

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE MATEMÁTICA

Tiara Martini

ALGORITMO NÃO MONÓTONO PARA MINIMIZAÇÃO
EM DOMÍNIOS ARBITRÁRIOS E APLICAÇÕES

Florianópolis

2012

Tiara Martini

**ALGORITMO NÃO MONÓTONO PARA MINIMIZAÇÃO
EM DOMÍNIOS ARBITRÁRIOS E APLICAÇÕES**

Dissertação submetida à Pós Graduação
em Matemática e Computação Científica
para a obtenção do Grau de Mestre
em Matemática, com Área de Con-
centração em Matemática Aplicada.
Orientador: Prof. Dr. Juliano de Bem
Francisco (UFSC)

Florianópolis

2012

Catálogo na fonte pela Biblioteca Universitária
da
Universidade Federal de Santa Catarina

M386a Martini, Tiara

Algoritmo não monótono para minimização em domínios arbitrários e aplicações [dissertação] / Tiara Martini ; orientador, Juliano de Bem Francisco. - Florianópolis, SC, 2012.

113 p.: grafs., tabs.

Dissertação (Mestrado) - Universidade Federal de Santa Catarina, Centro de Ciências Físicas e Matemáticas. Programa de Pós-Graduação em Matemática e Computação Científica.

Inclui referências

1. Matemática. 2. Algoritmos. 3. Autovalores. I. Francisco, Juliano de Bem. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Matemática e Computação Científica. III. Título.

CDU 51

Tiara Martini

ALGORITMO NÃO MONÓTONO PARA MINIMIZAÇÃO EM DOMÍNIOS ARBITRÁRIOS E APLICAÇÕES

Esta Dissertação foi julgada aprovada para a obtenção do Título de “Mestre em Matemática, com Área de Concentração em Matemática Aplicada”, e aprovada em sua forma final pela Pós Graduação em Matemática e Computação Científica.

Florianópolis, 02 de março 2012.

Prof. Dr. Ruy Exel Filho
Coordenador do Curso

Banca Examinadora:

Prof. Dr. Juliano de Bem Francisco (UFSC)
Orientador

Prof^a. Márcia Aparecida Gomes Ruggiero (Unicamp)

Prof. Fermín Sinfórano Viloche Bazán (UFSC)

Prof^ª. Melissa Weber Mendonça (UFSC)

À minha família.
À Deus.

AGRADECIMENTOS

À Deus acima de tudo.

Aos meus pais, Luiz Gonzaga e Maria Goreti, por me incentivarem sempre, por todo carinho e compreensão pelas vezes em que eu precisava estudar e não podia dar toda atenção que eles merecem.

Ao meu namorado, Bruno, por me ajudar a respirar fundo e seguir em frente, por compreender a dificuldade dessa jornada e pela ótima companhia.

Aos colegas das salas 106 e 107, em especial à Helena e Soyara, por toda ajuda, pelas conversas descontraídas e por fazerem da sala de aula um lugar mais agradável.

Ao meu orientador, professor Juliano, agradeço pelos ensinamentos transmitidos, por toda paciência, dedicação e constante incentivo.

Ao professores Fermín, Márcia e Melissa por aceitarem compor a banca, dedicarem tempo para a leitura do trabalho e contribuírem para seu aperfeiçoamento.

Também agradeço todos os outros professores que me auxiliaram deste a graduação até o término desta dissertação.

À CAPES, pelo auxílio financeiro nestes dois anos.

RESUMO

Apresentamos e analisamos um método globalmente convergente e não monótono para minimização em conjuntos fechados. Desenvolvido recentemente por Francisco e Viloche Bazán [17] esse método está baseado nas ideias dos métodos de região de confiança e Levenberg-Marquardt. Dessa maneira, os subproblemas consistem em minimizar um modelo quadrático da função objetivo sujeito ao conjunto de restrições.

Incorporamos conceitos de bidiagonalização e de cálculo da SVD de maneira “inexata” buscando melhorar o desempenho do algoritmo, visto que a solução do subproblema por técnicas tradicionais, necessária em cada iteração, é computacionalmente muito cara.

Outros métodos viáveis são citados, entre eles um método de busca curvilínea e um de minimização ao longo de geodésicas.

O desempenho dos métodos quando aplicados a problemas conhecidos é ilustrado numericamente.

Palavras-chave: Restrições de ortogonalidade, algoritmo não monótono, problema de autovalores, problema de Procrustes ortogonal

ABSTRACT

We present and analyze a nonmonotone globally convergent method for minimization on closed sets. Recently developed by Francisco e Viloche Bazán [17] this method is based on the ideas from trust-region and Levenberg-Marquardt methods. Thus, the subproblems consist on minimize a quadratic model of the objective function subject to the constraint set.

