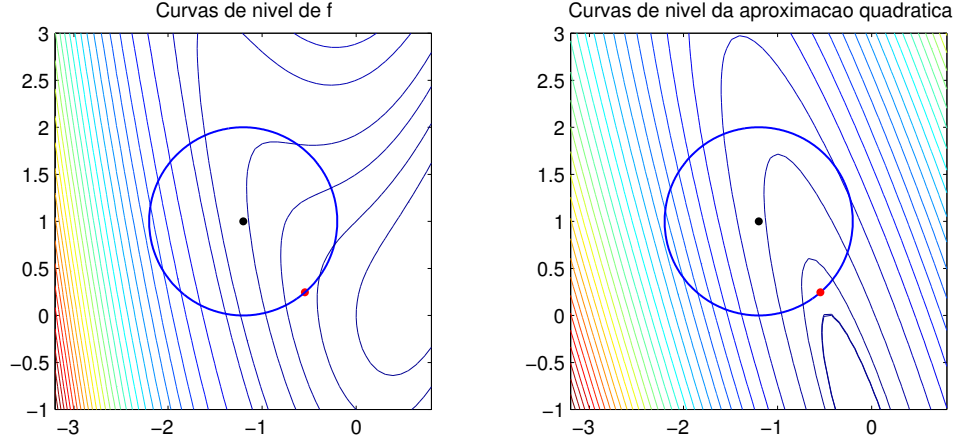


Figura 1 - Ilustração de uma iteração de um método de região de confiança.

Assim, a cada iteração deve-se resolver, aproximadamente, um subproblema que consiste em minimizar uma aproximação quadrática para o decréscimo da função objetivo:

$$\psi(s) = \frac{1}{2}s^T H_k s + g_k^T s$$

restrita à região de confiança

$$\|s\|_2 \leq \Delta_k,$$

onde x_k é o iterando da k -ésima iteração, $g_k = \nabla f(x_k)$ é o vetor gradiente, $H_k = \nabla^2 f(x_k)$ é a matriz Hessiana e Δ_k é o raio da região de confiança.

O algoritmo a seguir sistematiza os passos que produzem uma sequência de aproximações $\{x_k\}$ para a solução do problema (1.1) via regiões de confiança.

Algoritmo de região de confiança para minimização irrestrita (RCMI)

- Sejam $0 < \mu < \eta < 1$ e $0 < c_1 < c_2 < 1 < c_3$ constantes dadas.
- Escolher $x_0 \in \mathbb{R}^n$ e o raio inicial (Δ_0) e tomar $k = 0$.
- Enquanto o critério de parada não for satisfeito:
 1. Calcular $g_k = \nabla f(x_k)$ e $H_k = \nabla^2 f(x_k)$
 2. Obter uma solução aproximada s_k para o subproblema

$$\begin{aligned} \min \quad & \psi_k(s) = g_k^T s + \frac{1}{2}s^T H_k s \\ \text{s.a.} \quad & \|s\|_2 \leq \Delta_k, \end{aligned}$$

3. Computar $\rho_k = \frac{f(x_k + s_k) - f(x_k)}{\psi_k(s_k)}$.
4. Se $\rho_k \leq \mu$ então $\Delta_k \leftarrow \Delta \in [c_1\Delta_k, c_2\Delta_k]$ e vá para 2.
5. $x_{k+1} = x_k + s_k$.
6. Se $\rho_k \leq \eta$ então $\Delta_{k+1} \in [c_2\Delta_k, \Delta_k]$, caso contrário $\Delta_{k+1} \in [\Delta_k, c_3\Delta_k]$.
7. Tomar $k = k + 1$.

Este trabalho focaliza especialmente o passo 2 do algoritmo RCMI. Escolhemos dois algoritmos distintos para obter as aproximações para as soluções dos subproblemas, cujos desempenhos são comparados no contexto dos problemas de empacotamento de cilindros.

Antes de detalharmos cada um destes algoritmos, apresentaremos as propriedades intrínsecas dos subproblemas (função quadrática restrita a uma região elipsoidal), que serão convenientemente exploradas em cada um dos algoritmos.

1.1 Propriedades da solução dos subproblemas

Em algoritmos para métodos de região de confiança é necessária, a cada iteração, a resolução de subproblemas da forma

$$\begin{aligned} \min \quad & \psi(s) = g^T s + \frac{1}{2} s^T H s \\ \text{s. a} \quad & \|s\|_2 \leq \Delta \end{aligned} \tag{1.2}$$

onde $g \in \mathbb{R}^n$, $H \in \mathbb{R}^{n \times n}$ simétrica, $\Delta > 0$. Para simplificar a notação denotaremos ao longo deste texto a norma euclidiana por $\|\cdot\|_2 = \|\cdot\|$, e suprimiremos o subíndice da k -ésima iteração nos dados g_k , H_k , Δ_k dos subproblemas.

Uma importante classe de métodos para resolver o subproblema (1.2) é caracterizada pela resolução do sistema

$$(H - \lambda I)s = -g \tag{1.3}$$

com o multiplicador de Lagrange $\lambda \leq 0$ a ser determinado. Para uniformizar a apresentação vamos usar a notação de [37] para o sinal do multiplicador λ . Essa escolha gerou diversas modificações e adaptações nas expressões do algoritmo de Moré e Sorensen.

Os métodos apresentados nas próximas seções são fortemente baseados na busca pelo parâmetro λ que aparece no sistema (1.3) e são fundamentados pelo lema a seguir, cuja prova pode ser encontrada em [41]. Esse lema caracteriza a solução do problema (1.2), não havendo ‘gap’ entre as condições necessárias e suficientes de otimalidade, independentemente de qualquer hipótese de convexidade.

Lema 1 Um vetor viável s^* é solução de (1.2) com correspondente multiplicador de Lagrange λ^* se e somente se s^* e λ^* satisfazem $(H - \lambda^*I)s^* = -g$ com $H - \lambda^*I$ semidefinida positiva, $\lambda^* \leq 0$ e $\lambda^*(\Delta - \|s^*\|) = 0$.

Em geral, a matriz H pode não ser semidefinida positiva, tendo algum autovalor negativo. Para estes casos o problema (1.2) apresenta solução na fronteira da região de confiança. A única possibilidade do subproblema (1.2) não apresentar solução na fronteira da região de confiança é quando H é uma matriz positiva definida e a solução é dada por $s^* = -H^{-1}g$ com $\|s^*\| < \Delta$.

Muitos algoritmos são baseados no fato de que se (1.2) tem solução na fronteira do conjunto $\{s : \|s\| \leq \Delta\}$ então, na maioria dos casos, uma solução pode ser encontrada determinando $\lambda^* \leq 0$ tal que $H - \lambda^*I$ é definida positiva e $\|(H - \lambda^*I)^{-1}g\| = \Delta$. Porém em um caso, denominado *hard case*, a equação $\|(H - \lambda^*I)^{-1}g\| = \Delta$ não tem solução com $H - \lambda^*I$ definida positiva.

Para encontrar uma aproximação para o parâmetro λ^* e tratar o problema também na ocorrência do *hard case*, apresentaremos nos próximos capítulos dois algoritmos. O primeiro, proposto por Moré e Sorensen, trata o problema utilizando a fatoração de Cholesky de uma matriz da forma $H - \lambda I$ e resolve, através do método de Newton, um subproblema de encontrar o zero de uma função de uma variável. O segundo algoritmo, proposto por Rojas, Santos e Sorensen, trata o problema obtendo os menores autovalores e autovetores correspondentes de uma matriz ampliada através de um método tipo Lanczos. A principal diferença entre estes algoritmos é a necessidade de fatorar a matriz H no algoritmo de Moré e Sorensen, enquanto que no algoritmo de Rojas, Santos e Sorensen é necessário somente efetuar produtos da matriz por um vetor.

Capítulo 2

Algoritmo de Moré e Sorensen

O algoritmo de Moré e Sorensen [29] foi proposto em 1983 para a resolução dos subproblemas (1.2) no contexto do algoritmo de Newton com região de confiança para minimização irrestrita. De uma maneira geral, este algoritmo fornece a cada iteração uma aproximação λ para o parâmetro λ^* buscando obter uma matriz $H - \lambda I$ positiva definida e então obtém s , solução do sistema $(H - \lambda I)s = -g$, pedindo que $\|s\| = \Delta$ durante o processo de atualizar λ .

A vantagem deste tipo de algoritmo é que, a cada iteração, o problema de encontrar $s^* \in \mathbb{R}^n$ é transformado em um problema escalar de estimar $\lambda^* \in \mathbb{R}$.

Neste método isto é feito através de uma fatoração de Cholesky de uma matriz de dimensão $n \times n$ a cada iteração e da aplicação do método de Newton para obter o zero de uma função secular¹ de uma variável. A função secular é caracterizada pela presença de pólos. No nosso caso estes pólos serão dados pelos autovalores de uma matriz conhecida e se o pólo associado ao menor autovalor não está definido, teremos uma situação que chamaremos de *hard case*. Quando o problema apresenta *hard case*, este método obtém uma aproximação para uma solução na fronteira da região de confiança durante o processo e ativa um dos critérios de parada do algoritmo, como pode ser visto em maiores detalhes nas seções seguintes.

2.1 Estimando o parâmetro λ

Um primeiro resultado importante para a dedução do método é que o problema (1.2) não tem solução na fronteira da região de confiança se e somente se H é positiva definida e

¹A terminologia 'função secular' é empregada na literatura para designar a função característica de um operador linear. Esta nomenclatura é uma herança do cálculo de órbitas planetárias, para distinguir movimentos lentos (seculares) de movimentos rápidos (anuais) (cf. <http://en.wikipedia.org/wiki/secular-function>)

a solução é dada por $s^* = -H^{-1}g$ com $\|s^*\| < \Delta$. Neste caso temos a solução s^* com multiplicador de Lagrange associado $\lambda^* = 0$ no Lema 1.

Nos demais casos o problema apresentará solução na fronteira e então o algoritmo busca s tal que $\|s\| = \Delta$. Com isso teremos uma relação importante.

Notando que H é uma matriz simétrica, temos sua decomposição espectral:

$$H = QDQ^T$$

onde $D = \text{diag}(\delta_1, \dots, \delta_n)$ com $\delta_1 \leq \dots \leq \delta_n$ os autovalores da matriz H e $Q^T Q = I$ com as colunas de Q formadas por um conjunto de autovetores ortonormais associados aos autovalores de H .

Então, se $\lambda < \delta_1$ é uma estimativa para o parâmetro λ^* e $H - \lambda I$ é definida positiva, pelo Lema 1, s_λ tal que $s_\lambda = -(H - \lambda I)^{-1}g$ é uma aproximação para a solução.

Fazendo $\|s_\lambda\|^2 = \|(H - \lambda I)^{-1}g\|^2$ temos que:

$$\|s_\lambda\|^2 = \sum_{j=1}^n \frac{\gamma_j^2}{(\delta_j - \lambda)^2}$$

com $\gamma = Q^T g = (\gamma_1, \dots, \gamma_n)^T$.

Uma característica importante desta função é a presença de pólos em $\lambda = \delta_j$, $j = 1, \dots, n$ se $\gamma_j \neq 0$.

Quando $\lambda \rightarrow \delta_1$, $\|s_\lambda\| \rightarrow \infty$ e a função $\Delta - \|s_\lambda\|$ é decrescente no intervalo $(-\infty, \delta_1)$. Assim no caso de $\gamma_1 \neq 0$ temos sempre solução para $\|s_\lambda\| = \Delta$, como pode ser visto no exemplo da Figura 2 e formalizado no lema a seguir (cf. [29]).

Lema 2 Se g não é ortogonal a $S_1 = \{z | Hz = \delta_1 z, z \neq 0\}$ com δ_1 o menor autovalor de H , então $\|s_\lambda\| = \Delta$ tem solução $\lambda \leq 0$ em $(-\infty, \delta_1)$ onde $s_\lambda = -(H - \lambda I)^{-1}g$.

Uma implicação do Lema 2 é que, se $\gamma_1 \neq 0$, temos uma solução $\lambda < \delta_1$ tal que $\|s_\lambda\| = \Delta$ com $H - \lambda I$ definida positiva, já que seus autovalores são dados por $\delta_j - \lambda \geq \delta_1 - \lambda > 0$, $j = 1, \dots, n$. Então pelo Lema 1 teremos que s_λ é solução do problema (1.2).

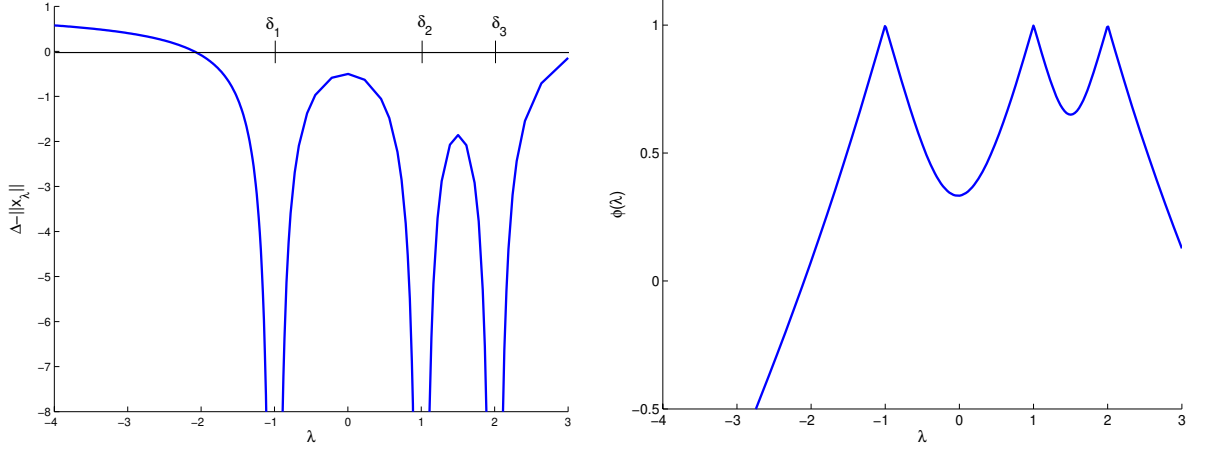


Figura 2 - Caso em que $\gamma_1 \neq 0$ e temos solução para $\Delta - \|s_\lambda\| = 0$ com $\lambda \in (-\infty, \delta_1)$.

Uma observação importante é que se H é definida positiva, temos que $\delta_1 > 0$ e poderíamos pensar que λ também pudesse ser positivo com $0 < \lambda < \delta_1$.

Para mostrar que isso não pode ocorrer, suponhamos que $\|s^*\| = \Delta$ e H é uma matriz definida positiva. Tomamos Δ_0 tal que $\Delta_0 = \|H^{-1}g\|$, ou seja, Δ_0 é igual à norma do minimizador da quadrática com Hessiana H para o problema irrestrito. Se mudássemos o raio da região de confiança de Δ para Δ_0 , estaríamos aumentando o raio ($\Delta_0 > \Delta$) a fim de englobar o minimizador do problema irrestrito. Neste caso teríamos a solução $s_\lambda = -(H - \lambda I)^{-1}g$ com $\lambda = 0$ e $\|s_\lambda\| = \Delta_0$, ou seja, $\lambda = 0$ é uma solução para $\Delta_0 - \|s_\lambda\| = 0$.

Agora, quando analisamos a função de λ original $\Delta - \|s_\lambda\|$ notamos que é a mesma, apenas deslocada da constante $-(\Delta_0 - \Delta)$. Isso faz com que a raiz da função também se desloque e então $\lambda < 0$, como pode ser visto no exemplo ilustrativo da Figura 3.

Assim, o algoritmo de Moré e Sorensen trabalha, de forma geral, estimando λ e calculando $\|s_\lambda\| = \|(H - \lambda I)^{-1}g\|$. Uma nova estimativa para λ é dada encontrando um zero da função $\|s_\lambda\| = \Delta$ no intervalo $(-\infty, \delta_1)$.

Para resolver $\|s_\lambda\| = \Delta$ podemos aplicar o método de Newton para encontrar o zero da função

$$\phi(\lambda) = \frac{1}{\Delta} - \frac{1}{\|s_\lambda\|}$$

com $(H - \lambda I)s_\lambda = -g$, como sugerido por Reinsch [34] e Hebden [21].

Como ilustrado na Figura 2, a função $\phi(\lambda)$ é mais suave do que $\Delta - \|s_\lambda\|$ e quase linear no intervalo de interesse, facilitando a busca do zero pelo método de Newton.

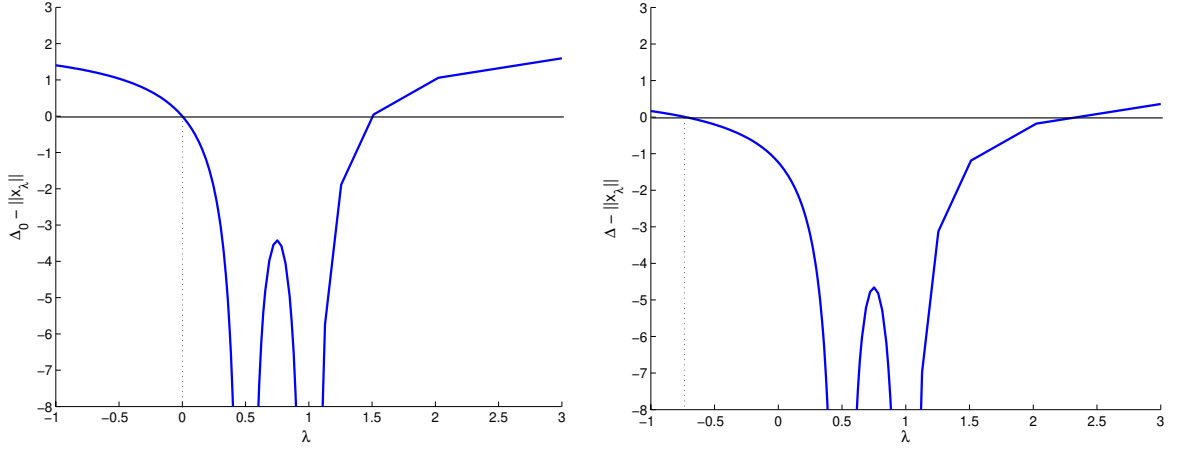


Figura 3 - Ilustração do fato que $\lambda \leq 0$ mesmo que $\delta_1 > 0$ (H definida positiva).

Calculando a derivada da função $\phi(\lambda)$ temos que

$$\phi'(\lambda) = \frac{s_\lambda^T (H - \lambda I) s_\lambda}{\|s_\lambda\|^3}.$$

Ao aplicar o método de Newton teremos então uma nova estimativa para λ da seguinte forma:

$$\lambda \leftarrow \lambda - \frac{\phi(\lambda)}{\phi'(\lambda)} = \lambda - \left(\frac{\|s_\lambda\| - \Delta}{\Delta} \right) \frac{\|s_\lambda\|^2}{s_\lambda^T (H - \lambda I)^{-1} s_\lambda}.$$

Como, em geral, excluindo o *hard case* que está detalhado na próxima seção, temos $H - \lambda I$ definida positiva, podemos utilizar a fatoração de Cholesky e então

$$H - \lambda I = R^T R.$$

Dessa forma temos que $s_\lambda^T (H - \lambda I)^{-1} s_\lambda = \|R^{-T} s_\lambda\|^2$ e então o passo geral do método de Newton é dado pelo seguinte algoritmo:

Um passo do método de Newton para encontrar o menor zero da função secular

Dado $\lambda \leq 0$ com $H - \lambda I$ definida positiva e $\Delta > 0$

1. Obter a fatoração $H - \lambda I = R^T R$
2. Resolver $R^T R s = -g$
3. Resolver $R^T q = s$

$$4. \lambda = \lambda - \left(\frac{\|s\|}{\|q\|} \right)^2 \left(\frac{\|s\| - \Delta}{\Delta} \right)$$

2.2 Tratamento do *hard case*

A garantia de existência de $\lambda \in (-\infty, \delta_1)$ tal que $\|s_\lambda\| = \|(H - \lambda I)^{-1}g\| = \Delta$ no Lema 2 foi possível devido a $\gamma_1 \neq 0$. Neste caso temos a existência de um pólo em $\lambda = \delta_1$.