We incorporate concepts of bidiagonalization and calculation of the SVD “with inaccuracy” to improve the performance of the algorithm, since the solution of the subproblem by traditional techniques, which is required in each iteration, is computationally very expensive.

Other feasible methods are mentioned, including a curvilinear search algorithm and a minimization along geodesics algorithm.

The performance of methods when applied to known problems is illustrated numerically.

Keywords: Orthogonality constraints, nonmonotone algorithm, eigenvalue problems, orthogonal Procrustes problem

LISTA DE FIGURAS

Figura 1	Exemplo de conjunto tangente em x^*	9
Figura 2	Função Rosenbrock (esquerda) e Função Cubo (direita)	16
Figura 3	Comportamento de $\ln f(x_k)$ para a função de Rosenbrock (esquerda) e Cubo (direita).....	17
Figura 4	Iteração do Algoritmo Modelo para minimização com restrições de ortogonalidade, em que $c_k = \rho + \sigma_\rho^k$	39
Figura 5	Comportamento do Algoritmo 4.1 não monótono para o Problema 1 com $m = 500$ e $q = 50$	55
Figura 6	Comportamento do Algoritmo 4.1 não monótono para o Problema 2 com $m = 50$ e $q = 10$	56
Figura 7	Comportamento do Algoritmo 4.1 não monótono para os Problemas 4.1 e 4.2 com $q = 10$	60
Figura 8	Comportamento do Algoritmo 4.1 para o Problema 5 com $n = 500$ e $q = 1$	73
Figura 9	Comportamento dos Algoritmos 4.1 e 5.3 para o Problema 6 com $n = 100$ e $q = 1$	75
Figura 10	Comportamento dos Algoritmos 5.3 e 5.2 para o Problema 7 com $n = 1000$ e $q = 6$	75
Figura 11	Comportamento dos Algoritmos 4.1 e 5.1 para o Problema 8 com $n = 100$ e $q = 1$	77

LISTA DE TABELAS

Tabela 1	Resultados numéricos para o Método de Newton.....	16
Tabela 2	Comparativo entre os casos monótono ($M = 0$) e não monótono ($M = 7$) do Algoritmo 4.1 para o Problema 1 com $m = 500$	54
Tabela 3	Desempenho dos métodos para o Problema 1	56
Tabela 4	Comparativo entre os casos monótono e não monótono do Algoritmo 4.1 para o Problema 2 com $m = 100$	57
Tabela 5	Desempenho dos métodos para o Problema 2	57
Tabela 6	Comparativo entre os casos monótono e não monótono do Algoritmo 4.1 para o Problema 3 com $m = 100$	58
Tabela 7	Desempenho dos métodos para o Problema 3	59
Tabela 8	Comparativo entre os casos monótono e não monótono do Algoritmo 4.1 para os Problemas 4.1 e 4.2.....	60
Tabela 9	Desempenho dos métodos para os Problemas 4.1 e 4.2	61
Tabela 10	Comparativo entre os Algoritmos 4.1 e 5.4 para os Problemas 1, 2 e 3.....	64
Tabela 11	Desempenho do método SGKB utilizando as matrizes dos Problema 1 e 4	69
Tabela 12	Comparativo entre os métodos para o Problema 5	72
Tabela 13	Comparativo entre os casos monótono e não monótono do Algoritmo 4.1 para o Problema 5.....	73
Tabela 14	Comparativo entre os métodos para o Problema 6	74
Tabela 15	Comparativo entre os métodos para o Problema 7	76
Tabela 16	Comparativo entre os casos monótono e não monótono do Algoritmo 4.1 para o Problema 8.....	77
Tabela 17	Comparativo entre os métodos para o Problema 8	78
Tabela 18	Comparativo entre os Algoritmos 4.1 e SISVD para os Problemas 5, 6 e 7	78

LISTA DE ALGORITMOS

2.1	Algoritmo para minimização irrestrita com Armijo Não Monótono	14
3.1	Algoritmo Modelo	21
4.1	Algoritmo para problemas com restrições de ortogonalidade (GPST)	42
5.1	Método da iteração ortogonal inversa (Potência)	48
5.2	Método de Newton para minimização em variedades de Stiefel (sg_min)	49
5.3	Método de busca curvilinear com Barzilai-Borwein (OptStiefel)	52
5.4	SVD Inexata	63
5.5	Processo de Lanczos	65
5.6	Bidiagonalização (GKB)	67