No caso em que g é ortogonal a S_1 , ou seja, $\gamma_1 = 0$ podemos não ter solução para $\|s_\lambda\| = \Delta$ com $\lambda < \delta_1$, como é visto no segundo exemplo na Figura 4. Em outras palavras, não conseguimos obter λ tal que $\|s_\lambda\| = \Delta$ e $H - \lambda I$ seja definida positiva. Com esta situação temos caracterizado o *hard case*.

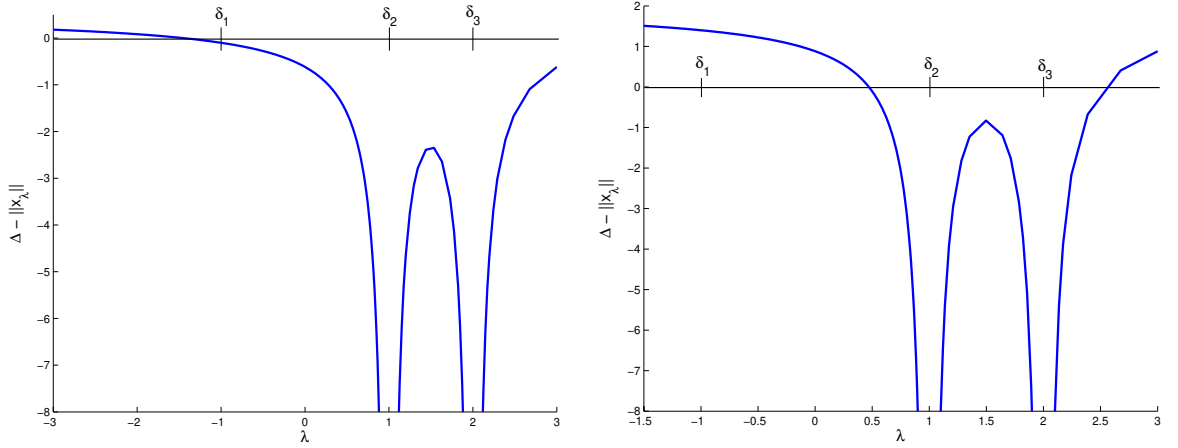


Figura 4 - Casos em que $\gamma_1 = 0$. À esquerda temos solução para $\Delta - \|s_\lambda\| = 0$ com $\lambda \in (-\infty, \delta_1)$ e à direita temos um exemplo do *hard case*.

O resultado a seguir, proposto em [29], fornece uma maneira para abordar o *hard case* no contexto de fatorações para $H - \lambda I$.

Lema 3 Dado $0 < \sigma < 1$, suponha que $H - \lambda I = R^T R$, $(H - \lambda I)s = -g$ com $\lambda \leq 0$.

Seja $z \in \mathbb{R}^n$ satisfazendo

$$\|s + z\| = \Delta, \text{ com } \|Rz\|^2 \leq \sigma(\|Rs\|^2 - \lambda\Delta^2). \quad (2.1)$$

Então

$$-\psi(s + z) \geq \frac{1}{2}(1 - \sigma)(\|Rs\|^2 - \lambda\Delta^2) \geq (1 - \sigma)|\psi^*|$$

onde ψ^* é o valor ótimo de (1.2).

Uma consequência importante do Lema 3 é que, se z satisfaz (2.1), então

$$|\psi(s+z) - \psi^*| \leq \sigma |\psi^*|,$$

ou seja, $s+z$ é uma boa aproximação para a solução do problema (1.2).

Com o *hard case*, se temos $H - \lambda I$ semidefinida positiva com $\lambda \leq 0$, então a solução s de $(H - \lambda I)s = -g$ é tal que $\|s\| < \Delta$.

Para satisfazer (2.1), tomamos $z = \tau \hat{z}$, escolhendo τ onde $\|s + \tau \hat{z}\| = \Delta$ e $\hat{z}$ com $\|\hat{z}\| = 1$ onde $\|R\hat{z}\|$ é o menor possível.

Fazendo $z = \tau \hat{z}$ e $\|s + z\|^2 = \Delta^2$ temos duas escolhas para τ .

Como $-\psi(s + \tau \hat{z}) = -\frac{1}{2}(\|Rs\|^2 + \lambda \Delta^2) + \frac{1}{2}\tau^2 \|R\hat{z}\|^2$, para minimizar $\psi(s + \tau \hat{z})$, escolhemos τ com menor magnitude. Assim

$$\tau = \frac{\Delta^2 - \|s\|^2}{s^T \hat{z} + \text{sgn}(s^T \hat{z})[(s^T \hat{z})^2 + (\Delta^2 - \|s\|^2)]^{\frac{1}{2}}} \quad (2.2)$$

Para a escolha de $\hat{z}$ devemos ter $\|\hat{z}\| = 1$ e $\|R\hat{z}\|$ o menor possível.

Levando em consideração o seguinte fato

$$\min_{\|s\|_2=1} \|Rs\|_2 = \sigma_{\min}$$

onde $\sigma_{\min}$ é o menor valor singular da matriz R , podemos obter $\hat{z}$ de maneira que $\|R\hat{z}\| \approx \sigma_{\min}$.

Em Cline, Moler, Stewart e Wilkinson [6] é sugerida uma estimativa para $\hat{z}$ buscando o autovetor associado ao menor autovalor de R através de uma iteração do método da potência inverso. A sugestão é tomar e , um vetor com componentes iguais a ± 1 , de forma que a solução w do sistema $R^T w = e$ tenha norma grande, e escolher $\hat{z} = \frac{v}{\|v\|}$ tal que v seja solução de $Rv = w$.

2.3 Detalhamento do Algoritmo

2.3.1 Salvaguardas

Como $\phi(\lambda)$ é uma função convexa e estritamente crescente no intervalo $(-\infty, \delta_1)$, temos que utilizar algumas salvaguardas para que o método de Newton encontre a raiz desejada de $\phi(\lambda)$.

Se o método de Newton é iniciado com $\lambda \in (-\infty, \delta_1)$ e $\phi(\lambda) > 0$, então, ao buscar λ^* tal que $\phi(\lambda^*) = 0$, o próximo iterando será tal que $\delta_1 < \lambda$, ou seja, o método irá produzir

uma sequência $\{\lambda_k\}$ no intervalo $(-\infty, \delta_1)$.

Por outro lado, o método de Newton pode ser iniciado com $\lambda \in (-\infty, \delta_1)$ tal que $\phi(\lambda) < 0$. Neste caso o próximo iterando deve ser tal que $\phi(\lambda_+) > \phi(\lambda)$ e como a função é crescente no intervalo $(-\infty, \delta_1)$, teremos que $\lambda_+ > \lambda$. Isso pode resultar em $\lambda_+ > \delta_1$.

Para assegurar que λ_+ fique próximo do intervalo $(-\infty, \delta_1)$ devemos utilizar um esquema de salvaguarda.

Para este esquema utilizamos os parâmetros λ_S , λ_U e λ_L , onde $[\lambda_L, \lambda_U]$ é um intervalo de incerteza que contém o λ desejado e λ_S é um limitante superior para δ_1 .

Com isso temos:

1. Se $\lambda < \lambda_S$, então

- $\lambda = \max\{\lambda, \lambda_L\}$
- $\lambda = \min\{\lambda, \lambda_U\}$

2. Se $\lambda \geq \lambda_S$ então $\lambda = \min\{0.001\lambda_L, -(\lambda_U\lambda_L)^{\frac{1}{2}}\}$

2.3.2 Inicialização dos limitantes λ_L , λ_U e λ_S

Para a inicialização de λ_S , λ_L e λ_U utilizamos a sugestão de [16] com as devidas adaptações para contemplar o sinal não positivo do multiplicador solução.

Pelo quociente de Rayleigh temos que $\delta_1 \leq \frac{u^T H u}{u^T u} \leq \delta_n$, $\forall u \in \mathbb{R}^n$. Em particular temos que $\delta_1 \leq \frac{e_i^T H e_i}{e_i^T e_i}$ com e_i o i -ésimo vetor canônico em $\mathbb{R}^n$. Assim $\delta_1 \leq h_{ii} \Rightarrow \delta_1 \leq \min\{h_{ii}\}$. Logo $\lambda_S = \min\{h_{ii}\}$ é uma cota superior para δ_1 .

Para a inicialização de λ_L e λ_U observamos que λ^* no Lema 1 é tal que $\Delta = \frac{\|g\|}{\bar{\lambda} - \lambda^*}$ para algum $\bar{\lambda}$ entre δ_1 e δ_n .

Para mostrar isso vamos separar a análise em dois casos:

- i. $\|s^*\| = \Delta$ e $\lambda^* < \delta_1$
- ii. $\|s^*\| < \Delta$ e $\lambda^* = 0$

Para o caso (i) temos que $\Delta^2 = \|s^*\|^2$. Com isso,

$$\Delta^2 = \sum_{j=1}^n \frac{\gamma_j^2}{(\delta_j - \lambda^*)^2}$$

com $H = QDQ^T$ e $\gamma = Q^T g$. Logo

$$\begin{aligned}\Delta^2 &\leq \frac{\sum_{j=1}^n \gamma_j^2}{(\delta_1 - \lambda^*)^2} = \frac{\|Q^T g\|^2}{(\delta_1 - \lambda^*)^2} = \frac{\|g\|^2}{(\delta_1 - \lambda^*)^2} \quad \text{e} \\ \Delta^2 &\geq \frac{\sum_{j=1}^n \gamma_j^2}{(\delta_n - \lambda^*)^2} = \frac{\|Q^T g\|^2}{(\delta_n - \lambda^*)^2} = \frac{\|g\|^2}{(\delta_n - \lambda^*)^2}.\end{aligned}$$

Dessas relações temos que

$$\frac{\|g\|}{\delta_n - \lambda^*} \leq \Delta \leq \frac{\|g\|}{\delta_1 - \lambda^*} \quad (2.3)$$

Agora, seja $\varphi(\lambda) = \frac{\|g\|}{\lambda - \lambda^*}$. Como $\lambda^* < \delta_1$, $\lambda^* \notin [\delta_1, \delta_n]$, então neste intervalo φ é contínua. Assim, pelo teorema do valor intermediário, existe $\bar{\lambda} \in [\delta_1, \delta_n]$ tal que $\varphi(\bar{\lambda}) = \Delta$.

Para o caso (ii) temos uma relação parecida com a obtida anteriormente. Primeiro, tomamos $\Delta_0 < \Delta$ tal que $\|s^*\|^2 = \Delta_0^2$. Então, utilizando o mesmo argumento do item anterior temos que

$$\Delta_0^2 \geq \frac{\|g\|^2}{(\delta_n - \lambda^*)^2}.$$

Como $\Delta^2 > \Delta_0^2$, temos a relação

$$\Delta^2 > \frac{\|g\|^2}{(\delta_n - \lambda^*)^2}.$$

Assim temos que

$$\delta_1 - \frac{\|g\|}{\Delta} \leq \lambda^* \leq \delta_n - \frac{\|g\|}{\Delta} \quad \text{quando } \|s^*\| = \Delta$$

e

$$0 = \lambda^* < \delta_n - \frac{\|g\|}{\Delta} \quad \text{quando } \|s^*\| < \Delta.$$

Se conseguirmos cotas inferior (E_{min}) e superior (E_{max}) para os autovalores de H , com $E_{min} \leq \delta_1 \leq \dots \leq \delta_n \leq E_{max}$ temos que $\min\{0, E_{min} - \frac{\|g\|}{\Delta}\} \leq \lambda^* \leq E_{max} - \frac{\|g\|}{\Delta}$. Assim

$$\lambda_L = \min\{0, E_{min} - \frac{\|g\|}{\Delta}\} \quad \text{e} \quad \lambda_U = E_{max} - \frac{\|g\|}{\Delta}$$

são possíveis inicializações para λ_L e λ_U .

Note que como a solução $\lambda^* \leq 0$ e o limitante λ_S é tal que $\lambda^* \leq \delta_1 \leq \lambda_S$, é interessante

que $\lambda_U = \min \left\{ 0, \lambda_S, E_{max} - \frac{\|g\|}{\Delta} \right\}$.

Vale ressaltar que no *hard case*, isto é, se $\lambda^* = \delta_1$, g é ortogonal ao espaço S_1 e $s_{\delta_1} = -(H - \delta_1 I)^\dagger g$ é tal que $\|s_{\delta_1}\| < \Delta$, então

$$\frac{\|g\|}{\delta_n - \delta_1} \leq \|s_{\delta_1}\| < \Delta \quad \Rightarrow \quad \delta_1 \leq \delta_n - \frac{\|g\|}{\Delta}.$$

Como $\delta_1 - \frac{\|g\|}{\Delta} \leq \delta_1$ segue que

$$E_{min} - \frac{\|g\|}{\Delta} \leq \delta_1 = \lambda^* \leq E_{max} - \frac{\|g\|}{\Delta},$$

ou seja, $\lambda_L \leq \delta_1 \leq \lambda_U$.

Para as escolhas de E_{min} e E_{max} , notamos primeiro, através dos círculos de Gerschgorin, que se h_{ij} são os elementos da matriz H e δ é um autovalor de H , então existe $j \in \{1, \dots, n\}$ tal que

$$|\delta - h_{jj}| \leq \sum_{i=1, i \neq j}^n |h_{ij}| = \sum_{i=1}^n |h_{ij}| - |h_{jj}|.$$

Assim temos que

$$h_{jj} - \sum_{i=1, i \neq j}^n |h_{ij}| \leq \delta \leq h_{jj} + \sum_{i=1, i \neq j}^n |h_{ij}|.$$

Então podemos escolher

$$E_{min} = \min_{1 \leq j \leq n} \left\{ h_{jj} - \sum_{i=1, i \neq j}^n |h_{ij}| \right\} \quad \text{e}$$

$$E_{max} = \max_{1 \leq j \leq n} \left\{ h_{jj} + \sum_{i=1, i \neq j}^n |h_{ij}| \right\}.$$

Portanto, a inicialização de λ_S , λ_L e λ_U é dada por:

- $\lambda_S = \min\{h_{ii}\}$ com h_{ii} o i -ésimo elemento da diagonal de H
- $\lambda_U = \min \left\{ 0, \lambda_S, E_{max} - \frac{\|g\|}{\Delta} \right\}$
- $\lambda_L = \min \left\{ 0, E_{min} - \frac{\|g\|}{\Delta} \right\}$

Para salvaguardar o caso $\lambda_L = \lambda_U$ fazemos $\lambda_U = \max\{\lambda_U, (1 + \epsilon)\lambda_L\}$, onde ϵ é uma tolerância.

2.3.3 Atualização dos limitantes

A cada iteração desejamos que o intervalo $[\lambda_L, \lambda_U]$ diminua e que λ_S fique cada vez mais próximo de δ_1 . Para isso devemos atualizar os parâmetros λ_S , λ_U e λ_L durante o processo iterativo.

Para a atualização de λ_S temos dois casos. O primeiro caso ocorre quando temos $\lambda \in (-\infty, \delta_1)$ e $\phi(\lambda) < 0$. Então, tomando τ e $\hat{z}$ como no final da subseção 2.2, temos que $\|R\hat{z}\|^2 \geq \sigma_{min}^2$ com σ_{min} o menor valor singular de R e como $H - \lambda I = R^T R$, então $\sigma_{min}^2 = \delta_1 - \lambda$, com δ_1 o menor autovalor de H .

Assim $\|R\hat{z}\|^2 + \lambda \geq \sigma_{min}^2 + \lambda = \delta_1$, ou seja, $\|R\hat{z}\|^2 + \lambda$ também é um limitante superior para δ_1 . Portanto, uma possível atualização para λ_S é dada por:

$$\lambda_S = \min\{\lambda_S, \lambda + \|R\hat{z}\|^2\}. \quad (2.4)$$

O segundo caso ocorre quando temos $\lambda > \delta_1$, fazendo com que a matriz $H - \lambda I$ tenha pelo menos um autovalor negativo. Nesta situação, ao tentar fatorar a matriz via Cholesky fazendo $H - \lambda I = R^T R$ teremos que, para algum $\ell \leq n$, $\left(b_{\ell\ell} - \sum_{k=1}^{\ell-1} r_{k\ell}^2\right) \leq 0$, onde $b_{\ell\ell}$ é o elemento da ℓ -ésima linha e coluna da matriz $H - \lambda I$. Isso impede prosseguir com a fatoração de Cholesky pois o próximo passo do processo seria calcular $r_{\ell\ell} = \sqrt{b_{\ell\ell} - \sum_{k=1}^{\ell-1} r_{k\ell}^2}$.

Porém, conforme sugerido em [16], podemos expressar a submatriz $\ell \times \ell$ de $H - \lambda I$ como $R_\ell^T M R_\ell$ onde R_ℓ é uma matriz triangular superior de ordem ℓ e $M = \text{diag}(1, 1, \dots, 1, \mu)$ com $\mu < 0$. Assim todos os elementos da matriz R_ℓ podem ser obtidos através da tentativa de realização da fatoração de Cholesky, com exceção do último elemento, que será dado por

$$r_{\ell\ell} = \sqrt{\frac{b_{\ell\ell} - \sum_{k=1}^{\ell-1} r_{k\ell}^2}{\mu}}.$$

Agora, com a escolha $\mu = -1$, seja z solução de $R_\ell z = e_\ell$ com $e_\ell \in \mathbb{R}^\ell$ o ℓ -ésimo vetor canônico. Então para $u = \begin{pmatrix} z^T & 0 \end{pmatrix}^T \in \mathbb{R}^n$ temos que $u^T (H - \lambda I) u = e_\ell^T M e_\ell = \mu$.

Pelo quociente de Rayleigh temos que para qualquer $u \in \mathbb{R}^n$

$$\eta_1 \leq \frac{u^T (H - \lambda I) u}{u^T u} \leq \eta_n$$

com $\eta_1 \leq \dots \leq \eta_n$ os autovalores da matriz $H - \lambda I$.

Em particular, para $u = \begin{pmatrix} z^T & 0 \end{pmatrix}^T$ temos

$$\eta_1 \leq \frac{\mu}{\|z\|^2}.$$

Como $\eta_1 = \delta_1 - \lambda$, temos que $\delta_1 \leq \frac{\mu}{\|z\|^2} + \lambda$, ou seja, temos uma cota superior para δ_1 . Então, uma atualização possível para λ_S neste caso pode ser dada por:

$$\lambda_S = \min \left\{ \lambda_S, \frac{\mu}{\|z\|^2} + \lambda \right\}. \quad (2.5)$$

Uma outra maneira de alcançar este mesmo objetivo é sugerida resumidamente no artigo [29], e detalhada a seguir, com as necessárias adaptações decorrentes das mudanças de sinal introduzidas neste trabalho. Quando $\lambda > \delta_1$, e com a notação acima, se no decorrer da fatoração de Cholesky tivermos $b_{\ell\ell} - \sum_{k=1}^{\ell-1} r_{k\ell}^2 \leq 0$, Moré e Sorensen afirmam que é possível calcular $\delta \geq 0$ tal que a submatriz principal de dimensões $\ell \times \ell$ de $(H - \lambda I + \delta e_\ell e_\ell^T)$ é singular (neste caso, e_ℓ é o ℓ -ésimo vetor canônico de $\mathbb{R}^n$) e que, além disso, é possível calcular $u \in \mathbb{R}^n$ com $u_\ell = -1$, $u_j = 0$, $j > \ell$ tal que

$$(H - \lambda I + \delta e_\ell e_\ell^T)u = 0,$$

seguindo, por um raciocínio análogo ao que fizemos anteriormente na dedução da estratégia (2.5), que a atualização do limitante λ_S é dada por:

$$\lambda_S = \min \left\{ \lambda_S, \frac{\delta}{\|u\|^2} + \lambda \right\}. \quad (2.6)$$

Após a análise do código **Fortran** ([12]), particularmente da rotina **dgqt.f** (disponível para *download* em <http://www-unix.mcs.anl.gov/~more/nmtr/>), notamos que a determinação do escalar δ e do vetor $u \in \mathbb{R}^n$ podem ser feitas como segue. Consideremos o bloco ‘bem sucedido’ da decomposição de Cholesky

$$(H - \lambda I)_{(\ell-1) \times (\ell-1)} = \bar{R}^T \bar{R},$$

a matriz aumentada

$$(H - \lambda I + \delta e_\ell e_\ell^T)_{\ell \times \ell} = \begin{pmatrix} \bar{R}^T \bar{R} & \bar{b} \\ \bar{b}^T & b_{\ell,\ell} + \delta \end{pmatrix}$$

e o sistema homogêneo

$$\begin{pmatrix} \bar{R}^T \bar{R} & \bar{b} \\ \bar{b}^T & b_{\ell,\ell} + \delta \end{pmatrix} \begin{pmatrix} \bar{u} \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

do qual segue que

$$\bar{R}^T \bar{R} \bar{u} = \bar{b} \quad (2.7)$$

e

$$\delta = \bar{b}^T \bar{u} - b_{\ell,\ell}. \quad (2.8)$$

Chamando $\bar{R} \bar{u} = y$, o vetor $\bar{u} \in \mathbb{R}^{\ell-1}$, solução de (2.7), é calculado resolvendo-se os sistemas triangulares $\bar{R}^T y = \bar{b}$ e $\bar{R} \bar{u} = y$. Além disso, de (2.7) segue que

$$\bar{b}^T \bar{u} = \bar{u}^T \bar{R}^T \bar{R} \bar{u} = y^T y$$

e portanto, (2.8) pode ser reescrito como

$$\delta = y^T y - b_{\ell,\ell},$$

o que completa a dedução dos elementos necessários para efetuar (2.6).

Assim, a atualização dos limitantes λ_S , λ_U e λ_L será dada por:

1. Se $\lambda \in (-\infty, \delta_1)$ e $\phi(\lambda) < 0$ então

$$\lambda_L = \max\{\lambda_L, \lambda\}$$

caso contrário

$$\lambda_U = \min\{\lambda_U, \lambda\}$$

2. Atualizar λ_S usando (2.4) ou (2.5) ou (2.6)
3. $\lambda_U = \min\{\lambda_U, \lambda_S\}$

2.3.4 Critérios de parada

Este algoritmo pode terminar em três situações: se encontrar durante o processo iterativo s suficientemente próximo da fronteira da região de confiança; se H é definida positiva

e s dado pelo algoritmo é solução do problema; ou se o *hard case* for detectado e então obtemos τ e $\hat{z}$ como em (2.2).

Assim, dados σ_1 e σ_2 em $(0, 1)$ e $\lambda \leq 0$ tal que $H - \lambda I$ é definida positiva e s é o vetor calculado no passo 2 do método de Newton para o menor zero da função secular, então o algoritmo termina se

$$|\Delta - \|s\|| \leq \sigma_1 \Delta \quad \text{ou} \quad \|s\| \leq \Delta \text{ e } \lambda = 0 \quad (2.9)$$

ou

$$\|R(\tau \hat{z})\|^2 \leq \sigma_1(2 - \sigma_1) \max\{\sigma_2, \|Rs\|^2 - \lambda \Delta^2\} \quad (2.10)$$

No caso (2.10) o algoritmo termina e $s + \tau \hat{z}$ é a aproximação desejada.

2.3.5 Algoritmo de Moré e Sorensen

A seguir apresentamos a esquematização do algoritmo de Moré e Sorensen implementado em **Matlab** com os elementos necessários para a implementação descritos ao longo deste capítulo.

Algoritmo de Moré e Sorensen

Dados Δ , H , g , tomamos $s = 0$.

Enquanto o critério de parada não for satisfeito

- Garantimos λ no intervalo de salvaguarda conforme 2.3.1
- Tentamos realizar a fatoração de Cholesky da matriz $H - \lambda I$
- Se a fatoração é realizada com sucesso
 1. Calculamos $\hat{z}$ e se $\|s\| < \Delta$, calculamos τ conforme (2.2)
 2. Verificamos se os critérios de convergência são satisfeitos conforme 2.3.4
 3. Atualizamos λ_S conforme (2.4) e λ_L e λ_U conforme descrito em 2.3.3
 4. Atualizamos λ utilizando um passo do método de Newton conforme descrito em 2.1
- Se a fatoração não pode ser realizada
 1. Atualizamos λ_S conforme (2.5) ou (2.6) e λ_L e λ_U conforme descrito em 2.3.3
 2. Verificamos se os critérios de convergência são satisfeitos conforme 2.3.4

2.4 Convergência global

O método de região de confiança para minimização sem restrições que utiliza o algoritmo de Moré e Sorensen para resolver os subproblemas apresenta alguns resultados importantes de convergência.

Se utilizamos o algoritmo de Moré e Sorensen para resolver o subproblema do passo 2 do algoritmo RCMI e a aproximação s_k da solução do subproblema satisfaz (2.9) ou (2.10) então é mostrado em [29] que s_k satisfaz

$$\psi_k(s_k) - \psi_k^* \leq \sigma_1(2 - \sigma_1) \max\{|\psi_k^*|, \sigma_2\}, \quad \|s_k\| \leq (1 + \sigma_1)\Delta_k.$$

Uma consequência desta relação é que, se $\psi_k^* \neq 0$, então a aproximação s_k dada pelo algoritmo de Moré e Sorensen com $\sigma_2 = 0$ satisfaz

$$-\psi_k(s_k) \geq \beta_1 |\psi_k^*| \quad \text{com } \|s_k\| \leq \beta_2 \Delta_k \quad (2.11)$$

onde $\beta_1, \beta_2 > 0$.

Com isso temos um primeiro resultado de convergência do algoritmo de região de confiança RCMI expresso pelo teorema a seguir, e cuja demonstração é feita em [29].

Teorema 1 Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ duas vezes continuamente diferenciável no conjunto de nível $\Omega = \{x | f(x) \leq f(x_0)\}$ e considere a sequência $\{x_k\}$ produzida pelo algoritmo RCMI, onde s_k satisfaz (2.11). Se Ω é um conjunto compacto então, ou o algoritmo termina em x_ℓ com $\nabla f(x_\ell) = 0$ e $\nabla^2 f(x_\ell)$ semidefinida positiva, ou $\{x_k\}$ tem um ponto limite $x^* \in \Omega$ com $\nabla f(x^*) = 0$ e $\nabla^2 f(x^*)$ semidefinida positiva.

Um segundo resultado importante de convergência não depende da compacidade do conjunto de nível Ω e é dado no próximo teorema, conforme [29].

Teorema 2 Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ duas vezes continuamente diferenciável no conjunto de nível $\Omega = \{x | f(x) \leq f(x_0)\}$ e considere a sequência $\{x_k\}$ produzida pelo algoritmo RCMI, onde s_k satisfaz (2.11). Se x^* é um ponto limite de $\{x_k\}$ e $\nabla^2 f(x^*)$ é não singular então $\{x_k\}$ converge para x^* e $\nabla^2 f(x^*)$ é definida positiva.

Este teorema pode ser estendido, com a hipótese adicional $\|s_k\| \leq \beta_3 \Delta_k$ onde $\beta_3 > 0$, para mostrar que $\{x_k\}$ converge para x^* com taxa de convergência q -superlinear e, se $\nabla^2 f$ é Lipschitz contínua, então a taxa de convergência é quadrática.

Capítulo 3

Algoritmo de Rojas, Santos e Sorensen

O algoritmo proposto em 2000 por Rojas, Santos e Sorensen [37] também pode ser utilizado para a resolução dos subproblemas (1.2) que aparecem no contexto de algoritmos de região de confiança.

A principal idéia deste algoritmo é, com base no Lema 1, reescrever o subproblema de região de confiança como um problema de autovalores parametrizado, e encontrar um valor ótimo para o parâmetro com o auxílio de um esquema de interpolação. A solução do subproblema estará relacionada então aos autovetores associados aos menores autovalores de um problema de autovalor com o parâmetro ótimo, que veremos com maiores detalhes neste capítulo.

Uma característica importante deste método é a utilização apenas de produtos de matriz por vetor, não sendo necessário o armazenamento explícito das matrizes envolvidas no processo.

3.1 Um problema de autovalores equivalente

Para deduzir o método proposto por Rojas, Santos e Sorensen primeiro consideramos a matriz $B_\alpha = \begin{pmatrix} \alpha & g^T \\ g & H \end{pmatrix}$. Assim temos que o problema (1.2) pode ser reescrito como:

$$\begin{aligned} \min \quad & \frac{1}{2} y^T B_\alpha y \\ \text{s. a.} \quad & y^T y \leq 1 + \Delta^2 \\ & e_1^T y = 1, \end{aligned}$$

onde e_1 é o primeiro vetor canônico de $\mathbb{R}^{n+1}$.

Supondo λ autovalor de B_α com autovetor associado $(1, s^T)^T$ temos que

$$\alpha - \lambda = -g^T s \quad \text{e} \quad (H - \lambda I)s = -g.$$

Observando o Lema 1 vemos que, se $H - \lambda I$ é semidefinida positiva, s pode ser um candidato à solução do subproblema (1.2), dependendo do sinal de λ e do produto $\lambda(\Delta - \|s\|)$ ser zero.

Porém, pela propriedade de entrelaçamento de Cauchy (cf. [17], pag. 396) temos que, se $\delta_1 \leq \dots \leq \delta_n$ são os autovalores de H e $\lambda_1(\alpha) \leq \dots \leq \lambda_{n+1}(\alpha)$ são autovalores de B_α então

$$\lambda_1(\alpha) \leq \delta_1 \leq \lambda_2(\alpha) \leq \dots \leq \lambda_n(\alpha) \leq \delta_n \leq \lambda_{n+1}(\alpha).$$

Assim $\lambda_1(\alpha) \leq \delta_1$, ou seja, os autovalores da matriz $H - \lambda_1(\alpha)I$ são sempre maiores ou iguais a zero, fazendo com que esta matriz seja semidefinida positiva, independentemente do valor de α .

Então, definindo

$$\phi(\lambda) = g^T (H - \lambda I)^\dagger g = -g^T s$$

temos

$$\phi'(\lambda) = g^T ((H - \lambda I)^\dagger)^2 g = s^T s$$

onde † denota a pseudoinversa da matriz. A idéia do algoritmo é ajustar α para que $\phi'(\lambda) = s^T s = \Delta^2$ e então encontrar o menor autovalor de B_α , normalizando a primeira componente do autovetor correspondente e obtendo s através das demais componentes.

Se $\lambda_1(\alpha) \leq 0$, então s será uma solução para o subproblema (1.2).

Durante o ajuste de α , se encontrarmos $\lambda > 0$ com $\|s\| < \Delta$, então H é definida positiva e $\|H^{-1}g\| < \Delta$. Nessa situação o problema tem uma única solução no interior da região de confiança satisfazendo $Hz = -g$.

Para o ajuste de α utilizamos um esquema de interpolação que está apresentado em maiores detalhes na próxima seção. A principal idéia para estimar α é utilizar uma função interpolante $\hat{\phi}(\lambda)$ e encontrar $\hat{\lambda}$ tal que $\hat{\phi}'(\hat{\lambda}) = \Delta^2$. Então α será dado pela relação $\alpha - \hat{\lambda} = \hat{\phi}(\hat{\lambda})$. Com a estimativa de α , é resolvido então o problema de encontrar o menor autovalor de B_α e seu correspondente autovetor.

Na Figura 5 temos um exemplo ilustrativo, no qual temos solução para $\phi'(\lambda) = \Delta^2$ com $\lambda \leq \delta_1$, onde

$$\phi(\lambda) = -g^T s = \sum_{j=1}^d \frac{\beta_j^2}{\delta_j - \lambda}$$

com β_j^2 a soma dos quadrados dos coeficientes da expansão de g na base dos autovetores associados com os autovalores distintos δ_j , $j = 1, \dots, d$. A reta representa $y = \alpha_* - \lambda$.

Neste primeiro caso, vemos que sempre temos solução para $\phi'(\lambda) = \Delta^2$ com $\lambda \leq \delta_1$, independentemente do valor de Δ .

Para estimar o menor autovalor de B_α e um de seus autovetores é utilizado um método tipo Lanczos (cf. [42]).

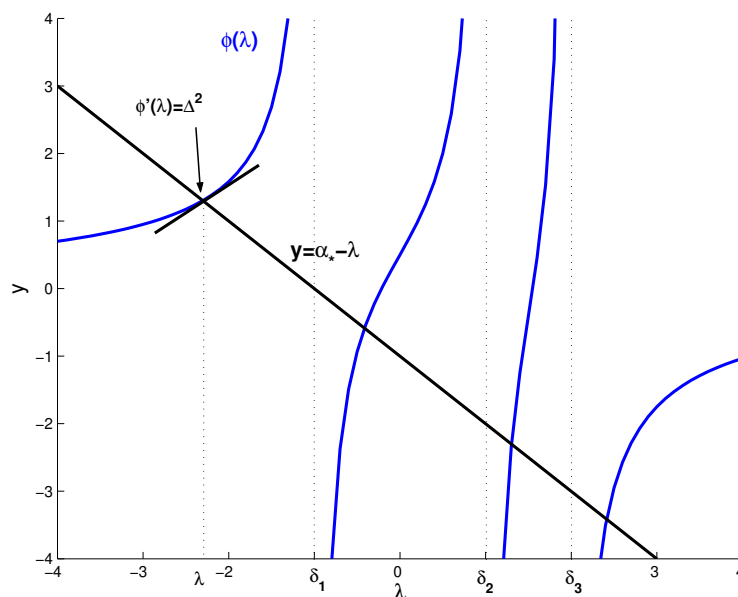


Figura 5 - Exemplo de $\phi(\lambda)$ onde a solução de $\phi'(\lambda) = \Delta^2$ com $\lambda \leq \delta_1$ é sempre possível.

3.2 Tratamento do hard case

Uma observação importante é que, se utilizássemos a verdadeira função ϕ ao invés da função interpolante $\hat{\phi}$, $\hat{\lambda}$ poderia ser exatamente o autovalor procurado. Assim, ao aproximar ϕ por $\hat{\phi}$, procuramos obter $\hat{\lambda}$ tal que $\hat{\lambda} \leq \delta_1$. Porém, isso não é sempre trivial.

Uma situação que pode ocorrer é o fato dos autovetores associados ao menor autovalor de B_α terem primeira componente nula. Neste caso podemos não ter solução para $\phi'(\lambda) = \Delta^2$ com $\lambda \leq \delta_1$, o que caracteriza o *hard case*.

Na Figura 6 temos um exemplo do *hard case*, associado à inexistência do pólo da função $\phi(\lambda)$ em δ_1 (já que $\beta_1^2 = 0$). Neste caso, dependendo do valor de Δ , podemos não ter solução com $\lambda \leq \delta_1$.

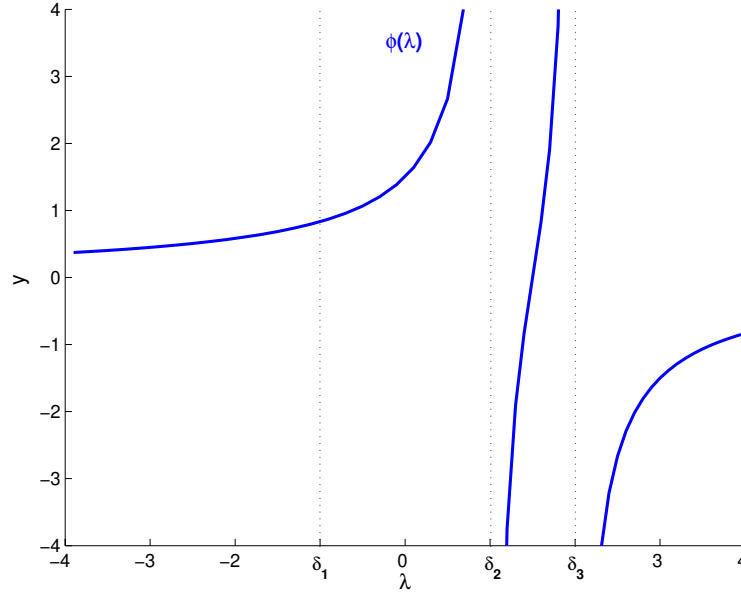


Figura 6 - Exemplo de $\phi(\lambda)$ para um problema em que ocorre o *hard case*, com a ausência do pólo de $\phi(\lambda)$ em δ_1 .

A ausência do pólo em δ_1 está associada ao fato de g ser ortogonal a S_1 onde $S_j = \{q \in \mathbb{R}^n | Hq = \delta_j q\}$, $j = 1, \dots, d$, conforme estabelece o resultado a seguir.

Lema 4 Para qualquer $\alpha \in \mathbb{R}$ e $q \in S_j$, $1 \leq j \leq d$, $\{\delta_j, (0, q^T)^T\}$ é um autopar de B_α se e somente se g é ortogonal a S_j .

Uma consequência importante do *hard case* é o fato dos autovetores associados ao menor autovalor da matriz B_α terem primeira componente nula, exceto para um valor especial do parâmetro α . Este fato é formalizado a seguir, cuja prova pode ser vista em [37].

Lema 5 Seja g ortogonal a S_j , $1 \leq j < d$ e $p_j = -(H - \delta_j I)^\dagger g$. Então $\{\delta_j, (1, p_j^T)^T\}$ é um autopar de B_α se e somente se $\alpha = \tilde{\alpha}_j$ onde $\tilde{\alpha}_j = \delta_j - g^T p_j$.

Um outro resultado importante é dado no próximo teorema, cuja demonstração também está em [37].

Teorema 3 Seja $\lambda(\alpha)$ a menor solução da equação

$$\phi(\lambda) = \alpha - \lambda$$

Então, para qualquer valor de α , $\lambda(\alpha)$ é autovalor de B_α com um correspondente autovetor que pode ser escalado para ter primeira componente igual a 1.