LISTA DE ABREVIATURAS E SIGLAS

KKT	Condições de Karush-Kuhn-Tucker	8
LICQ	Condição de Qualificação de Independência Linear	9
SVD	Decomposição em Valores Singulares	14
CRCQ	Condição de Qualificação de Posto Constante	43
LI	Linearmente Independente	43
LD	Linearmente Dependente	45
WOPP	Problema de Procrustes Ortogonal Ponderada	53
OPP	Problema de Procrustes Ortogonal	53

LISTA DE SÍMBOLOS

$f \in \mathcal{C}^k$	f tem derivadas contínuas até ordem k em um aberto que contém Ω	3
$A > 0$	Matriz definida positiva.....	4
$A \geq 0$	Matriz semidefinida positiva.....	4
$g(x)$	$\nabla f(x)$	4
$\nabla^2 f(x)$	Matriz Hessiana de f no ponto x	7
$X \subseteq^\infty Y$	X é um subconjunto infinito de Y	26
$\lim_{k \in \mathcal{K}} x_k$	$\lim_{k \rightarrow \infty} x_k$, restrito a $k \in \mathcal{K}$	26
I_n	Matriz identidade $n \times n$	35
$\text{vec}(X)$	Dada uma matriz $X \in \mathbb{R}^{m \times n}$, denotamos $\text{vec}(X) = (x_1^T \ \cdots \ x_n^T)^T \in \mathbb{R}^{mn}$, em que $x_i \in \mathbb{R}^m$ é a i -ésima coluna de X	43
$e_k \in \mathbb{R}^n$	k - ésimo vetor canônico de \mathbb{R}^n	66

SUMÁRIO

INTRODUÇÃO	1
1 CONCEITOS PRELIMINARES DE OTIMIZAÇÃO RESTRITA	5
1.1 CONDIÇÕES DE OTIMALIDADE	6
1.2 CONDIÇÕES DE KARUSH-KUHN-TUCKER	8
1.2.1 Restrições de Igualdade	8
1.2.2 Restrições de Desigualdade	11
2 ALGORITMO NÃO MONÓTONO PARA MINIMIZAÇÃO IRRESTRITA	13
3 ALGORITMO PARA MINIMIZAÇÃO EM CONJUNTOS FECHADOS	19
3.1 DESCRIÇÃO DO ALGORITMO	20
3.2 ANÁLISE DE CONVERGÊNCIA	23
4 APLICAÇÃO EM MINIMIZAÇÃO COM RESTRIÇÕES DE ORTOGONALIDADE	33
4.1 CONDIÇÕES DE OTIMALIDADE PARA MINIMIZAÇÃO COM RESTRIÇÕES DE ORTOGONALIDADE	34
4.2 UM ALGORITMO PARA MINIMIZAÇÃO COM RESTRIÇÕES DE ORTOGONALIDADE	38
5 RESULTADOS NUMÉRICOS	47
5.1 MÉTODO DA ITERAÇÃO ORTOGONAL INVERSA	47
5.2 MÉTODO DE MINIMIZAÇÃO ATRAVÉS DE GEODÉSICAS	48
5.3 MÉTODO DE BUSCA CURVILINEAR	50
5.4 TESTE NUMÉRICOS E PROBLEMAS	52
5.4.1 Problema de Procrustes Ortogonal	52
5.4.2 Usando a SVD Inexata	60
5.4.3 Usando Bidiagonalização de Golub-Kahan	65
5.4.4 Cálculo dos menores autovalores de uma matriz simétrica	69
6 CONSIDERAÇÕES FINAIS	79
REFERÊNCIAS	81

INTRODUÇÃO

Assim como em muitos ramos da matemática, a otimização teve sua origem nas aplicações. Apesar de ter fundamental importância nos dias atuais, durante muito tempo ficou às margens da ciência. Apenas no final dos anos 30 e início da década de 40, durante a Segunda Guerra Mundial, que apareceram os resultados inspiradores para o desenvolvimento dessa área do conhecimento, que durante muito tempo ficou restrita às operações militares. Após a guerra, com a ascensão industrial, as técnicas existentes foram utilizadas para auxiliar o planejamento e controle de produção, novos resultados teóricos começaram a ser formulados, e aos poucos foi conquistando a comunidade matemática.