No *hard case* o que acontece é que a solução do Teorema 3 pode não corresponder a $\lambda_1(\alpha)$. De fato, isso só ocorrerá para $\alpha = \tilde{\alpha}_1$ dado pelo Lema 5. Para outros valores de α , $\lambda_1(\alpha)$ estará relacionado com autovetores com primeira componente nula, mas como visto antes, isso só ocorre quando g é ortogonal a S_1 .

Então, quando g é ortogonal a S_1 temos um *hard case potencial*. Para caracterizar os $\ell + 1$ menores autovalores de B_α se g é ortogonal aos autoespaços correspondentes aos menores ℓ autovalores distintos de H , temos o resultado a seguir, que está demonstrado em [37].

Lema 6 Seja $\{\lambda(\alpha), v(\alpha)\}$ o autopar de B_α dado pelo Teorema 3 e $\tilde{\alpha}_i$ como no Lema 4.

Se g não é ortogonal a S_1 , então $\lambda_1(\alpha) = \lambda(\alpha)$.

Se g é ortogonal a S_k para $k = 1, \dots, \ell$ com $1 \leq \ell < d$, então:

- i. se $\alpha = \tilde{\alpha}_i$, $1 \leq i \leq \ell$ então $\lambda_j(\alpha) = \delta_j$, $j = 1, \dots, \ell$. Neste caso $\lambda_{\ell+1}(\alpha)$ é a segunda menor raiz da equação $\phi(\lambda) = \alpha - \lambda$;
- ii. se $\alpha < \tilde{\alpha}_1$, então $\lambda_1(\alpha) = \lambda(\alpha)$ e $\lambda_j(\alpha) = \delta_{j-1}$, $j = 2, \dots, \ell + 1$;
- iii. se $\tilde{\alpha}_{i-1} < \alpha < \tilde{\alpha}_i$, $2 \leq i \leq \ell$ então $\lambda_i(\alpha) = \lambda(\alpha)$, $\lambda_j(\alpha) = \delta_j$ para $j = 1, \dots, i-1$ e $\lambda_j(\alpha) = \delta_{j-1}$ para $j = i+1, \dots, \ell + 1$;
- iv. se $\alpha > \tilde{\alpha}_\ell$, então $\lambda_j(\alpha) = \delta_j$, $j = 1, \dots, \ell$ e $\lambda_{j+1}(\alpha) = \lambda(\alpha)$.

A Figura 7 ilustra os casos do Lema 6 para $\ell = 2$ e diferentes possibilidades para α . Vemos que mesmo com g ortogonal a S_1 podemos ter solução com $\lambda(\alpha) < \delta_1$ como ocorre no caso ii. Porém, nos demais casos, $\lambda_1(\alpha) = \delta_1$ e temos potenciais *hard cases*.

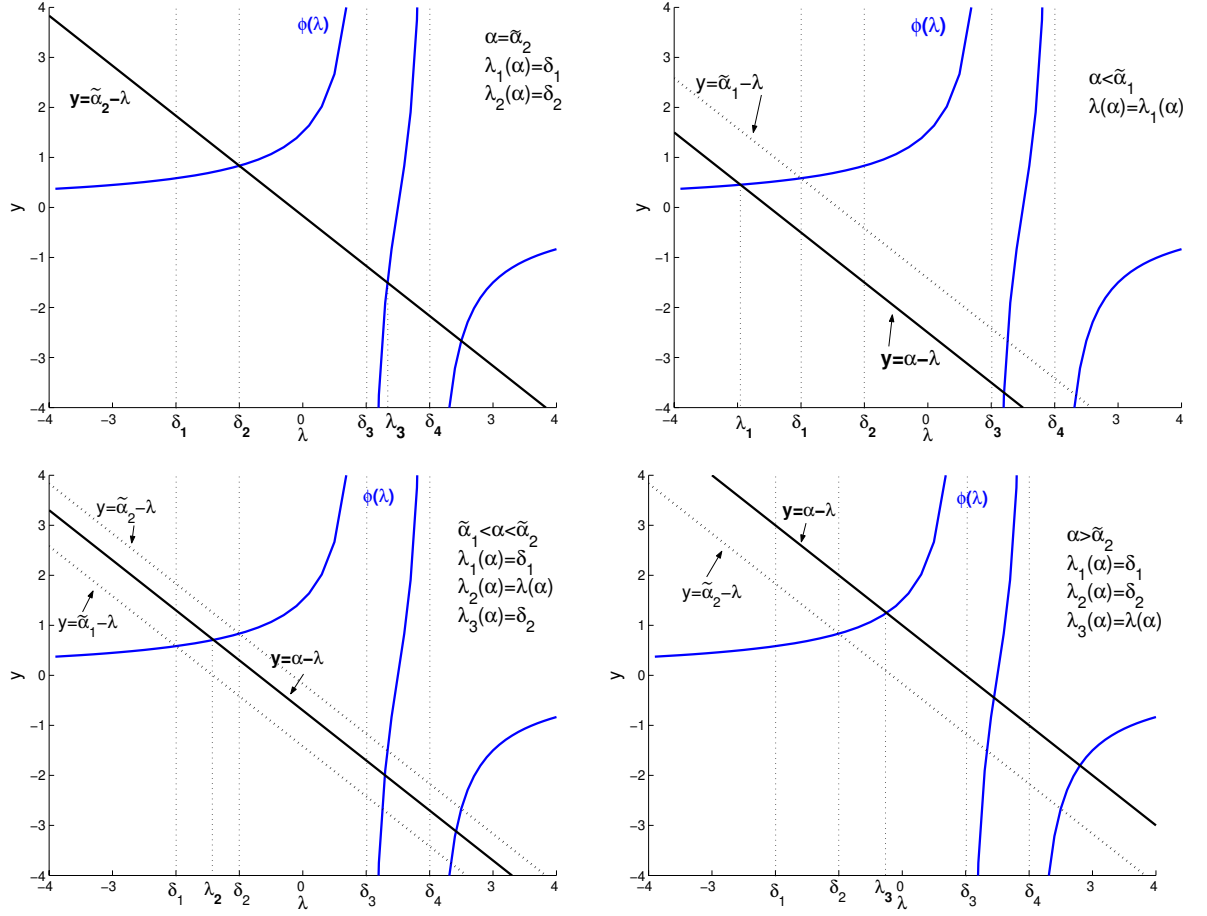


Figura 7 - Casos abordados no Lema 6. i) $\alpha = \tilde{\alpha}_2$ ii) $\alpha < \tilde{\alpha}_1$ iii) $\tilde{\alpha}_1 < \alpha < \tilde{\alpha}_2$ iv) $\alpha > \tilde{\alpha}_2$

Quando é detectado *hard case* no algoritmo podemos obter uma aproximação para a solução com o cálculo baseado nos resultados a seguir.

O Teorema 4 abaixo, apresentado e demonstrado em [37], estabelece que, sob determinadas condições conseguimos uma solução bem próxima da solução verdadeira do subproblema (1.2) quando é detectado o *hard case*. Essa solução aproximada é formada por uma combinação linear de componentes de dois autovetores de B_α , que são obtidos sem maiores custos durante as iterações do algoritmo.

Teorema 4 Seja $z_1 = (\nu_1, \tilde{z}_1^T)^T$ um autovetor de B_α associado a $\lambda_1(\alpha)$. Seja $\lambda_i(\alpha)$ qualquer outro autovalor de B_α com correspondente autovetor $z_i = (\nu_i, \tilde{z}_i^T)^T$. Definimos $Z = \begin{bmatrix} z_1 & z_i \end{bmatrix}$, $\tilde{Z} = \begin{bmatrix} \tilde{z}_1 & \tilde{z}_i \end{bmatrix}$, assumindo que $Z^T Z = I$. Seja $\eta > 0$.

Se existe $t = (\tau_1, \tau_2)^T$ com $\|t\| = 1$ tal que

$$1. (e_1^T Z t)^2 = \frac{1}{1+\Delta^2} \text{ e}$$

$$2. (\lambda_i(\alpha) - \lambda_1(\alpha))\tau_2^2(1 + \Delta^2) \leq -2\eta\psi(\tilde{s}) \text{ para } \tilde{s} = \frac{\tilde{Z}t}{e_1^T Zt},$$

então

$$\psi(s_*) \leq \psi(\tilde{s}) \leq \frac{1}{1 + \eta}\psi(s_*),$$

onde s_* é uma solução na fronteira para o problema (1.2), com $\psi(s_*) \leq 0$.

Uma consequência importante do Teorema 4 é que $\tilde{s}$ será um ponto quase ótimo, já que $\psi(\tilde{s})$ está relativamente próximo de $\psi(s^*)$, de acordo com o parâmetro $\eta > 0$.

Sob determinadas condições é possível calcular o vetor $t \in \mathbb{R}^2$ do Teorema 4. Nos próximos dois lemas, cujas provas podem ser vistas em [37], são estabelecidas as condições para o cálculo de t e como efetuar este cálculo.

Lema 7 Seja $z_i = (\nu_i, \tilde{z}_i^T)^T$, com $\nu_i \in \mathbb{R}$, $\tilde{z}_i \in \mathbb{R}^n$ para $i = 1, 2$. Definimos as matrizes $Z = \begin{bmatrix} z_1 & z_2 \end{bmatrix}$ e $\tilde{Z} = \begin{bmatrix} \tilde{z}_1 & \tilde{z}_2 \end{bmatrix}$, assumindo que $Z^T Z = I$.

Se $\|Z^T e_1\|^2 \geq \frac{1}{\beta}$ para $\beta > 0$, então existe $t \in \mathbb{R}^2$ com $t \neq 0$ que satisfaz

$$\|Zt\|^2 = \beta(e_1^T Zt)^2.$$

Ao escolher $\beta = 1 + \Delta^2$ e normalizar t para que $\|t\| = 1$, a primeira condição do Teorema 4 será satisfeita.

O próximo lema nos fornece um caminho para calcular este vetor t .

Lema 8 Sejam $\beta \in \mathbb{R}$, $\beta > 0$ e $z \in \mathbb{R}^n$. A equação

$$t^T[I - \beta z z^T]t = 0$$

em t com $t \in \mathbb{R}^n$ tem $2(n - 1)$ soluções não triviais se a matriz $M = I - \beta z z^T$ é indefinida e tem uma solução não trivial se M é semidefinida positiva e singular.

Estas soluções não triviais são dadas por $t = Py$ onde $P \in \mathbb{R}^{n \times n}$, tal que $P^T z = \|z\|e_1$ com $P^T P = I$, e

- Se $\theta = -1 + \beta\|z\|^2 > 0$ então

$y = (1, \sqrt{\theta}e_i^T)^T$ e $y = (-1, \sqrt{\theta}e_i^T)^T$ com e_i o i -ésimo vetor canônico em $\mathbb{R}^{n-1}$, $i = 1, \dots, n - 1$ (M é indefinida);

- Se $\theta = -1 + \beta\|z\|^2 = 0$ então $y = e_1$ (M é semidefinida positiva e singular).

Assim, no algoritmo, podemos computar o vetor t como no Lema 8 com $\beta = 1 + \Delta^2$, se as condições do Lema 7 estão verificadas. Caso também se verifique a condição 2 do Teorema 4, já que a condição 1 estará automaticamente verificada, então temos satisfeito o critério de parada e a aproximação para a solução será dada por $\tilde{s}$.

3.3 Ajuste do parâmetro α

Para o ajuste de α é utilizado um esquema de interpolação a cada iteração. A idéia do algoritmo é obter λ tal que $\phi'(\lambda) = \Delta^2$ e então atualizar o parâmetro α correspondente através da relação $\alpha - \lambda = \phi(\lambda)$.

Para isso, a cada iteração, é utilizada uma função $\hat{\phi}(\lambda)$ que interpola ϕ e ϕ' em dois pontos escolhidos adequadamente. Então, com este esquema de interpolação em cada iteração k , são obtidos $\hat{\lambda}$ e α_{k+1} tais que $\hat{\phi}'(\hat{\lambda}) = \Delta^2$ e $\alpha_{k+1} - \hat{\lambda} = \hat{\phi}(\hat{\lambda})$. Com esta atualização α_{k+1} é computada uma estimativa para o menor autovalor da matriz $B_{\alpha_{k+1}}$ através de um método tipo Lanczos.

No início do processo temos apenas o conhecimento de um ponto s_0 com correspondente λ_0 e α_0 . Assim, inicialmente, é utilizado um esquema de interpolação com um único ponto.

Definindo $\hat{\phi} = \frac{\gamma^2}{\delta - \lambda}$ e impondo que $\hat{\phi}(\lambda_0) = \phi(\lambda_0) = -g^T s_0$ e $\hat{\phi}'(\lambda_0) = \phi'(\lambda_0) = s_0^T s_0$, temos que $\delta = \lambda_0 - \frac{g^T s_0}{s_0^T s_0}$ e $\gamma^2 = \frac{(g^T s_0)^2}{s_0^T s_0}$. Assim, temos $\hat{\lambda}$ e α_1 , que satisfazem $\hat{\phi}'(\hat{\lambda}) = \Delta^2$ e $\alpha_1 - \hat{\lambda} = \hat{\phi}(\hat{\lambda})$, dados por:

$$\begin{aligned} \hat{\lambda} &= \delta + \frac{g^T s_0}{\|s_0\| \Delta} \quad \text{e} \\ \alpha_1 &= \alpha_0 + \frac{\alpha_0 - \lambda_0}{\|s_0\|} \left(\frac{\Delta - \|s_0\|}{\Delta} \right) \left(\Delta + \frac{1}{\|s_0\|} \right). \end{aligned} \quad (3.1)$$

Para obter uma melhor aproximação com o esquema de interpolação, a partir da segunda iteração é utilizado um esquema com dois pontos, utilizando as informações $\phi(\lambda_{k-1})$, $\phi'(\lambda_{k-1})$, $\phi(\lambda_k)$ e $\phi'(\lambda_k)$.

Para isso definimos $\hat{\phi}(\lambda) = \frac{\gamma^2}{\delta - \lambda} + \eta$. Então $\hat{\phi}'(\lambda) = \frac{\gamma^2}{(\delta - \lambda)^2}$.

Utilizando a informação da derivada da função interpolante, pedimos que $\hat{\phi}'(\lambda_{k-1}) = \phi'(\lambda_{k-1}) = \|s_{k-1}\|^2$ e $\hat{\phi}'(\lambda_k) = \phi'(\lambda_k) = \|s_k\|^2$.

Destas relações temos então que

$$\begin{aligned} \gamma^2 &= \frac{(\lambda_k - \lambda_{k-1})^2 \|s_{k-1}\|^2 \|s_k\|^2}{(\|s_k\| - \|s_{k-1}\|)^2} \quad \text{e} \\ \delta &= \frac{\lambda_k \|s_k\| - \lambda_{k-1} \|s_{k-1}\|}{\|s_k\| - \|s_{k-1}\|}. \end{aligned}$$

E então $\hat{\lambda}$ é calculado para que $\hat{\phi}'(\hat{\lambda}) = \Delta^2$ (ou $\frac{1}{\sqrt{\hat{\phi}'(\hat{\lambda})}} = \frac{1}{\Delta}$). Assim

$$\hat{\lambda} = \frac{\lambda_{k-1}\|s_{k-1}\|(\|s_k\| - \Delta) + \lambda_k\|s_k\|(\Delta - \|s_{k-1}\|)}{\Delta(\|s_k\| - \|s_{k-1}\|)}.$$

Para calcular α_{k+1} utilizamos a relação $\alpha_{k+1} = \hat{\lambda} + \hat{\phi}(\hat{\lambda})$, onde $\hat{\phi}(\hat{\lambda}) = \frac{\gamma^2}{\delta - \hat{\lambda}} + \eta$.

Para a definição de η utilizamos outro esquema de interpolação buscando que η se aproxime de $\phi(\hat{\lambda}) - \frac{\gamma^2}{\delta - \hat{\lambda}}$ para que $\hat{\phi}(\hat{\lambda}) \approx \phi(\hat{\lambda})$.

Para isso definimos $\eta_j = \phi(\lambda_j) - \frac{\gamma^2}{\delta - \lambda_j}$ para $j = k-1, k$ e $\eta = \eta(\hat{\lambda})$ onde $\eta(\lambda)$ é uma função interpolante em λ_{k-1} e λ_k tal que $\eta(\lambda_{k-1}) = \eta_{k-1}$ e $\eta(\lambda_k) = \eta_k$, ou seja

$$\eta(\lambda) = \left(\frac{\lambda_k - \lambda}{\lambda_k - \lambda_{k-1}} \right) \eta_{k-1} + \left(\frac{\lambda - \lambda_{k-1}}{\lambda_k - \lambda_{k-1}} \right) \eta_k$$

Assim temos que

$$\eta = \left(\frac{\lambda_k - \hat{\lambda}}{\lambda_k - \lambda_{k-1}} \right) \eta_{k-1} + \left(\frac{\hat{\lambda} - \lambda_{k-1}}{\lambda_k - \lambda_{k-1}} \right) \eta_k$$

e então

$$\alpha_{k+1} = \omega \alpha_{k-1} + (1 - \omega) \alpha_k + \frac{\|s_{k-1}\| \|s_k\| (\|s_k\| - \|s_{k-1}\|)}{\omega \|s_k\| + (1 - \omega) \|s_{k-1}\|} \frac{(\lambda_{k-1} - \hat{\lambda})(\lambda_k - \hat{\lambda})}{(\lambda_k - \lambda_{k-1})} \quad (3.2)$$

onde $\omega = \frac{\lambda_k - \hat{\lambda}}{\lambda_k - \lambda_{k-1}}$, $\alpha_{k-1} = \lambda_{k-1} + \phi(\lambda_{k-1})$ e $\alpha_k = \lambda_k + \phi(\lambda_k)$.

A escolha dos pontos de interpolação s_k com correspondentes λ_k e α_k sofre algumas modificações no algoritmo quando é detectada uma situação próxima ao *hard case*.

Quando temos o *hard case*, o menor autovalor da matriz B_α é igual a δ_1 e está associado a autovetores com primeira componente nula. Neste caso, para o esquema de interpolação é utilizada uma aproximação para o segundo menor autovalor de B_α tal que a primeira componente de um autovetor correspondente possa ser normalizada.

Uma situação de *hard case potencial* é detectada quando ν_1 é pequeno, com $(\nu_1, u_1^T)^T$ um autovetor unitário de B_{α_k} correspondente a $\lambda_1(\alpha_k)$.

Como $(\nu_1, u_1^T)^T$ é autovetor associado a $\lambda_1(\alpha_k)$, então temos que $(H - \lambda_1(\alpha_k)I)u_1 = -g\nu_1$ e então $\|(H - \lambda_1(\alpha_k)I)u_1\| = |\nu_1|\|g\|$. Como $\|(\nu_1, u_1^T)^T\| = 1$, segue que $\|u_1\| = \sqrt{1 - \nu_1^2}$ e então

$$\frac{\|(H - \lambda_1(\alpha_k)I)u_1\|}{\|u_1\|} = \frac{|\nu_1|\|g\|}{\sqrt{1 - \nu_1^2}}.$$

Assim $|\nu_1|\|g\| \leq \varepsilon \sqrt{1 - \nu_1^2}$ com $\varepsilon \in (0, 1)$ garante que $\|(H - \lambda_1(\alpha_k)I)u_1\| \leq \varepsilon \|u_1\|$, ou

seja, $\{\lambda_1(\alpha_k), u_1\}$ é uma aproximação para um autopar de H , o que nos indica que ν_1 é próximo de zero.

Logo, no algoritmo, detectamos uma situação de *hard case potencial* quando $|\nu_1| \|g\| \leq \varepsilon \sqrt{1 - \nu_1^2}$.

Quando essa situação é detectada podemos também verificar $|\nu_2|$ e, se $|\nu_1|$ e $|\nu_2|$ são pequenos, fazemos o parâmetro α_k sofrer um decréscimo.

3.4 Salvaguardas e critérios de parada

Para garantir a convergência de α para o valor ótimo α^* , temos no algoritmo um esquema de salvaguarda com a introdução do intervalo $[\alpha_L, \alpha_U]$.

A cada iteração este intervalo é reduzido com a atualização dos limitantes α_L e α_U e então é verificado se α está neste intervalo.

3.4.1 Inicialização dos parâmetros

Para a inicialização dos parâmetros α_L , α_U e α , primeiro supomos que λ^* e s^* satisfazem as condições do Lema 1, sendo este par correspondente à solução do subproblema (1.2). Se a solução s^* não está no interior da região de confiança então temos que $s^* = -(H - \lambda^* I)^\dagger g$ tal que $\|s^*\| = \Delta$.

Como $\alpha^* = \lambda^* - g^T s^*$, em Rendl e Wolkowicz [35] é dada a relação:

$$\delta_1 - \frac{\|g\|}{\Delta} \leq \alpha^* \leq \delta_1 + \|g\| \Delta. \quad (3.3)$$

Pelo quociente de Rayleigh temos que

$$\delta_1 \leq \frac{v^T H v}{v^T v}$$

para qualquer $v \in \mathbb{R}^n$. Então, $\alpha^* \leq \delta_1 + \|g\| \Delta \leq \frac{v^T H v}{v^T v} + \|g\| \Delta$.

Em especial, se v é um vetor canônico e_i , temos que o elemento da diagonal h_{ii} da matriz H é uma cota superior para δ_1 . Assim, se é possível avaliar explicitamente a diagonal de H , tomamos $\delta_U = \min\{h_{ii}, i = 1, \dots, n\}$, caso contrário tomamos $\delta_U = \frac{v^T H v}{v^T v}$ com v um vetor com componentes aleatórias. E então $\alpha_U = \delta_U + \|g\| \Delta$ é um limitante superior para α^* .

Para a inicialização de α_L , supondo que já temos α_0 , uma estimativa inicial para α^* , podemos então obter $\lambda_1(\alpha_0)$, o menor autovalor da matriz B_{α_0} . Pela propriedade de entrelaçamento temos que $\lambda_1(\alpha_0) \leq \delta_1$. Assim, por (3.3), $\alpha_L = \lambda_1(\alpha_0) - \frac{\|g\|}{\Delta}$ é um limitante

inferior para α^* .

Agora, para inicializar α_0 devemos levar em consideração que, se $\alpha \leq 0$, então B_α não é definida positiva. Assim escolhemos $\alpha_0 \leq 0$ para garantir que $\lambda_1(\alpha_0) \leq 0$. Uma escolha simples é tomar $\alpha_0 = \min\{0, \alpha_U\}$.

Assim temos as inicializações:

$$\begin{aligned}\alpha_U &= \delta_U + \|g\|\Delta, \\ \alpha_0 &= \min\{0, \alpha_U\} \quad \text{e} \\ \alpha_L &= \lambda_1(\alpha_0) - \frac{\|g\|}{\Delta}.\end{aligned}$$

3.4.2 Atualização de α_L e α_U

Para a atualização de α_U são utilizadas informações do menor autovalor e dos correspondentes autovetores da matriz B_α em cada iteração, fazendo

$$\delta_U = \min \left\{ \delta_U, \frac{u_1^T H u_1}{u_1^T u_1} \right\}$$

onde $\frac{u_1^T H u_1}{u_1^T u_1} = \lambda_1(\alpha) - \nu_1 \frac{g^T u_1}{u_1^T u_1}$.

Quando um potencial *hard case* é detectado, temos que $\{\lambda_1(\alpha), u_1\}$ aproxima um autopar de H , então $\lambda_1(\alpha)$ é uma boa aproximação para δ_1 . Logo, δ_U torna-se uma boa estimativa para δ_1 neste caso.

Ao atualizar α_L ou α_U , a cada iteração, o tamanho do intervalo $[\alpha_L, \alpha_U]$ é reduzido e, caso α_{k+1} obtido pelo esquema de interpolação não pertença ao intervalo, então tomamos $\alpha_{k+1} = \frac{\alpha_L + \alpha_U}{2}$.

3.4.3 Critérios de parada

Como no algoritmo anterior, um dos critérios de parada é dado pela obtenção de um ponto próximo da fronteira da região de confiança com o sinal correto do multiplicador de Lagrange:

$$|\Delta - \|s_k||| \leq \varepsilon_\Delta \Delta \quad \text{e} \quad \lambda_1(\alpha_k) \leq 0, \quad (3.4)$$

onde $\varepsilon_\Delta \in (0, 1)$.

Quando (3.4) ocorre, então o algoritmo pára e a solução é dada por $\lambda^* = \lambda_1(\alpha_k)$ e $s^* = s_k$.

O segundo critério de parada é definido para detectar uma solução no interior da região

de confiança:

$$\|u_1\| < \Delta|\nu_1| \quad \text{e} \quad \lambda_1(\alpha_k) > -\varepsilon_{int}, \quad (3.5)$$

onde $\varepsilon_{int} \in (0, 1)$ e $(\nu_1, u_1^T)^T$ é o autovetor correspondente a $\lambda_1(\alpha_k)$. Neste caso a solução é dada por $\lambda^* = 0$ e s^* é o vetor satisfazendo $HS^* = -g$, sendo H definida positiva.

Outro critério de parada é definido para obter uma solução quase ótima. Computando $t = (\tau_1, \tau_2)$ e $\tilde{s}$ como no Lema 6, se as condições do Teorema 3 são satisfeitas, então $\tilde{s}$ é uma solução quase ótima do subproblema (1.2) e então o algoritmo pára com $\lambda^* = \tilde{\lambda}$ e $s^* = \tilde{s}$.

O último critério de parada do algoritmo é definido para detectar que o intervalo de salvaguarda $[\alpha_L, \alpha_U]$ é muito pequeno e não temos ainda uma aproximação para uma solução na fronteira:

$$|\alpha_U - \alpha_L| \leq \varepsilon_\alpha \max\{|\alpha_L|, |\alpha_U|\} \quad (3.6)$$

com $\varepsilon_\alpha \in (0, 1)$.

Neste caso temos o *hard case* com α^* próximo de $\tilde{\alpha}_1$. Então o algoritmo termina com $\lambda^* = \lambda_1(\alpha_k)$ com α^* satisfazendo $|\alpha^* - \tilde{\alpha}_1| \leq \varepsilon_\alpha$.

Se ν_1 é suficientemente grande, podemos tomar $p = \frac{u_1}{\nu_1}$. Como $\|p\| < \Delta$ neste caso, uma aproximação para uma solução é dada por $s^* = p + \tau\hat{z}$ tal que $\|s^*\| = \Delta$, onde o vetor $\hat{z}$ é uma aproximação para o autovetor associado com o menor autovalor de H .

Para a escolha de τ temos duas possibilidades e então escolhemos a de menor magnitude, que minimizará $\psi(p + \tau\hat{z})$ como visto no algoritmo anterior, na expressão dada por (2.2).

3.5 Algoritmo de Rojas, Santos e Sorensen

Este algoritmo foi apresentado e implementado em `Matlab` em [38].

A seguir temos a esquematização de duas subrotinas utilizadas no algoritmo principal de RSS.

Ajuste de α

Dados $\varepsilon_\nu, \varepsilon_\alpha \in (0, 1)$ e $\alpha_L, \alpha_U, \alpha$ com $\alpha \in [\alpha_L, \alpha_U]$

Se $k > 0$, computar os autopares $\left\{ \lambda_1(\alpha), (\nu_1, u_1^T)^T \right\}$ e $\left\{ \lambda_i(\alpha), (\nu_i, u_i^T)^T \right\}$ correspondentes ao menor autovalor e um autovalor próximo do segundo menor autovalor de B_α

Enquanto $\|g\|\|\nu_1\| \leq \varepsilon_\nu\sqrt{1 - \nu_1^2}$, $\|g\|\|\nu_i\| \leq \varepsilon_\nu\sqrt{1 - \nu_i^2}$ e $|\alpha_U - \alpha_L| > \varepsilon_\alpha \max\{|\alpha_L|, |\alpha_U|\}$

- $\alpha_U = \alpha$
- $\alpha = (\alpha_L + \alpha_U)/2$

- Computar $\left\{ \lambda_1(\alpha), (\nu_1, u_1^T)^T \right\}$ e $\left\{ \lambda_i(\alpha), (\nu_i, u_i^T)^T \right\}$

Salvaguarda para α

Dados α , $\delta_U \geq \delta_1$, α_L , α_U , $\phi_i = -g^T s_i$ e $\phi'_i = \|s_i\|^2$ para $i = k-1, k$
 Se $\alpha \notin [\alpha_L, \alpha_U]$

- Se $k = 0$ então $\alpha = \delta_U + \phi_k + \phi'_k(\delta_U - \lambda_k)$
- Caso contrário
 - Se $\|s_k\| < \|s_{k-1}\|$ então $\alpha = \delta_U + \phi_k + \phi'_k(\delta_U - \lambda_k)$
 - Caso contrário $\alpha = \delta_U + \phi_{k-1} + \phi'_{k-1}(\delta_U - \lambda_{k-1})$
- Se $\alpha \notin [\alpha_L, \alpha_U]$ então $\alpha = (\alpha_L + \alpha_U)/2$

Com estas subrotinas e os elementos abordados ao longo deste capítulo temos o algoritmo RSS descrito a seguir.

Algoritmo de Rojas, Santos e Sorensen

Dados H , g e Δ e as tolerâncias referentes aos critérios de parada.

- Computar δ_L e δ_U tais que $\delta_L \leq \delta_1 \leq \delta_U$ e o intervalo de salvaguarda inicial $[\alpha_L, \alpha_U]$ como apresentado na seção 3.4.1
- Computar α_0 e α_1 como em (3.1)
- Tomar $k = 0$
- Enquanto os critérios de parada não forem satisfeitos
 1. Ajustar α
 2. Atualizar $\delta_U = \min \left\{ \delta_U, \frac{u_1^T H u_1}{u_1^T u_1} \right\}$
 3. Se $\|g\| |\nu_1| > \varepsilon_\nu \sqrt{1 - \nu_1^2}$ então tomar $\lambda_k = \lambda_1(\alpha_k)$ e $s_k = u_1/\nu_1$
 - Se $\|s_k\| < \Delta$ então $\alpha_L = \alpha_k$

- Se $\|s_k\| > \Delta$
 - * então $\alpha_U = \alpha_k$
 - * caso contrário $\lambda_k = \lambda_i(\alpha_k)$, $s_k = u_i/\nu_i$ e $\alpha_U = \alpha_k$
- 4. Computar α_{k+1} pelos esquemas de interpolação apresentados em 3.3
- 5. Verificar salvaguarda para α
- 6. Tomar $k = k + 1$

3.6 Convergência do método na resolução dos subproblemas

Para este algoritmo não foram apresentados em [37] resultados de convergência no contexto de métodos de região de confiança para a resolução do problema geral de minimização irrestrita, mas que seriam similares aos de Moré e Sorensen desde que fossem satisfeitas as condições para as soluções aproximadas dos subproblemas.

Porém, considerando isoladamente a resolução dos subproblemas (1.2) utilizando o algoritmo proposto por Rojas, Santos e Sorensen temos um resultado de convergência local formalizado pelo teorema a seguir, cuja demonstração é dada em [37].

Teorema 5 Seja $\lambda^* \leq \delta_1$ o multiplicador de Lagrange correspondente à solução na fronteira do problema (1.2), com $g \neq 0$. Sejam $\{\lambda_k\}$ e $\{s_k\}$ as sequências geradas pelo algoritmo de Rojas, Santos e Sorensen usando o esquema de interpolação de dois pontos dado por (3.2) para atualizar α . Existe uma vizinhança $\mathcal{B}$ de λ^* tal que se $\lambda_{i-1}, \lambda_i \in \mathcal{B}$, então para $k \geq i - 1$

- i. $\{\lambda_k\}$ permanece em $\mathcal{B}$ e converge com taxa q -superlinear para λ^* .
- ii. $\{s_k\}$ converge com taxa q -superlinear para $y = -(H - \lambda^* I)^\dagger g$.

Além deste resultado, também é provado em [37] que o algoritmo é globalmente convergente, ou seja, independentemente do ponto inicial dado, o algoritmo convergirá.

Capítulo 4

Problema de empacotamento de cilindros

Um importante problema na vida real é o problema de empacotamento de cilindros. Muitas vezes deseja-se empacotar vários cilindros em embalagens retangulares. Para a indústria é desejável, por razões econômicas, que o maior número possível de cilindros sejam embalados juntos, diminuindo assim o custo.

Trabalhando com cilindros de mesmo raio, o nosso problema consiste em, dados kc cilindros de raio r e embalagens de dimensão fixa, verificar se é possível empacotar os cilindros em uma mesma embalagem e, se for possível, qual a posição dos cilindros na embalagem.

Este problema pode ser modelado como um problema bidimensional, de colocar kc círculos de raio r em um retângulo de dimensões $d_1 \times d_2$.

Denotando por $p^1, \dots, p^{kc}$ as posições dos centros dos círculos onde $p^j = (p_1^j, p_2^j)$ e o retângulo $R = \{p \in \mathbb{R}^2 | 0 \leq p_1 \leq d_1, 0 \leq p_2 \leq d_2\}$, para que os círculos estejam no interior do retângulo devemos ter as componentes dos centros $r \leq p_1 \leq d_1 - r$ e $r \leq p_2 \leq d_2 - r$, como ilustra a Figura 8.

Outro fato importante para o problema ter solução é que, além dos círculos estarem no interior do retângulo, o que é equivalente aos cilindros estarem dentro da embalagem retangular, é preciso que os círculos não se sobreponham.

Como a distância entre dois círculos C_i e C_j é dada por $\|p^i - p^j\| - 2r$, se $\|p^i - p^j\| - 2r \geq 0$ (ou $2r - \|p^i - p^j\| \leq 0$) então os círculos não estão sobrepostos. Logo, se não há sobreposição de C_i e C_j , $\max\{(2r)^2 - \|p^i - p^j\|^2, 0\} = 0$.

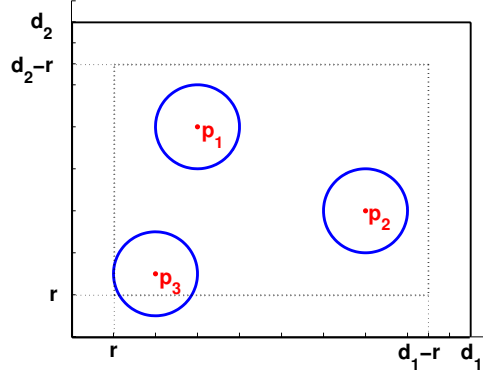


Figura 8 - Ilustração de um problema de empacotamento com apenas três cilindros em uma caixa suficientemente grande

Assim modelamos o problema como um problema de otimização da seguinte forma:

$$\begin{aligned} \min \quad & \sum_{i=1}^{kc-1} \sum_{j=i+1}^{kc} \max \left\{ 0, (2r)^2 - \|p^i - p^j\|^2 \right\}^2 \\ \text{s. a} \quad & r \leq p_1^i \leq d_1 - r, \quad r \leq p_2^i \leq d_2 - r, \quad i = 1, \dots, kc \end{aligned}$$

Se, ao resolver este problema, obtivermos como solução um vetor $(p_1^1, \dots, p_2^{kc})$ tal que o valor da função objetivo seja zero neste ponto, então o problema original tem solução. Se o valor da função objetivo em $(p_1^1, \dots, p_2^{kc})$ for maior que zero, então ocorre sobreposição de pelo menos um par de círculos, o que indica que não é possível colocar kc cilindros de raio r na embalagem, ou que encontramos uma solução local do problema.

Nosso objetivo então é resolver este problema de otimização. Entretanto, para resolvê-lo com os algoritmos de região de confiança descritos nas seções anteriores reescrevemos o problema de otimização transformando-o em um problema irrestrito, incorporando as restrições na nova função objetivo:

$$\min f(p_1^1, \dots, p_2^{kc})$$

onde

$$\begin{aligned} f(p_1^1, \dots, p_2^{kc}) = & \sum_{i=1}^{kc-1} \sum_{j=i+1}^{kc} \max \left\{ 0, (2r)^2 - \|p^i - p^j\|^2 \right\}^2 + \rho \left(\sum_{i=1}^{kc} \left[\max \left\{ 0, r - p_1^i \right\}^2 \right. \right. \\ & \left. \left. + \max \left\{ 0, r - p_2^i \right\}^2 + \max \left\{ 0, p_1^i - d_1 + r \right\}^2 + \max \left\{ 0, p_2^i - d_2 + r \right\}^2 \right] \right). \end{aligned}$$

Utilizamos o parâmetro $\rho \in \mathbb{R}_+$ para ponderar a função objetivo original e o termo correspondente às restrições. O valor deste parâmetro inicialmente está livre para ser

calibrado nos experimentos numéricos, podendo ajudar no desempenho dos algoritmos.

Para a utilização dos algoritmos são necessários também os cálculos das derivadas de primeira e segunda ordem. Para esta formulação as derivadas de primeira ordem são dadas por:

$$\frac{\partial f}{\partial p_1^\ell} = \sum_{j=1}^N -4[(2r)^2 - \|p^\ell - p^{i_j}\|^2](p_1^\ell - p_1^{i_j}) + \rho \begin{cases} 0, & \text{se } r \leq p_1^\ell \leq d_1 - r \\ -2(r - p_1^\ell), & \text{se } p_1^\ell < r \\ 2(p_1^\ell - d_1 + r), & \text{se } p_1^\ell > d_1 - r \end{cases}$$

$$\frac{\partial f}{\partial p_2^\ell} = \sum_{j=1}^N -4[(2r)^2 - \|p^\ell - p^{i_j}\|^2](p_2^\ell - p_2^{i_j}) + \rho \begin{cases} 0, & \text{se } r \leq p_2^\ell \leq d_2 - r \\ -2(r - p_2^\ell), & \text{se } p_2^\ell < r \\ 2(p_2^\ell - d_2 + r), & \text{se } p_2^\ell > d_2 - r \end{cases}$$

onde $i_j = i_1, \dots, i_N$ são tais que $\|p^\ell - p^{i_j}\| < 2r$.