Em um problema de otimização busca-se determinar os valores extremos de uma função, isto é, o maior ou o menor valor que a função pode assumir. Podemos classificar estes problemas com respeito às variáveis e com respeito às funções envolvidas. Para o primeiro deles, temos os problemas irrestritos ou restritos. No caso irrestrito busca-se encontrar um ponto de \mathbb{R}^n que resolva o problema, enquanto que no caso restrito busca-se um ponto em um subconjunto de \mathbb{R}^n . Com respeito às funções envolvidas, temos os problemas lineares e não-lineares. No problema linear todas as funções consideradas, inclusive as restrições, devem ser lineares. Já no caso não-linear muitas outras funções podem ser exploradas: quadráticas, cúbicas, trigonométricas, etc., ou seja, funções não-lineares.

O objetivo deste trabalho é minimizar uma função em conjuntos fechados arbitrários. O método escolhido para direcionar esse estudo foi recentemente introduzido por Francisco e Viloche Bazán [17], e estende as ideias apresentadas em [16]. Para tanto, foram acrescentados um esquema não monótono, a condição de gradiente Lipschitz e retirada a hipótese de regularidade fraca introduzida por Martínez e Santos [32]. Como resultado os autores obtiveram um algoritmo que converge para um ponto estacionário do problema independentemente da aproximação inicial utilizada. Tal algoritmo está baseado nos métodos de região de confiança [32] e nas ideias do método de Levenberg-Marquardt [33], gerando novos iterados através da minimização de um modelo quadrático da função em torno do iterado atual.

Aplicamos o algoritmo estudado em um tipo particular de restrições, a saber, restrições de ortogonalidade. Ou seja, almejamos resol-

ver, de maneira rápida e eficiente, o seguinte problema de programação não linear:

$$\begin{aligned} \min \quad & f(X) \\ \text{s.a.} \quad & X^T X = I \\ & X \in \mathbb{R}^{m \times n}, \quad m \geq n, \end{aligned} \tag{1}$$

em que $f : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ é continuamente diferenciável. Como consequência, obtemos um algoritmo globalmente convergente do tipo gradiente projetado espectral para resolver (1).

Problemas formulados como em (1) têm aplicações em otimização polinomial [12], otimização combinatorial [9], problemas de autovalores (linear e não-linear), movimento de corpos rígidos [39], psicometria [23], processamento de sinal [43], dinâmica estrutural [36], planejamento, cálculo de estruturas eletrônicas, entre outros. Resolvê-los, porém, não é uma tarefa fácil. Na maioria dos casos, encontrar uma solução é possível apenas computacionalmente, através de um método iterativo capaz de lidar com dificuldades como vários pontos estacionários e restrições não convexas, gerando uma sequência de pontos viáveis que converjam para um minimizador local. Gerar essa sequência factível acarreta um custo computacional elevado, pois envolve reortogonalização da matriz, ou então, gerar pontos ao longo de geodésicas. No primeiro caso, é necessário alguma decomposição da matriz, em geral a Decomposição em Valores Singulares, e o segundo está relacionado com exponencial de matrizes ou solução de equações diferenciais parciais. Além disso, na maioria dos casos, esses problemas são classificados como NP-hard¹. Recentemente foi proposto um método livre de SVD e de geodésicas que parece ser bastante promissor [44]. No capítulo de resultados numéricos falaremos mais sobre ele.

Métodos usais para a resolução dessa classe de problemas constituem em região de confiança [16], busca curvilinear [44], relaxação [6], métodos de Newton e gradiente conjugado ao longo de geodésicas [14].

O método estudado utiliza um parâmetro de regularização que controla o tamanho do passo, à exemplo dos métodos de região de confiança. O minimizador do modelo é considerado um candidato a próximo iterado, e é aceito se o valor da função objetivo apresentar

¹Diz-se que um problema Π é NP-hard se um algoritmo de tempo polinomial para Π implicar em um algoritmo de tempo polinomial para todos os problemas em NP. Isto é, Π é NP-hard \Leftrightarrow (se Π pode ser resolvido em tempo polinomial, então $P=NP$). A classe de problemas P é definida por todos os problemas que podem ser resolvidos em tempo polinomial, e a classe de problemas NP é definida por todos os problemas que podem ser verificados mas não podem ser resolvidos em tempo polinomial.

uma redução suficiente quando comparada com a redução do modelo neste mesmo ponto. Porém, se o decréscimo não for considerado suficiente, aumentamos o parâmetro de regularização e obtemos um novo candidato. Aumentar o parâmetro de regularização é equivalente a reduzir o raio da região de confiança, no método de região de confiança.

A convergência global para pontos estacionários é obtida através do uso das ideias de não monotonia de Grippo, Lampariello e Lucidi [21]. Ressaltamos que, devido à eficiência e competitividade apresentados por métodos com busca linear não monótona [11], vários autores passaram a incorporar essa estratégia de globalização em seus algoritmos [5, 18, 22, 41, 45].

Buscando melhorar o desempenho do método aqui estudado, incorporamos uma abordagem alternativa para o cálculo da SVD e, também, uma estratégia de bidiagonalização. Com a primeira, nosso objetivo é reduzir o esforço computacional envolvido nesse processo, já com a segunda, pretendemos resolver um problema de menor dimensão, equivalente ao original.

O algoritmo é então aplicado para resolver os problemas de Procrustes Ortogonal e de encontrar o subespaço invariante associado aos menores autovalores de uma matriz simétrica. Para cada um desses problemas, comentamos suas aplicações e comparamos seu desempenho com métodos usuais.

A organização do trabalho encontra-se da seguinte maneira. No primeiro capítulo relembramos os conceitos básicos de otimização com restrições de igualdade e desigualdade, incluindo as condições de optimalidade para tais problemas. No Capítulo 2 apresentamos as ideias de não monotonia e realizamos dois experimentos numéricos, comparando busca linear monótona e não monótona. No Capítulo 3 é colocado o algoritmo principal do trabalho e é feita a análise de convergência. No Capítulo 4 aplicamos o algoritmo para minimização com restrições de ortogonalidade. No Capítulo 5 apresentamos os problemas testados, assim como os métodos utilizados para comparar o desempenho do algoritmo proposto. Posteriormente os resultados obtidos são exibidos. Finalmente, no último capítulo as considerações finais sobre o trabalho são colocadas.

Notação:

- Para indicar que $f : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ tem derivadas contínuas até ordem k em um aberto que contém Ω , faremos uso do símbolo $f \in \mathcal{C}^k$.

- Para uma matriz $A \in \mathbb{R}^{m \times n}$ definida positiva, reservamos o símbolo $A > 0$, da mesma maneira $A \geq 0$ indicará que A é uma matriz semidefinida positiva.
- Vamos denotar por $g(x)$ o gradiente de f no ponto x , isto é, $g(x) \equiv \nabla f(x)$.
- Dado $x \in \mathbb{R}^n$, $\|x\|$ representa a norma euclidiana de \mathbb{R}^n , isto é, $\|x\| = \sqrt{x^T x}$.
- Dadas $A, B \in \mathbb{R}^{m \times n}$, o produto interno entre A e B é definido por $\langle A, B \rangle = \text{traço}(A^T B)$.

1 CONCEITOS PRELIMINARES DE OTIMIZAÇÃO RESTRITA

A necessidade de minimizar ou maximizar uma função é mais natural do que imaginamos. Problemas práticos de otimização emergem, por exemplo, da modelagem matemática de situações cotidianas em engenharia, administração, transporte, economia e biologia. Podemos pensar em minimizar custos, riscos, distâncias ou em maximizar lucro, desempenho, entre outros. Dependendo dos valores que a variável considerada pode assumir, classificamos o problema como irrestrito ou restrito. No problema irrestrito busca-se encontrar um ponto de \mathbb{R}^n que minimiza (ou maximiza) uma função, enquanto que no caso restrito busca-se um ponto em um subconjunto de \mathbb{R}^n . Neste trabalho vamos focar no último deles e começamos resumindo sua teoria.

Um problema clássico de minimização restrita pode ser escrito da seguinte maneira

$$\begin{aligned} \min \quad & f(x) \\ \text{s.a} \quad & h_1(x) = 0 \\ & h_2(x) \leq 0, \end{aligned} \tag{1.1}$$

em que a função objetivo $f : \mathbb{R}^n \rightarrow \mathbb{R}$, e as restrições $h_1 : \mathbb{R}^n \rightarrow \mathbb{R}^p$ e $h_2 : \mathbb{R}^n \rightarrow \mathbb{R}^q$ são, em geral, funções continuamente diferenciáveis. Dizemos que $\Omega = \{x \in \mathbb{R}^n : h_1(x) = 0 \text{ e } h_2(x) \leq 0\}$ é o conjunto viável e que os pontos de Ω são os pontos viáveis de (1.1).

O ponto $x^* \in \Omega$ que resolve o problema (1.1) é chamado de minimizador de f em Ω , este pode ser local ou global, conforme a definição a seguir.