Para as derivadas de segunda ordem temos as seguintes relações:

$$\frac{\partial^2 f}{\partial (p_1^\ell)^2} = \sum_{j=1}^N \{8(p_1^\ell - p_1^{i_j})^2 - 4[(2r)^2 - \|p^\ell - p^{i_j}\|^2]\} + \rho \begin{cases} 0, & \text{se } r < p_1^\ell < d_1 - r \\ 2, & \text{se } p_1^\ell < r \text{ ou } p_1^\ell > d_1 - r \end{cases}$$

$$\frac{\partial^2 f}{\partial (p_2^\ell)^2} = \sum_{j=1}^N \{8(p_2^\ell - p_2^{i_j})^2 - 4[(2r)^2 - \|p^\ell - p^{i_j}\|^2]\} + \rho \begin{cases} 0, & \text{se } r < p_2^\ell < d_2 - r \\ 2, & \text{se } p_2^\ell < r \text{ ou } p_2^\ell > d_2 - r \end{cases}$$

$$\frac{\partial^2 f}{\partial p_1^\ell \partial p_1^\ell} = \begin{cases} 0, & \text{se } \|p^\ell - p^m\| > 2r \\ -8(p_1^\ell - p_1^m)^2 + 4[(2r)^2 - \|p^\ell - p^m\|^2], & \text{se } \|p^\ell - p^m\| < 2r \end{cases}$$

$$\frac{\partial^2 f}{\partial p_2^\ell \partial p_2^\ell} = \begin{cases} 0, & \text{se } \|p^\ell - p^m\| > 2r \\ -8(p_2^\ell - p_2^m)^2 + 4[(2r)^2 - \|p^\ell - p^m\|^2], & \text{se } \|p^\ell - p^m\| < 2r \end{cases}$$

$$\frac{\partial^2 f}{\partial p_2^\ell \partial p_1^\ell} = \frac{\partial^2 f}{\partial p_1^\ell \partial p_2^\ell} = \begin{cases} 0, & \text{se } \|p^\ell - p^m\| > 2r \\ -8(p_1^\ell - p_1^m)(p_2^\ell - p_2^m), & \text{se } \|p^\ell - p^m\| < 2r \end{cases}, \text{ com } \ell \neq m$$

$$\frac{\partial^2 f}{\partial p_2^\ell \partial p_1^\ell} = \frac{\partial^2 f}{\partial p_1^\ell \partial p_2^\ell} = \sum_{j=1}^n 8(p_1^\ell - p_1^{i_j})(p_2^\ell - p_2^{i_j}), \text{ com } i_j \text{ tal que } \|p^\ell - p^{i_j}\| < 2r.$$

Uma característica importante desta função é que, na solução, ou seja, quando os cilindros não estão sobrepostos ($\|p^\ell - p^m\| \geq 2r$) e as restrições não são violadas ($0 \leq p_1^\ell \leq d_1 - r$ e $0 \leq p_2^\ell \leq d_2 - r$), a matriz Hessiana é nula.

Este fato pode dificultar o desempenho dos algoritmos, fazendo com que a matriz Hessiana seja singular em algumas iterações do algoritmo, principalmente quando nos aproximamos da solução do problema.

Outra característica deste problema é a multiplicidade de soluções, quando há pelo menos uma. Isto é causado pelo fato dos cilindros serem iguais, ou seja, trocando dois cilindros de lugar, temos a mesma solução na prática, mas são soluções distintas para o tratamento matemático dado ao problema.

4.1 Armazenamento da matriz Hessiana e programação

Um aspecto importante para ser explorado nesta classe de problemas é a esparsidade da matriz Hessiana. Apesar da grande quantidade de elementos nulos da matriz Hessiana na maioria das iterações, não há uma estrutura de esparsidade pré-definida. A estrutura desta matriz está fortemente relacionada com a ordem das coordenadas dos centros dos cilindros que aparece na função objetivo e com a quantidade de sobreposições de cilindros em uma determinada iteração.

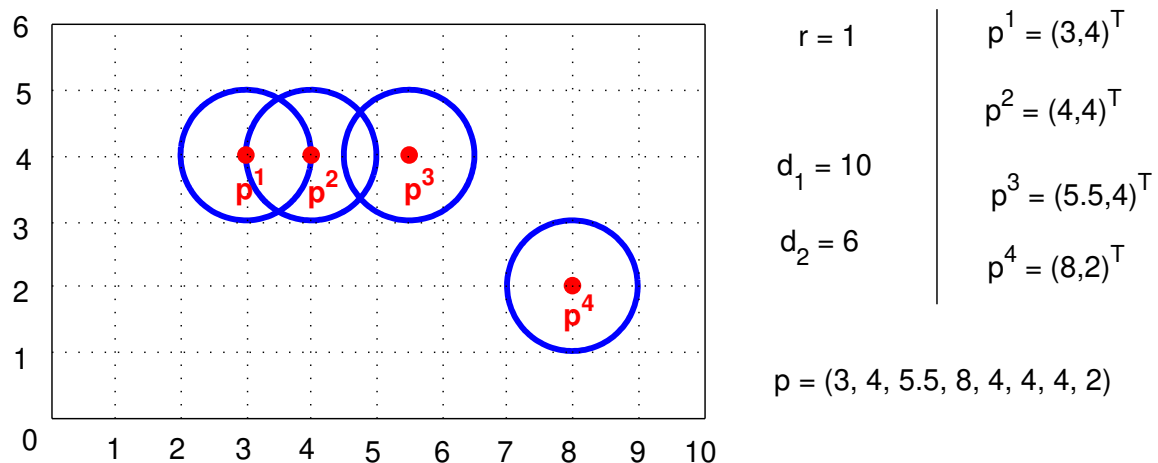
Uma propriedade intrínseca deste problema, que influencia a sua programação, é a contribuição nula dos termos correspondentes a cilindros que não se sobrepõe ($\|p^\ell - p^m\| > 2r$) nos cálculos tanto da função objetivo, como do vetor gradiente e também da matriz Hessiana.

Outro fato importante é que, a cada iteração interna do algoritmo de região de confiança, não há alteração do ponto. Portanto, para resolver os subproblemas através dos algoritmos de Moré e Sorensen ou Rojas, Santos e Sorensen, precisamos calcular a matriz Hessiana e o vetor gradiente apenas uma vez.

Assim, para a programação em `Matlab`, optamos pela criação de uma estrutura a qual denotaremos por m . A cada cilindro do problema está associado um elemento de m , composto por uma matriz com duas colunas, e número de linhas igual ao número de cilindros que o sobrepõe, sendo nula caso não haja sobreposição. Cada matriz da estrutura m contém na primeira coluna o índice dos cilindros que sobrepõem o dado cilindro e, na segunda coluna, as respectivas distâncias entre seus centros.

Para facilitar a compreensão desta estrutura, apresentamos um exemplo ilustrado na Figura 9.

Como podemos observar, esta estrutura é composta por um conjunto de matrizes de dimensões variadas, sempre com duas colunas ou vazia.



ℓ	i_ℓ	$\ p^\ell - p^{i_\ell}\ $
$\downarrow$	$\downarrow$	$\downarrow$
$m\{1\}$	$=$	$(2, \quad 1)$
$m\{2\}$	$=$	$\begin{pmatrix} 1, & 1 \\ 3, & 1.5 \end{pmatrix}$
$m\{3\}$	$=$	$(2, \quad 1.5)$
$m\{4\}$	$=$	$[\quad]$

Figura 9 - Exemplo para a construção da estrutura m

A cada iteração do algoritmo de região de confiança, a função que calcula esta estrutura é chamada, tendo como dado de entrada o ponto $(p_1^1, \dots, p_1^{kc}, p_2^1, \dots, p_2^{kc})$ onde as kc primeiras coordenadas correspondem às primeiras coordenadas dos centros dos círculos, seguidas das segundas coordenadas.

Assim, ao computar a função objetivo, o vetor gradiente ou a matriz Hessiana, devemos apenas nos preocupar com os índices que aparecem na primeira coluna das estruturas m . São estes índices que contribuirão com a parte não nula nos somatórios que aparecem na expressões da função, do gradiente e da Hessiana.

Outro fato importante a respeito da programação é o cálculo e o armazenamento da matriz Hessiana.

Para o cálculo da matriz temos quatro partes distintas, como observamos abaixo.

$$H = \begin{bmatrix} \frac{\partial^2 f}{(\partial p_1^1)^2} & \frac{\partial^2 f}{\partial p_1^1 \partial p_1^2} & \cdots & \frac{\partial^2 f}{\partial p_1^1 \partial p_1^{kc}} & \frac{\partial^2 f}{\partial p_1^1 \partial p_2^1} & \cdots & \frac{\partial^2 f}{\partial p_1^1 \partial p_2^{kc}} \\ & & \ddots & \vdots & \vdots & & \vdots \\ & & & \frac{\partial^2 f}{\partial p_1^{kc-1} \partial p_1^{kc}} & & & \\ & & & \frac{\partial^2 f}{(\partial p_1^{kc})^2} & \frac{\partial^2 f}{\partial p_1^{kc} \partial p_2^1} & \cdots & \frac{\partial^2 f}{\partial p_1^{kc} \partial p_2^{kc}} \\ & & & & \frac{\partial^2 f}{(\partial p_2^1)^2} & \frac{\partial^2 f}{\partial p_2^1 \partial p_2^2} & \cdots & \frac{\partial^2 f}{\partial p_2^1 \partial p_2^{kc}} \\ & & & & & \ddots & \vdots \\ & & & & & & \frac{\partial^2 f}{\partial p_2^{kc-1} \partial p_2^{kc}} \\ & & & & & & \frac{\partial^2 f}{(\partial p_2^{kc})^2} \end{bmatrix}$$

Estas partes são: a diagonal, derivadas mistas com respeito apenas às primeiras coordenadas dos centros dos cilindros, derivadas mistas envolvendo as primeiras e segundas coordenadas e as derivadas mistas envolvendo apenas as segundas coordenadas.

Como esta matriz é simétrica, mostramos no esquema apenas a parte triangular superior, que será utilizada para o nosso esquema de armazenamento.

Para isso utilizamos apenas vetores, armazenando somente os elementos não nulos da parte triangular superior da matriz.

Para exemplificar este armazenamento temos um caso de dimensão pequena:

$$H = \begin{bmatrix} 1 & 2 & 0 & 1 & 0 & 0 \\ 2 & 2 & 1 & 0 & 0 & 2 \\ 0 & 1 & 3 & 2 & 3 & 0 \\ 1 & 0 & 2 & 1 & 1 & 4 \\ 0 & 0 & 3 & 1 & 2 & 1 \\ 0 & 2 & 0 & 4 & 1 & 5 \end{bmatrix} \quad \begin{aligned} d &= [1, 2, 3, 1, 2, 5] \\ h &= [2, 1, 1, 2, 3, 1, 2, 4, 1] \\ linh &= [1, 2, 1, 3, 3, 4, 2, 3, 4] \\ ncol &= [1, 1, 2, 2, 3] \\ nzeros &= [0, 1, 0, 2, 1] \end{aligned}$$

Utilizamos um vetor d para armazenar os elementos da diagonal. O vetor h armazena os elementos não nulos fora da diagonal, percorrendo a matriz por colunas, da menor para a maior linha. O vetor $linh$ armazena o índice das linhas correspondentes aos elementos de h . Os vetores $ncol$ e $nzeros$ armazenam, respectivamente, o número de elementos não

nulos e o número de zeros antes do primeiro elemento não nulo de cada coluna, a partir da segunda coluna. Estes vetores facilitam os cálculos que envolvem a matriz Hessiana nos dois algoritmos analisados.

4.2 Decomposição de Cholesky para uma matriz esparsa

Para o algoritmo de Rojas, Santos e Sorensen, considerando somente as operações que envolvem a matriz Hessiana, é necessária apenas a programação de uma rotina que computa o produto da matriz por um vetor, que fizemos aproveitando a estrutura esparsa da matriz e seu armazenamento como descrito anteriormente.

Já para o algoritmo de Moré e Sorensen, após a realização da decomposição de Cholesky, são necessárias rotinas que resolvem sistemas triangulares superiores e inferiores. Estas rotinas também foram implementadas em `Matlab` levando em consideração o armazenamento da matriz Hessiana.

Com o objetivo de aproveitar a estrutura e o armazenamento da matriz durante a decomposição para efetuar as atualizações dos limitantes do algoritmo de Moré e Sorensen, propusemos uma pequena modificação no modo de calcular o fator de Cholesky usual.

Em [9] foi proposta uma rotina para computar a decomposição LDL^T de uma matriz esparsa e simétrica A , onde D é uma matriz diagonal e L uma matriz triangular inferior.

A principal diferença entre esta implementação e a tradicional é que, neste caso, o fator L é computado por linhas, em contraste com o método convencional, em que a decomposição é computada por colunas. Este fato privilegia o uso deste tipo de rotina para a decomposição de matrizes esparsas em combinação com o algoritmo de Moré e Sorensen, no qual a atualização dos limitantes para λ é feita com base na submatriz que corresponde à porção bem sucedida da fatoração de Cholesky, no caso em que essa fatoração falha.

Neste trabalho adaptamos as idéias apresentadas em [9] para a decomposição da matriz A na forma GG^T , onde G é uma matriz triangular inferior.

Na decomposição de Cholesky tradicional, em uma determinada iteração k , temos computados os elementos das primeiras $k - 1$ colunas da matriz triangular inferior G . Na nossa rotina, teremos as $k - 1$ primeiras linhas, ou seja, teremos a submatriz triangular de G de dimensão $k - 1$.

Supondo então que estamos na k -ésima iteração da decomposição, o próximo passo será computar a k -ésima linha de G .

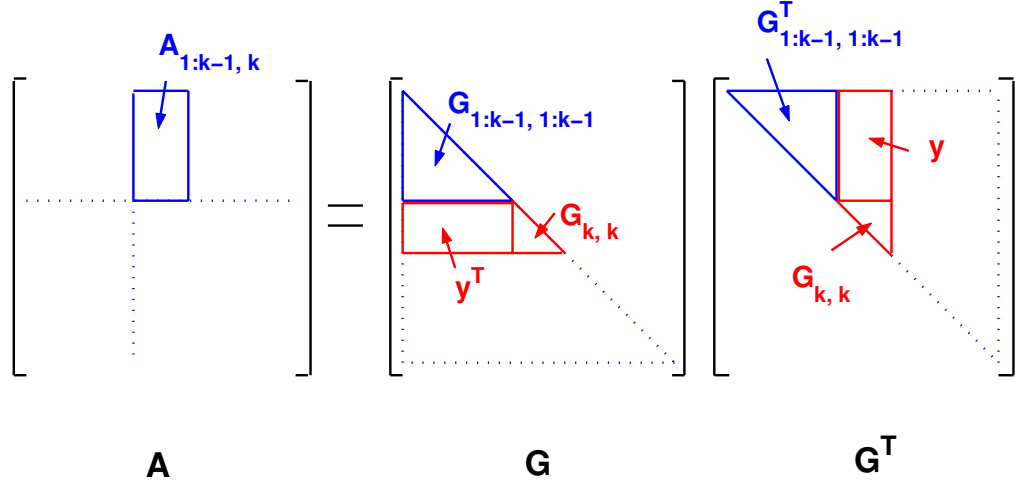


Figura 10 - Esquema da k -ésima iteração de uma decomposição de Cholesky com os elementos computados por linha.

Denotando por y o vetor formado pelos $k - 1$ primeiros elementos da k -ésima linha da matriz G , ou seja, $y^T = G_{k,1:k-1}$, temos que

$$G_{1:k-1,1:k-1}y = A_{1:k-1,k} \quad (4.1)$$

como exemplifica a Figura 10.

Como a submatriz $G_{1:k-1,1:k-1}$ já é conhecida nesta etapa do processo, obtemos y resolvendo o sistema triangular (4.1).

Com y calculado, podemos então computar $G_{k,k}$ através da relação:

$$y^T y + G_{k,k}^2 = A_{k,k} \Rightarrow G_{k,k} = \sqrt{A_{k,k} - y^T y}$$

Assim como na decomposição usual de Cholesky, temos o resultado que $A_{k,k} - y^T y > 0$ sempre que A é definida positiva. No caso da matriz não ser definida positiva durante a resolução dos subproblemas, esta rotina de decomposição é interrompida e o valor $A_{k,k} - y^T y < 0$ será um dado de saída, utilizado para atualizar os limitantes no algoritmo de Moré e Sorensen, como descrito anteriormente.

O armazenamento de G é feito de forma semelhante ao armazenamento da matriz Hessiana, através de vetores que armazenam os elementos da diagonal e os elementos não nulos.

A estrutura de esparsidade da matriz Hessiana não é totalmente preservada, mas é repercutida para a matriz G . Ao decompor a matriz $A = GG^T$, com $A = H - \lambda I$, apenas

temos a garantia de que os elementos nulos que aparecem antes do primeiro elemento não nulo de cada coluna de H serão preservados na matriz G .

Logo, o vetor $nzeros$ do armazenamento de H será o mesmo para G , mas agora indicando o número de zeros que aparecem em cada linha de G antes do primeiro elemento não nulo.

Como para o problema de empacotamento de cilindros não temos uma estrutura de esparsidade constante da matriz, poderíamos, em uma dada iteração, ter uma matriz com a primeira linha com elementos não nulos. Isso poderia provocar o surgimento do fator G totalmente denso. Esta é uma dificuldade inerente ao uso da decomposição de Cholesky neste tipo de problema, mas que poderia ser evitada com um pré-processamento do ponto antes do cálculo da matriz Hessiana.

Como a ordem dos cilindros é escolhida de forma arbitrária, se trocássemos a ordem em que eles aparecem na chamada da função objetivo e das derivadas, alteraríamos a estrutura da matriz H . Porém, a escolha desta ordem também provocaria um aumento de custo computacional para a resolução do problema. Neste trabalho, não realizamos este pré-processamento.

Capítulo 5

Experimentos computacionais

Nosso principal objetivo neste trabalho é a comparação do desempenho dos métodos de Moré e Sorensen e de Rojas, Santos e Sorensen na resolução dos subproblemas no contexto de métodos de região de confiança para problemas de empacotamento de cilindros.

Para isso implementamos totalmente o método MS em `Matlab` para compararmos com o desempenho da implementação do método RSS, já disponível em `Matlab` (versão 1.2 de agosto de 2006).

Os testes foram realizados em um computador com processador de 2793 Mhz e memória de 512 Mb. A versão do `Matlab` utilizada foi 6.5.

Para o método MS utilizamos 4 critérios de parada. O primeiro ocorre quando o número máximo de iterações é atingido. No nosso caso utilizamos o número máximo de iterações igual à dimensão $2kc$, onde kc é o número de cilindros do problema.

Outros dois critérios de parada utilizados indicam a convergência do método sem a detecção do *hard case*. Neste caso, ou a solução é encontrada na fronteira com $|\delta - \|p\|| \leq r_{tol}\delta$, ou a solução é encontrada no interior da região de confiança com $\|p\| \leq (1 + r_{tol})\delta$ e $|\lambda| \approx 0$. Para estes critérios de parada relativos utilizamos $r_{tol} = 10^{-3}$.

O último critério de parada detecta o *hard case* como descrito em 2.3.4 e então o ponto $p + \tau\hat{z}$ é a solução retornada pelo algoritmo.

Para os critérios de parada do algoritmo RSS, utilizamos também o número máximo de iterações igual a $2kc$ e os parâmetros sugeridos como *default*, alterando apenas dois deles com a finalidade de compatibilizar a precisão das soluções encontradas com as do algoritmo MS. Foram eles `epsilon.Delta` e `epsilon.Int`, que passaram de 10^{-4} e 10^{-8} para 10^{-3} e 10^{-16} , respectivamente.

Para a resolução do problema principal pelo algoritmo RCMI utilizamos as constantes $\mu = 0.1$, $\eta = 1 - \mu$, $c_1 = 0.0625$, $c_2 = 0.5$, $c_3 = 2$, tomamos raio de região de confiança inicial $\Delta = 0.1\|g(p^0)\|$ e definimos 4 critérios de parada.