Definição 1.1 Dizemos que x^* é **minimizador local** de (1.1) se existe $\varepsilon > 0$ tal que $f(x^*) \leq f(x)$, para todo $x \in \Omega$ satisfazendo $\|x - x^*\| \leq \varepsilon$. Se, além disso, $f(x^*) \leq f(x)$, para todo $x \in \Omega$, então x^* é dito **minimizador global** de (1.1). Neste caso, $f(x^*)$ é chamado de **mínimo** de f em Ω .

A garantia de existência de um minimizador global é estabelecida no Teorema de Bolzano-Weierstrass, conforme se segue:

Teorema 1.2 Se Ω é compacto e $f : \Omega \rightarrow \mathbb{R}$ é contínua, então existe $x^* \in \Omega$ minimizador global de (1.1).

Demonstração: Ver Marsden [30].

Visto que maximizar os valores de uma função f é equivalente a minimizar os valores de $-f$, tudo o que apresentarmos para um problema de minimização continua válido para um problema de maximização.

1.1 CONDIÇÕES DE OTIMALIDADE

Condições de otimalidade são condições necessárias e/ou suficientes que um minimizador local do problema de otimização deve satisfazer, geralmente relacionam as derivadas da função objetivo e as derivadas das restrições. As condições que vamos estabelecer nesta seção são provenientes de curvas especiais definidas no conjunto viável, conforme a definição a seguir.

Definição 1.3 *Dados $x^* \in \Omega$ e $\varepsilon > 0$, dizemos que $\alpha : [0, \varepsilon) \rightarrow \Omega$ é uma curva viável partindo de x^* se α é contínua e $\alpha(0) = x^*$.*

Tendo em vista o conceito de curva viável, podemos enunciar uma condição de otimalidade que envolve apenas as derivadas de primeira ordem da função objetivo. E, como consequência imediata deste resultado, uma das condições de otimalidade mais simples de serem verificadas.

Teorema 1.4 *(Condição Necessária de Primeira Ordem) Sejam x^* minimizador local de (1.1) e α curva viável partindo de x^* , então $\langle \nabla f(x^*), \alpha'(0) \rangle \geq 0$.*

Demonstração: Ver Martínez e Santos [31].

Corolário 1.5 *Seja x^* ponto interior de Ω tal que x^* é minimizador local de f em Ω , então $\nabla f(x^*) = 0$.*

Demonstração: Sejam $\alpha_1, \alpha_2 : [0, \varepsilon) \rightarrow \mathbb{R}^n$ curvas viáveis partindo de x^* definidas, respectivamente, por:

$$\alpha_1(t) = x^* + t\nabla f(x^*)$$

e

$$\alpha_2(t) = x^* - t\nabla f(x^*).$$

Note que, $\alpha'_1(t) = \nabla f(x^*)$ e $\alpha'_2(t) = -\nabla f(x^*)$. Logo, pelo Teorema 1.4

$$\langle \nabla f(x^*), \nabla f(x^*) \rangle \geq 0$$

e

$$\langle \nabla f(x^*), -\nabla f(x^*) \rangle \geq 0.$$

Segue que $\langle \nabla f(x^*), \nabla f(x^*) \rangle = 0$ e, portanto $\nabla f(x^*) = 0$. \square

O próximo resultado relaciona informação de segunda ordem da função objetivo com minimizadores locais.

Teorema 1.6 *Seja x^* minimizador local de (1.1). Então, para toda curva viável α partindo de x^* temos que:*

(i) $\langle \nabla f(x^*), \alpha'(0) \rangle \geq 0;$

(ii) *Se $\langle \nabla f(x^*), \alpha'(0) \rangle = 0$, então $\phi''(0) \geq 0$, em que $\phi(t) = f(\alpha(t))$.*

Demonstração: Ver Martínez e Santos [31].

Com a definição a seguir, introduzimos uma classe de curvas no conjunto viável que nos permite enunciar resultados um pouco mais fortes.

Definição 1.7 *Dados $x^* \in \Omega$ e $\varepsilon > 0$, dizemos que $\alpha : (-\varepsilon, \varepsilon) \rightarrow \Omega$ é uma **curva viável passando por x^*** se α é contínua e $\alpha(0) = x^*$.*

Teorema 1.8 *Se x^* é minimizador local de f em Ω , então para toda curva viável passando por x^* tem-se $\langle \nabla f(x^*), \alpha'(0) \rangle = 0$.*

Demonstração: Ver Martínez e Santos [31].

Enunciamos a seguir duas condições de otimalidade envolvendo a matriz Hessiana da função objetivo. O primeiro resultado estabelece uma condição necessária de segunda ordem e o segundo, uma condição suficiente de otimalidade. Ambos para pontos no interior de Ω .

Teorema 1.9 *Seja x^* ponto interior de Ω tal que x^* é minimizador local de (1.1). Então, $\nabla f(x^*) = 0$ e $\nabla^2 f(x^*) \geq 0$, em que $\nabla^2 f(x)$ denota a matriz Hessiana de f no ponto x .*

Demonstração: Ver Martínez e Santos [31].

Teorema 1.10 *Seja $f \in C^2(\Omega)$. Seja x^* ponto interior de Ω tal que $\nabla f(x^*) = 0$ e $\nabla^2 f(x^*) > 0$, então x^* é minimizador local de f em Ω .*

Demonstração: Ver Martínez e Santos [31].

1.2 CONDIÇÕES DE KARUSH-KUHN-TUCKER

Os matemáticos William Karush, da Universidade Estadual da Califórnia, e Harold William Kuhn juntamente com Albert William Tucker, ambos da Universidade de Princeton, estudaram de maneira independente as condições de otimalidade conhecidas atualmente como Condições de Karush-Kuhn-Tucker (KKT). Karush foi o primeiro a estudá-las durante seu curso de Mestrado na Universidade de Chicago [25], porém só tornaram-se conhecidas com o artigo de Kuhn e Tucker apresentado no Segundo Simpósio de Matemática, Estatística e Probabilidade de Berkeley [26].

1.2.1 Restrições de Igualdade

Nesta subseção vamos nos concentrar no seguinte problema

$$\begin{aligned} \min \quad & f(x) \\ \text{s.a.} \quad & h(x) = 0 \end{aligned} \tag{1.2}$$

em que, $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$. Neste caso, $\Omega = \{x \in \mathbb{R}^n : h(x) = 0\}$.

Antes de enunciar as condições KKT para o problema (1.2), apresentamos a definição de conjunto tangente, essencial para os resultados mais adiante.

Definição 1.11 *Dado $x \in \Omega$ o **conjunto tangente** de Ω em x é definido como sendo o conjunto dos vetores tangentes às curvas viáveis passando por x , isto é,*

$$M(x) = \{v \in \mathbb{R}^n : v = \alpha'(0) \text{ para alguma curva viável } \alpha \text{ passando por } x\}.$$

Note que há uma relação entre o conjunto tangente e o núcleo da matriz

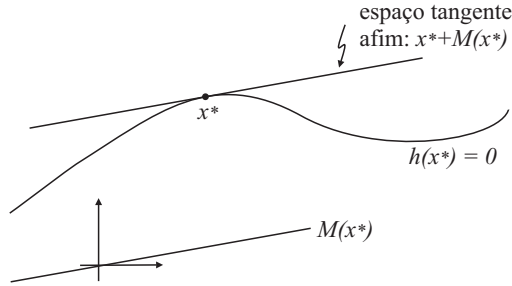


Figura 1: Exemplo de conjunto tangente em x^*

$$h'(x) = \begin{pmatrix} \nabla h_1(x)^T \\ \vdots \\ \nabla h_p(x)^T \end{pmatrix},$$

a saber, para todo $x \in \Omega$, $M(x) \subseteq N(h'(x))$. A igualdade é válida quando o ponto é regular, isto é, quando a matriz $h'(x)$ tem posto completo. Neste caso, iremos dizer que x satisfaz a condição de qualificação de independência linear (LICQ). O resultado citado é colocado no próximo teorema.

Teorema 1.12 *Se x é regular então $M(x) = N(h'(x))$.*

Demonstração: Ver Martínez e Santos [31].

O próximo teorema é bem conhecido da teoria de otimização e tem implicação direta nos métodos computacionais para minimização.

Teorema 1.13 (*Multiplicadores de Lagrange*) *Se x^* é minimizador local regular de (1.2), então existem únicos $\lambda_1, \dots, \lambda_p$ reais tais que $\nabla f(x^*) + \sum_{i=1}^p \lambda_i \nabla h_i(x^*) = 0$.*

Demonstração: Seja $v \in N(h'(x^*))$. Como x^* é ponto regular, existe uma curva viável α passando por x^* tal que $\alpha'(0) = v$. Definindo $\varphi(t) = f(\alpha(t))$, tem-se que 0 é minimizador local de φ , ou seja, $\varphi'(0) = 0$.

Por outro lado,

$$\varphi'(0) = \langle \nabla f(x^*), \alpha'(0) \rangle = \langle \nabla f(x^*), v \rangle.$$

Segue que, $v \perp \nabla f(x^*)$ e, portanto, $N(h'(x^*)) \perp \nabla f(x^*)$.