O primeiro critério é satisfeito quando o número máximo de iterações é atingido. Para

este número utilizamos o valor de duas vezes a dimensão do problema ($4kc$).

Quando há convergência na resolução do problema, o algoritmo pode parar com a norma do gradiente pequena ($\|g(p^k)\| \leq ep * \max\{\|g(p^0)\|, 1\}$, com $ep = 10^{-4}$) ou com o valor da função objetivo próximo de zero ($f(p^k) \leq 10^{-8}$).

O último critério de parada é dado pela estagnação do algoritmo com $\|p^k - p^{k-1}\| \leq 10^{-6}\|p^0\|$.

5.1 Testes iniciais

Antes de comparar o desempenho dos algoritmos de Moré e Sorensen e de Rojas, Santos e Sorensen na resolução dos problemas abordados em [5], comparamos os seus desempenhos para um problema de empacotamento de 15 cilindros em uma caixa quadrada de dimensões $d_1 \times d_2 = 10 \times 10$.

Como nosso principal objetivo é a comparação dos métodos e não propriamente desenvolver uma estratégia para obtenção da solução global do problema, não utilizamos o número máximo de cilindros que poderiam ser empacotados nesta caixa.

Nos primeiros testes realizados analisamos também a influência da escolha do parâmetro de penalidade ρ na resolução do problema.

Para isso fizemos 5 testes com os valores 1, 0.1, 5, 20 e 100 para ρ , o parâmetro que pondera a inviabilidade com relação à caixa original na função objetivo. Cada um destes testes foi resolvido 10 vezes, com diferentes escolhas para a configuração inicial dos cilindros.

O vetor p^0 de dimensão $2kc = 30$ com coordenadas correspondentes aos centros dos cilindros na configuração inicial, foi gerado aleatoriamente com as 15 primeiras componentes entre r e $d_1 - r$ e as 15 últimas entre r e $d_2 - r$. Assim, com a configuração inicial, as restrições da caixa não são violadas, mas alguns cilindros podem estar sobrepostos.

Para comparar o desempenho dos métodos analisamos os tempos de execução (t) e sua média ($\bar{t}$) para o conjunto de 10 testes para cada ρ , o valor da função no ponto final (f), o número de iterações externas ($IText$) e sua média ($\overline{IText}$) para cada 10 testes, a razão entre o número de iterações internas pelo número de iterações externas para cada problema e os critérios de parada satisfeitos.

Tabela 1: Testes com 15 cilindros, caixa $d = [10; 10]$
e p^0 gerado aleatoriamente 10 vezes

ρ	MS			RSS		
	$\overline{IText}$	$\bar{t}$	nsolv	$\overline{IText}$	$\bar{t}$	nsolv
0.1	38.6	0.73	1	42.6	30.37	7
1	57.5	1.47	0	21	14.18	9
5	39.1	1.06	4	18.1	12.17	9
20	35.4	1.21	8	16.3	10.30	10
100	35.9	1.12	6	15.7	11.70	9

Na Tabela 1 são mostrados a média do número de iterações por problema ($\overline{IText}$), a média do tempo de execução por problema ($\bar{t}$) e o número de problemas considerados resolvidos ($nsolv$) dos testes comparativos iniciais. Consideramos os problemas resolvidos se o algoritmo de região de confiança convergiu com a norma do gradiente no ponto final pequena, o valor da função objetivo final menor que 10^{-8} ou ainda se foi detectada a estagnação do método mas o valor da função objetivo alcançado foi menor que 10^{-5} .

Observando estes resultados iniciais, percebemos vantagens e desvantagens entre os dois algoritmos para resolver os subproblemas que aparecem no método de região de confiança.

Uma primeira observação importante é o fato do algoritmo RSS ser mais robusto que o MS para este conjunto de testes, no sentido de que para todos os valores de ρ , o algoritmo de região de confiança com RSS resolve mais problemas que o algoritmo com MS.

Porém, apesar da robustez do algoritmo RSS no contexto da resolução de problemas de empacotamento de cilindros, quando analisamos os tempos médio de execução dos algoritmos por problema, temos que o algoritmo MS resolve os subproblemas muito mais rápido que o RSS. Nesta tabela mostramos o tempo médio por problema, sem levar em consideração quantas iterações externas foram feitas para o cálculo desta média. Com esta medida temos que o algoritmo MS é cerca 10 vezes mais rápido do que de RSS por problema. Se levarmos em consideração o fato de MS ter realizado mais iterações externas por problema, temos uma rapidez ainda maior de MS quando comparados os dois algoritmos.

Aqui temos uma importante observação para fazer. Apesar do algoritmo MS utilizar fatorações de Cholesky, o que poderia aumentar seu tempo de execução com uma matriz muito densa, para estes problemas iniciais de dimensão 30, o tempo de execução se mostrou melhor do que o algoritmo de RSS que utiliza estratégias para obter aproximações de autovalores, talvez pelo fato do bom aproveitamento da esparsidade das matrizes ao efetuar os cálculos da fatoração de Cholesky.

Outro fato que merece destaque é o excelente desempenho do algoritmo RSS no contexto de resolver subproblemas num algoritmo de região de confiança, já que até então ele só havia sido testado ao resolver problemas isolados. O fato de seu tempo de execução ter sido maior pode ser amenizado com um ajuste mais refinado dos seus parâmetros internos para os critérios de parada, podendo ser menos exigente quando utilizado no contexto iterativo de região de confiança.

Analizamos também quais critérios de parada foram satisfeitos para os dois algoritmos nos 100 testes realizados (50 para cada algoritmo).

O algoritmo RSS parou na maioria das vezes com a norma do gradiente pequena. Quando o problema do empacotamento de cilindros não foi resolvido com a precisão desejada, o algoritmo de região de confiança atingiu o número máximo de iterações ou algum subproblema não pode ser resolvido pelo algoritmo RSS, retornando um vetor vazio, o que nos obrigou a parar a iteração externa, retornando o último iterando. Mas, apesar desta interrupção, o valor da função objetivo sempre foi menor do que 10^{-3} nestes casos.

Ao contrário do algoritmo de região de confiança com RSS, que nunca parou com a detecção de estagnação do processo, o algoritmo com MS parou 22 vezes com este critério de parada. Porém, em 12 destes casos, o valor da função objetivo já estava menor do que 10^{-3} . Isto indica uma dificuldade de mobilidade dos cilindros quando aplicado o algoritmo de MS.

Outro fato relevante é uma maior sensibilidade do método de MS com a variação do parâmetro ρ quando comparado ao método de RSS. Ao aumentar o valor de ρ estamos penalizando mais o ponto quando as restrições da caixa são violadas, sendo preferível sobrepor os cilindros em uma determinada iteração a ultrapassar os limites da caixa. Este foi um fato observado acompanhando-se os gráficos das configurações dos cilindros a cada iteração externa com o algoritmo de MS.

A Figura 11 mostra as configurações para um problema com $\rho = 1$, ou seja, sem penalizar mais as restrições. O primeiro gráfico mostra a configuração inicial gerada aleatoriamente. Após a primeira iteração (segundo gráfico da Figura 11) vemos um rápido afastamento dos cilindros, porém com a violação das restrições. Já na décima iteração temos todos os cilindros sem sobreposição e a partir deste momento, não há mais sobreposição em nenhuma iteração. Isso gera uma dificuldade para a convergência do método, com a pouca mobilidade dos cilindros para não sobrepor os demais, os cilindros que violam as restrições da caixa não conseguem espaço para se moverem. Neste caso, o método progrediu pouco a partir da décima iteração como é mostrado com o gráfico na iteração 60.

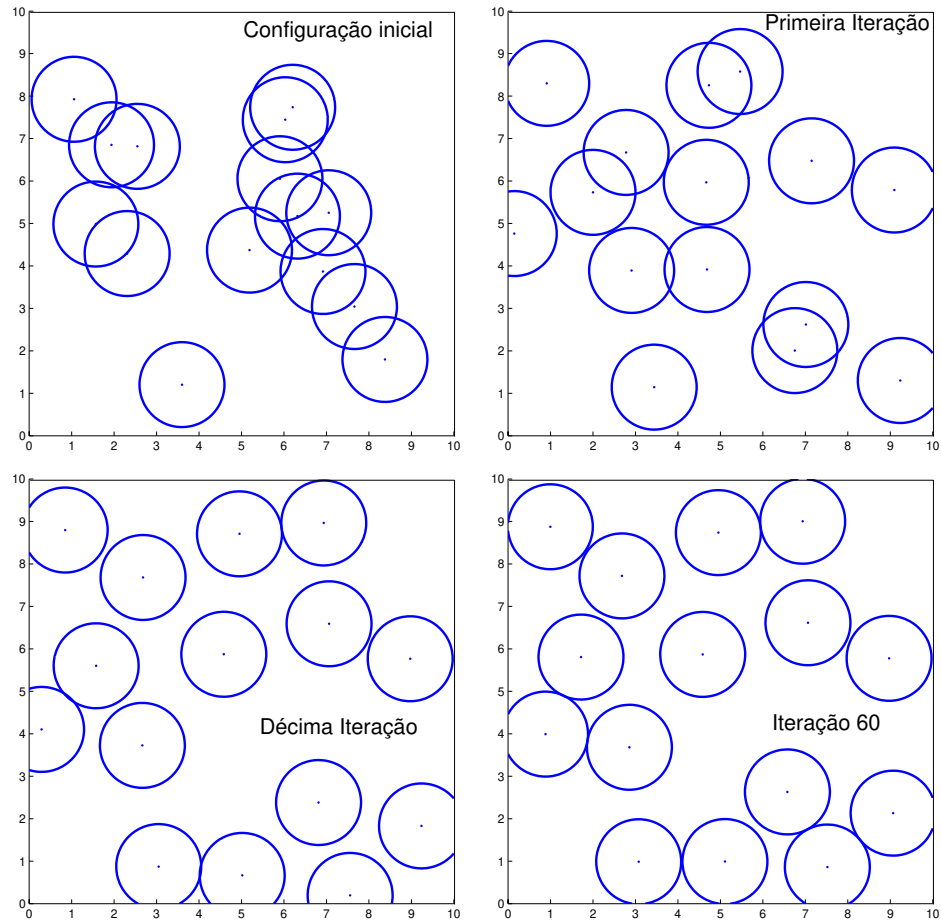


Figura 11 - Configurações dos cilindros na resolução de um teste com $\rho = 1$ pelo método de MS para o ponto inicial e para as iterações externas 1, 10 e 60 respectivamente.

Quando aumentamos o parâmetro ρ temos uma penalização maior para a violação das restrições da caixa.

Como vemos na Figura 12, ao contrário do que ocorreu com o teste com $\rho = 1$, com $\rho = 20$ os cilindros se sobrepõem mesmo que já estivessem separados, já que a penalização por sobreposição passa a ter um peso menor do que a penalização por violação da restrição.

No primeiro gráfico da Figura 12 temos o círculo em vermelho quase sem sobreposição com os demais na iteração 15. Na iteração seguinte já temos a sobreposição, dando mais espaço para o círculo que viola a restrição à esquerda se deslocar na próxima iteração. Com este comportamento, o método convergiu em 27 iterações e temos a configuração final mostrada no último gráfico da Figura 12.

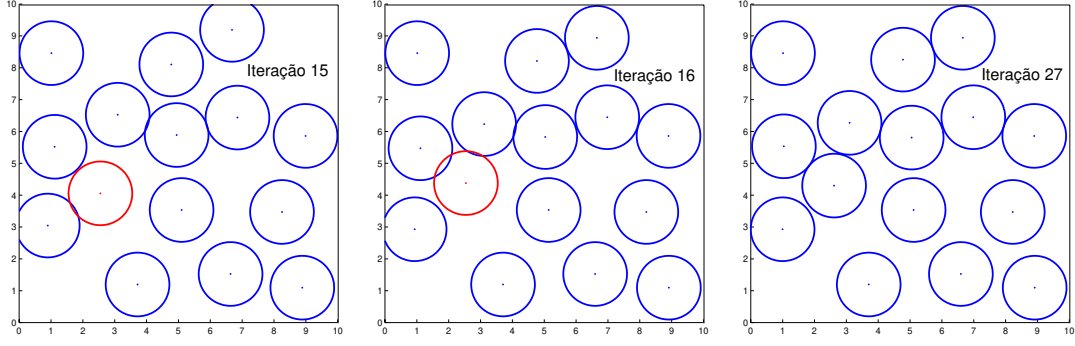


Figura 12 - Configurações dos cilindros na resolução de um teste com $\rho = 20$ pelo método de MS para as iterações externas 15, 16 e 27 respectivamente.

Outro aspecto analisado foi o desempenho interno: o número de iterações internas e os critérios de parada internos satisfeitos em os dois tipos de algoritmo. Para ilustrar a análise, mostramos os resultados para $\rho = 20$ na Tabela 2, onde $IText$ é o número de iterações externas, f é o valor da função objetivo no ponto final, $ITint/IText$ é a razão entre o número de iterações internas pelo número de iterações externas e t é o tempo de execução total.

Tabela 2: Teste com 15 cilindros, $d = [10; 10]$ e $\rho = 20$

	MS				RSS			
	$IText$	f	$\frac{ITint}{IText}$	t	$IText$	f	$\frac{ITint}{IText}$	t
teste 1	26	6.95e-10	5.61	0.75	12	2.29e-10	5.08	4.28
teste 2	32	2.53e-4	2.97	0.66	21	0.00	4.00	4.59
teste 3	30	2.46e-13	4.63	0.84	14	0.00	4.43	6.61
teste 4	23	1.42e-6	4.30	0.61	10	0.00	4.70	5.76
teste 5	49	3.53e-13	9.41	1.92	16	4.65e-10	4.69	14.06
teste 6	38	1.54e-12	2.97	0.78	18	6.10e-14	6.50	15.59
teste 7	41	4.53e-4	2.63	2.05	18	1.25e-13	4.78	7.67
teste 8	29	6.91e-8	5.59	0.83	12	4.90e-19	5.67	6.41
teste 9	52	0.00	7.19	2.34	21	1.25e-9	7.33	16.98
teste 10	34	3.30e-8	7.35	1.31	21	2.26e-8	6.05	21.02

Este teste ilustra bem o que aconteceu nos demais testes (com diferentes valores para ρ), ao compararmos o desempenho dos dois algoritmos quando ambos convergem para um ponto quase ótimo.

Na Tabela 2 podemos observar que de uma forma geral, o algoritmo de região de confiança com MS realiza mais iterações do que o algoritmo com RSS, o que pode ser observado em *IText*.

Apesar disso, o algoritmo com RSS consegue uma precisão melhor para a solução, com o valor da função objetivo mais próximo de zero, como pode ser visto pelos valores de f .

Analisando o número de iterações internas a cada iteração externa, temos resultados parecidos neste caso, com uma pequena vantagem para o algoritmo de MS.

A grande vantagem do algoritmo de Moré e Sorensen é o tempo de execução sempre menor que o do algoritmo de Rojas, Santos e Sorensen para este conjunto de testes. Como pode ser observado na Tabela 2, em alguns casos o algoritmo MS é até 20 vezes mais rápido do que o RSS, como ocorre no caso do teste 6.

Realizamos também 10 testes com $\rho = 20$ e dimensão da caixa $d = 20 \times 5$, com pontos iniciais gerados aleatoriamente. Para este problema com caixa retangular ainda obtivemos um melhor desempenho de RSS resolvendo 9 problemas contra apenas 5 de MS. Aqui admitimos que o algoritmo convergiu se o valor da função objetivo é menor do que 10^{-6} , mesmo que o número máximo de iterações tenha sido atingido.

A média de iterações externas por problema foi de 46.4 para MS e 25 para RSS. Quando comparados com os resultados dos problemas com caixa de dimensão 10×10 temos um aumento deste número para os dois métodos. Assim percebemos uma maior dificuldade na resolução dos problemas em caixas mais alongadas.

Uma observação importante é que poderíamos escolher pesos diferentes para penalizar a violação das restrições nas diferentes dimensões. Ao invés de adotar um único ρ , poderíamos trabalhar com ρ_{d1} e ρ_{d2} , o que poderia melhorar o desempenho dos métodos em caixas muito alongadas.

Embora não tenhamos efetuado testes para validar esta conjectura, nos pareceu interessante associar pesos distintos para a violação da caixa, favorecendo a movimentação dos cilindros ao longo da direção correspondente à maior dimensão.

5.2 Testes da literatura

A etapa seguinte dos experimentos numéricos consiste dos testes com os problemas de empacotamento de cilindros que aparecem em [5].

O nosso objetivo nestes experimentos é ligeiramente diferente do de Birgin, Martínez e Ronconi. Estes autores desejavam empacotar o máximo número de cilindros possível. Para isso o número de cilindros em uma determinada caixa aumentava gradativamente até que o problema não pudesse mais ser resolvido e o último número de cilindros era aceito como o número máximo que caberia naquela caixa.

De maneira diferente, nós não estamos preocupados com quantos cilindros cabem em determinada caixa. Nosso objetivo é comparar o desempenho do método de região de confiança com os subproblemas resolvidos pelo método de Moré e Sorensen e pelo método de Rojas, Santos e Sorensen.

Para isso utilizamos 14 problemas descritos em [5], sendo que em alguns deles utilizamos um número de cilindros menor do que o máximo obtido no artigo. Os critérios de parada são os mesmos utilizados nos testes anteriores, com exceção do critério de parada por estagnação do processo, que foi removido, e uma exigência maior para a precisão da norma do gradiente no ponto final, trocando o critério relativo ($\|g(p^k)\| \leq ep * \max\{\|g(p^0)\|, 1\}$) pelo critério absoluto ($\|g(p^k)\| \leq 10^{-8}$).

A primeira alteração do critério de parada foi devida a notarmos, após alguns testes iniciais, a estagnação do processo em poucas iterações para o algoritmo de MS. Com o objetivo de analisar o comportamento do método sem esta parada prematura, inibimos este critério de parada.

Já a mudança do critério de parada com a norma do gradiente foi feita para podermos analisar o comportamento dos métodos também com um maior número de iterações, o que contribui para verificarmos se há convergência para minimizadores locais do problema.

Para este conjunto de problemas também utilizamos uma estratégia diferente para a escolha do ponto inicial. Ao invés de tentarmos resolver cada problema mais de uma vez com diferentes escolhas aleatórias para o ponto inicial, utilizamos um ponto inicial ‘quente’ construído segundo a estratégia descrita a seguir.

O ponto inicial é construído tomando como o número de cilindros alinhados na menor dimensão, a parte inteira da divisão da menor dimensão da caixa pelo diâmetro dos cilindros ($2r$). Se esta divisão não é exata, o espaço excedente será utilizado para espaçar igualmente os cilindros ao longo desta direção. No exemplo da Figura 13, temos que esta divisão resultou em três cilindros em cada linha. O restante dos cilindros é igualmente dividido nas colunas formadas pelos primeiros cilindros. Se esta divisão não for exata, o espaço que sobra é utilizado para acomodar os cilindros excedentes, que mostramos em verde no exemplo da Figura 13.

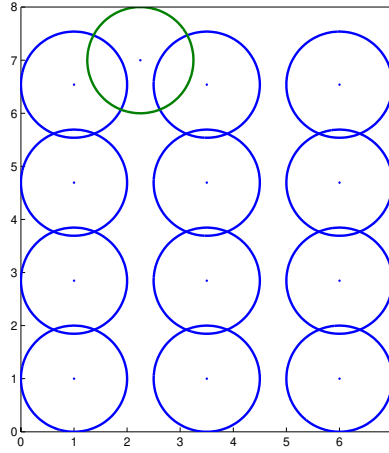


Figura 13 - Exemplo de um ponto inicial ‘quente’

Com este ponto inicial, temos os resultados dos testes exibidos na Tabela 3. Na tabela são mostrados a dimensão da caixa (d), o raio dos cilindros (r), o número de cilindros do problema (kc), o valor da função objetivo no ponto final (f), o número de iterações do método de região de confiança ($IText$) e o tempo de execução (t).

Tabela 3: Testes de [5] com o ponto inicial ‘quente’

	Problema			MS			RSS		
Teste	d	r	kc	f	$IText$	t	f	$IText$	t
1	[12;8]	1.02	20	0.3798	80	7.20	1.06e-10	40	54.91
2	[12;8]	1.01	20	1.6710	80	7.86	5.71e-7	45	83.44
3	[12;12]	2.1	6	5.5899	8	0.12	2.03e-10	18	2.84
4	[10;10]	1.8	6	9.1534	18	0.18	9.1534	4	1.14
5	[8;8]	1.4	6	1.9320	11	0.17	0.0070	24	5.26
6	[12;8]	1.7	7	0.7264	28	0.39	0.8030	28	6.81
7	[10;10]	1.3	13	1.7321	52	2.26	0.0242	52	100.08
8	[12;10]	1.4	14	1.1654	56	3.00	0.2132	56	87.31
9	[12;24]	2.1	15	27.5626	60	3.06	3.1438	60	68.52
10	[10;20]	1.8	15	75.6509	60	2.81	9.1607	60	121.69
11	[16;8]	1.4	15	3.5973	60	3.69	1.0807	60	119.91
12	[10;10]	0.9	28	0.3689	112	33.03	1.34e-14	33	36.42
13	[16;8]	1	30	15.6384	120	25.28	1.42e-8	22	43.86
14	[4.71;1.96]	0.14	120	4.78e-4	480	652.81	0.0013	94	2609.78

Uma observação importante é o reescalamento dos problemas apresentados no artigo. Para o teste 1, por exemplo, originalmente $d = [1200; 800]$ e $r = 102$. Esta decisão de reescalar o problema foi tomada levando em consideração os desempenhos dos métodos para alguns problemas testados com e sem reescalamento. Outro fator importante foi nossa decisão de fixar o parâmetro ρ para estes testes. Sabemos que este parâmetro está relacionado com as restrições da caixa e, portanto, é natural que o parâmetro deva ser alterado com o aumento da dimensão da caixa.

Analisando os testes iniciais para caixas de dimensão $d = [10; 10]$, decidimos fixar $\rho = 20$ para os testes com os problemas reescalados, já que para este valor de ρ obtivemos, de uma forma geral, os melhores resultados, como pode ser visto na Tabela 1.

Observando os resultados da Tabela 3, vemos que o método de Moré e Sorensen não se mostrou eficiente para este conjunto de problemas. Para a maioria dos problemas, o algoritmo de região de confiança com o subproblema abordado via MS parou com o número máximo de iterações externas atingido ($4kc$).

Já o algoritmo de Rojas, Santos e Sorensen convergiu para a solução desejada em 5 problemas e na maior parte dos demais conseguiu uma aproximação melhor para a solução do que o de Moré e Sorensen, apesar de executar o algoritmo em um tempo maior.

Um fato curioso ocorreu para os testes 3, 4 e 5 resolvidos com o algoritmo de Moré e Sorensen. Para estes testes, o algoritmo convergiu com a norma do gradiente no ponto final menor do que 10^{-8} , porém com o valor da função objetivo muito maior do que zero. Isso indica a convergência para um minimizador local do problema.

Para o teste 4, o algoritmo RSS também convergiu para um minimizador local. Porém para o teste 3, RSS encontrou um minimizador global.

Outra observação é o fato de termos escolhido como número máximo de iterações externas o dobro da dimensão do problema ($2(2kc)$). A fim de observar o progresso do método de MS com um número maior de iterações externas, aumentamos este valor, porém os resultados não foram melhores.

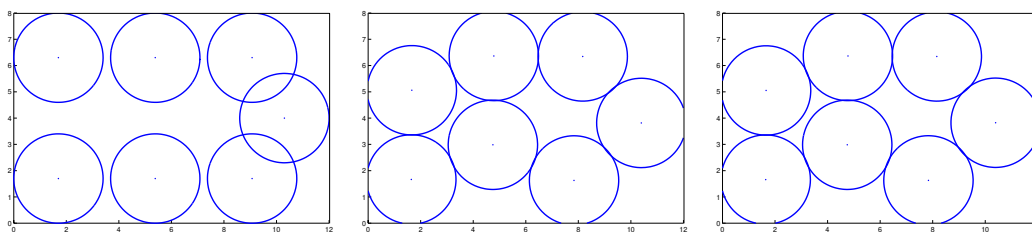


Figura 14 - Configurações dos cilindros na resolução do teste 6 pelo método de MS para o ponto inicial (esquerda) e para o ponto final com $4kc$ (meio) e $8kc$ (direita) iterações externas, respectivamente.

Como podemos observar na Figura 14, o método de MS não obtém progressos significativos em alguns casos. Nesta figura utilizamos o teste 6 da Tabela 3 com o aumento do número de iterações máximo de $4kc$ para $8kc$.

Um outro fato importante deste conjunto de testes é que a maioria foi testada com o número máximo de cilindros empacotados em [5]. Isso foi devido à nossa escolha para o ponto inicial ‘quente’, que em alguns casos não nos permitiu testar os algoritmos com um número menor de cilindros, já que o ponto inicial já seria a solução desejada.

Assim, este conjunto de testes é mais difícil para ser resolvido e era esperada a não convergência para alguns problemas.

Para testar alguns destes problemas com um número menor de cilindros dentro da caixa, utilizamos pontos iniciais com componentes geradas aleatoriamente, e tais que todos os cilindros estão inicialmente no interior da caixa.

Para isso rodamos 5 dos 14 problemas mostrados na Tabela 3, 5 vezes cada um. Os resultados estão mostrados na Tabela 4, onde $(\overline{IText})$ é a média do número de iterações por problema, $(\bar{t})$ é a média do tempo de execução por problema e $(nsolv)$ é o número de problemas considerados resolvidos.

Tabela 4: Testes de [5] com pontos iniciais aleatórios

Teste	Problema			MS			RSS		
	d	r	kc	$\overline{IText}$	$\bar{t}$	$nsolv$	$\overline{IText}$	$\bar{t}$	$nsolv$
1	[10;10]	1.8	5	18.4	0.22	3	15.40	1.76	4
2	[12;10]	1.4	12	48	1.45	0	25.20	22.51	5
3	[12;24]	2.1	14	56	2.55	0	47.40	76.42	3
4	[10;10]	0.9	25	96	17.78	2	43.60	160.25	5
5	[16;8]	1	28	112	36.89	0	47.20	188.33	5

Com a diminuição do número de cilindros, vemos uma melhora no desempenho do algoritmo de Rojas, Santos e Sorensen, resolvendo mais problemas e atingindo poucas vezes o número máximo de iterações, o que é repercutido na média do número de iterações externas por problema.

Consideramos como problemas resolvidos aqueles para os quais o valor da função objetivo no ponto final é menor do que 10^{-6} . Assim, em alguns casos, embora o método de Moré e Sorensen tenha atingido o número máximo de iterações, consideramos que o problema foi resolvido.

Para este conjunto de testes vemos novamente uma maior eficiência do algoritmo de região de confiança com a resolução dos subproblemas com o algoritmo de RSS, ainda que o custo de uma iteração RSS seja superior ao de uma iteração MS.

Considerações finais

Este trabalho consistiu essencialmente na comparação entre os algoritmos de Moré e Sorensen (MS) e de Rojas, Santos e Sorensen (RSS) para a solução aproximada dos subproblemas dos métodos de região de confiança para otimização irrestrita. Os dois algoritmos foram estudados em profundidade em seus aspectos teóricos, com o objetivo de colocar em perspectiva tanto as características comuns quanto os aspectos distintos destes dois algoritmos. Visando compatibilizar suas apresentações, o sinal do parâmetro ajustado pelo algoritmo MS foi trocado, o que gerou a necessidade de uma série de adaptações para as expressões do artigo original, e que estão registradas neste texto.

Foi desenvolvida a implementação em **Matlab** do algoritmo de Moré e Sorensen, especialmente planejada para a solução de problemas de empacotamento de cilindros. Nesse sentido, foi criada uma estrutura de dados conveniente para armazenar as informações próprias da função objetivo, vetor gradiente e matriz Hessiana. Também foi implementada uma rotina que efetua a fatoração de Cholesky por linhas de matrizes simétricas, armazenadas da maneira proposta neste trabalho. Nesta implementação, ainda que, pelo fato da matriz em questão não ser positiva definida, a fatoração de Cholesky não possa ser completada, as informações correntes são utilizadas para o ajuste dos parâmetros (limitantes) do algoritmo MS. A opção pelo ambiente de programação do **Matlab** deveu-se à disponibilidade do algoritmo RSS neste *software*. Assim, foi programado um algoritmo básico de região de confiança para otimização irrestrita, com as duas versões para a solução aproximada do subproblema quadrático.

Com relação aos experimentos numéricos realizados, as principais conclusões foram uma maior eficiência do algoritmo de Rojas, Santos e Sorensen quando comparado ao algoritmo de Moré e Sorensen, ao analisarmos o número de iterações externas do algoritmo de região de confiança e a quantidade de problemas resolvidos (convergência para o minimizador global) por conjunto de testes. Este melhor desempenho pode estar relacionado à natureza dos problemas e às formas diferentes com que os algoritmos utilizam informações da matriz Hessiana. O algoritmo MS utiliza decomposições de Cholesky enquanto o algoritmo RSS utiliza um processo iterativo que conta apenas com o cálculo de produtos matriz por vetor. A utilização de uma decomposição de Cholesky por linhas pode ter

influenciado o desempenho do algoritmo de Moré e Sorensen, no sentido que, quando a matriz não é definida positiva, o método calcula o novo parâmetro atualizando o intervalo de salvaguarda através de informações provenientes da decomposição de Cholesky.

Outro aspecto observado ao longo dos experimentos foi a ocorrência do *hard case* em ambos os métodos durante as resoluções dos subproblemas, porém sem grandes diferenças entre os métodos quanto ao número destas ocorrências.

Em termos do número médio de iterações internas, os dois algoritmos tiveram um desempenho semelhante. Quanto ao esforço global do algoritmo de região de confiança RCMI utilizando MS e RSS no passo 2, observamos que o tempo gasto por uma iteração RSS foi superior ao de uma iteração MS típica para os problemas analisados.

Em termos de trabalhos futuros, planejamos prosseguir com o desenvolvimento de algoritmos de restrições ativas de segunda ordem para otimização com restrições lineares, nos quais as estratégias de restrições ativas se combinam com métodos Newtonianos nas faces do politopo. Tendo em vista a solução de problemas de grande porte via estratégias livres de fatoração, a investigação da repercussão da precisão adotada para o problema de autovalores na qualidade da solução dos subproblemas obtidos pelo algoritmo RSS, e como esta qualidade influencia o desempenho global do método proposto é um aspecto que nos interessa.

Referências Bibliográficas

- [1] Página do ADIFOR: <http://www-unix.mcs.anl.gov/autodiff/ADIFOR/>
- [2] Página do NEOS-server: <http://www-neos.mcs.anl.gov>
- [3] Página do PACKMOL: <http://www.ime.unicamp.br/~martinez/packmol/index.html>
- [4] A. Björck, *Numerical methods for least squares problems*, Philadelphia: SIAM, 1996.
- [5] E. G. Birgin, J. M. Martínez & D. Ronconi. Optimizing the packing of cylinders into a rectangular container. A nonlinear approach, *European Journal on Operations Research* **160**, pp. 19–33, 2005.
- [6] A. K. Cline, C. B. Moler, G. W. Stewart & J. H. Wilkinson, An estimate for the condition number of a matrix, *Journal on Numerical Analysis* **16**, pp. 368–375, 1979.
- [7] A. R. Conn, N. I. M. Gould & Ph. L. Toint, *Trust-Region Methods*, MPS-SIAM series on optimization MP01, Philadelphia: SIAM, 2000.
- [8] Conway, J. H. & Sloane, N. J. A., *Sphere packings, lattices and groups*, with additional contributions by E. Bannai et al. - New York: Springer, 1993.
- [9] T. A. Davis, Algorithm 8xx: a concise sparse Cholesky factorization package, extraído de www.cise.ufl.edu/tech_reports/tr04/tr04-001.pdf, acessado em 25/07/2006.
- [10] J. E. Dennis Jr. & R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, Philadelphia, 1996.
- [11] M. A. Diniz-Ehrhardt, M. A. Gomes-Ruggiero & S. A. Santos, Comparing the numerical performance of two trust-region algorithms for large-scale bound-constrained minimization, *Investigación Operativa* **7**, pp. 23–54, 1997.

- [12] E. D. Dolan, R. Fourer, J. J. Moré & T. S. Munson, The NEOS Server for Optimization - Version 4 and Beyond. Preprint ANL/MCS-TM-253, February 2002. Página web: <http://www-neos.mcs.anl.gov>.
- [13] A. Friedlander, J. M. Martínez & S. A. Santos, A New Trust Region Algorithm for Bound Constrained Minimization, *Applied Mathematics & Optimization* **30**, pp. 235–266, 1994.
- [14] C. Fortin & H. Wolkowicz, The Trust Region Subproblem and Semidefinite Programming. *Optimization Methods and Software*, 2003, to appear.
- [15] W. Gander, Least Squares with a Quadratic Constraint, *Numerische Mathematik* **36**, pp. 291–307, 1981.
- [16] D. M. Gay, Computing optimal locally constrained steps, *SIAM Journal on Scientific and Statistic Computing* **2**, pp. 186–197, 1981.
- [17] G. H. Golub & C. F. van Loan, *Matrix Computations*, 3ed, Baltimore and London: The Johns Hopkins University Press, 1996.
- [18] G. H. Golub & U. von Matt, Quadratically constrained least squares and quadratic problems, *Numerische Mathematik* **59**, pp. 561–580, 1991.
- [19] N. I. M. Gould, S. Lucidi, M. Roma & Ph. L. Toint, Solving the trust-region subproblem using the Lanczos method, *SIAM Journal on Optimization* **9**, pp. 504–525, 1999.
- [20] W. W. Hager, Minimizing a quadratic over a sphere, *SIAM Journal on Optimization* **12**, pp. 188–208, 2001.
- [21] M. D. Hebden, An algorithm for minimization using exact second derivatives, Atomic Energy Research Establishment, Report T.P. 515, Harwell, England, 1973.
- [22] K. Levenberg, A method for the solution of certain problems in least squares, *Quarterly of Applied Mathematics* **2**, pp. 164–168, 1944.
- [23] S. Lucidi, L. Palagi & M. Roma, On some properties of quadratic programs with a convex quadratic constraint, *SIAM Journal on Optimization* **8**, pp. 105–122, 1998.
- [24] D. Marquardt, An algorithm for least-squares estimation of nonlinear parameters, *SIAM Journal on Applied Mathematics* **11**, pp. 431–441, 1963.

- [25] J. M. Martínez & L. Martínez, Packing optimization for automated generation of complex system's initial configurations for molecular dynamics and docking, *Journal of Computational Chemistry* **24**, pp. 819–825, 2003.
- [26] J. M. Martínez & S. A. Santos, A trust-region strategy for minimization on arbitrary domains, *Mathematical Programming* **65**, pp. 267–301, 1995.
- [27] J. M. Martínez & S. A. Santos, New convergence results on an algorithm for norm constrained regularization and related problems, *Recherche opérationnelle/Operations Research* **31**, pp. 269–294, 1997.
- [28] J. J. Moré, Recent advances in algorithms and software for trust region methods, (A. Bachem, M. Grötschel & B. Korte eds.), *Mathematical Programming: The State of the Art*, pp.258–287. Heidelberg, Berlin, New York: Springer-Verlag, 1983.
- [29] J. J. Moré & D. C. Sorensen, Computing a trust region step, *SIAM Journal on Scientific Computing* **4**, pp. 553–572, 1983.
- [30] T. Pham Dinh & L. T. Hoai An, A D. C. Optimization Algorithm for Solving the Trust-Region Subproblem, *SIAM Journal on Optimization* **8**, pp. 476–505, 1998.
- [31] M. J. D. Powell, A new algorithm for unconstrained optimization. *Nonlinear Programming* (J. B. Rosen, O. L. Mangasarian, K. Ritter eds.), pp. 31–65, London: Academic Press, 1970.
- [32] M. J. D. Powell, A hybrid method for nonlinear equations. *Numerical Methods for Nonlinear Algebraic Equations* (P. Rabinowitz ed.), pp. 88–114, London: Gordon and Breach, 1970.
- [33] C. Reinsch, Smoothing by Spline Functions, *Numerische Mathematik* **10**, pp. 177–183, 1967.
- [34] C. H. Reinsch, Smoothing by spline functions II, *Numerische Mathematik* **16**, pp. 451–454, 1971.
- [35] F. Rendl & H. Wolkowicz, A semidefinite framework for trust region subproblems with application to large scale minimization, *Mathematical Programming* **77**, pp. 273–299, 1997.
- [36] M. Rojas, A large-scale trust-region approach to the regularization of discrete ill-posed problems. Ph.D.thesis, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, Technical Report TR98-19, May 1998.

- [37] M. Rojas, S. A. Santos & D. C. Sorensen, A New Matrix-Free Algorithm for the Large-Scale Trust-Region Subproblem, *SIAM Journal on Optimization* **11**, pp. 611–646, 2000.
- [38] M. Rojas, S. A. Santos & D. C. Sorensen, LSTRS: Matlab Software for Large-Scale Trust-Region Subproblems and Regularization, Technical Report 2003-4, Department of Mathematics, Wake Forest University, August 26, 2003. A aparecer em *ACM Transactions on Mathematical Software*, 2007.
- [39] M. Rojas & D. C. Sorensen, A trust-region approach to the regularization of large-scale discrete forms of ill-posed problems, *SIAM Journal on Scientific Computing* **23** (3), pp. 1843–1861, 2002.
- [40] D. C. Sorensen, Minimization of a large-scale quadratic function subject to a spherical constraint, *SIAM Journal on Optimization* **7**, pp. 141–161, 1997.
- [41] D. C. Sorensen, Newton’s method with a model trust region modification, *SIAM Journal on Numerical Analysis* **19**, pp. 409–426, 1982.
- [42] D. C. Sorensen, Implicit application of polynomial filters in a K-step Arnoldi method, *SIAM Journal on Matrix Analysis and Applications* **13**, pp. 357–385, 1992.
- [43] T. Steihaug, The conjugate gradient method and trust regions in large scale optimization, *SIAM Journal on Numerical Analysis* **20**, pp. 626–637, 1983.
- [44] A. N. Tikhonov, A. V. Goncharsky, V. V. Stepanov & A. G. Yagola, *Numerical Methods for the Solution of Ill-Posed Problems*, Mathematics and Its Applications, vol. 328, Dordrecht: Kluwer, 1995.
- [45] Ph. L. Toint, Towards an efficient sparsity exploiting Newton methods for minimization, *Sparse Matrices and Their Uses* (I. S. Duff ed.), pp. 57–88, London: Academic Press, 1981.