Logo, $\nabla f(x^*) \in R(h'(x^*)^T)$ e existem $\lambda_1, \dots, \lambda_p$ tais que

$$\nabla f(x^*) + \sum_{i=1}^p \lambda_i \nabla h_i(x^*) = 0.$$

Para mostrar a unicidade, suponha que existam μ_1, \dots, μ_p satisfazendo

$$\nabla f(x^*) + \sum_{i=1}^p \mu_i \nabla h_i(x^*) = 0.$$

Então,

$$\sum_{i=1}^p (\lambda_i - \mu_i) \nabla h_i(x^*) = 0.$$

Como $h'(x^*)$ tem posto completo, $\mu_i = \lambda_i$, $i = 1, \dots, p$, e o resultado segue. □

Os números $\lambda_1, \dots, \lambda_p$ são chamados multiplicadores de Lagrange de (1.2), e o sistema não linear

$$\begin{cases} \nabla f(x^*) + \sum_{i=1}^p \lambda_i \nabla h_i(x^*) = 0 \\ h(x^*) = 0 \end{cases}$$

estabelecem as Condições KKT do problema.

Defina a função Lagrangiana para o problema (1.2) por $l(x, \lambda) = f(x) + \sum_{i=1}^p \lambda_i h_i(x)$. Note que

$$\nabla_x l(x, \lambda) = \nabla f(x) + \sum_{i=1}^p \lambda_i \nabla h_i(x)$$

e

$$\nabla_\lambda l(x, \lambda) = h(x),$$

portanto, as Condições KKT do problema (1.2) podem ser reescritas como $\nabla l(x^*, \lambda^*) = 0$. A seguir, iremos enunciar as condições necessárias de segunda ordem para o problema com restrições de igualdade em termos de seu Lagrangeano.

Teorema 1.14 *Sejam $f, h \in \mathcal{C}^2$, x^* minimizador local regular de (1.2) e λ^* o vetor com os multiplicadores de Lagrange associados à x^* . Então, a matriz Hessiana $\nabla_{xx}^2 l(x^*, \lambda^*) = \nabla^2 f(x^*) + \sum_{i=1}^p \lambda_i^* \nabla^2 h_i(x^*)$ é semi-definida positiva em $N(h'(x^*))$, ou seja, $x^T \nabla_{xx}^2 l(x^*, \lambda^*) x \geq 0$, para*

todo $x \in N(h'(x^*))$.

Demonstração: Ver em Luenberger [29].

Se $\nabla_{xx}^2 l(x^*, \lambda)$ for definida positiva em $N(h'(x^*))$, podemos estabelecer a seguinte condição suficiente de segunda ordem para problemas com restrições de igualdade:

Teorema 1.15 *Sejam $f, h \in \mathcal{C}^2$, x^* satisfazendo as condições necessárias de primeira ordem para (1.2) e λ^* o vetor com os multiplicadores de Lagrange associados à x^* . Se a matriz $\nabla_{xx}^2 l(x^*, \lambda^*)$ é definida positiva em $N(h'(x^*))$, então x^* é minimizador local para (1.2) tal que $f(x^*) < f(x)$.*

Demonstração: Ver Luenberger [29].

1.2.2 Restrições de Desigualdade

Consideremos agora o problema

$$\begin{aligned} \min \quad & f(x) \\ \text{s.a.} \quad & h(x) \leq 0 \end{aligned} \tag{1.3}$$

em que, $h : \mathbb{R}^n \rightarrow \mathbb{R}^q$. Neste caso, $\Omega = \{x \in \mathbb{R}^n : h(x) \leq 0\}$.

As restrições de desigualdade são classificadas em restrições ativas e inativas, conforme a definição seguinte.

Definição 1.16 *Dado um ponto $x \in \Omega$, definimos as **restrições ativas em x** como sendo aquelas para as quais $h_i(x) = 0$, $i \in \{1, \dots, q\}$. De mesmo modo, chamamos **restrições inativas em x** aquelas para as quais $h_i(x) < 0$, $i \in \{1, \dots, q\}$. Os **pontos regulares** de (1.3) são tais que os gradientes das restrições ativas são linearmente independentes.*

Note que, se x^* é minimizador local de (1.3) e $\mathcal{A}(x^*) = \{i \in \{1, \dots, q\} : h_i(x^*) = 0\}$, o conjunto das restrições ativas em x^* , então x^* é, também, minimizador local para o problema

$$\begin{aligned} \min \quad & f(x) \\ \text{s.a.} \quad & h_i(x) = 0, \quad i \in \mathcal{A}(x^*). \end{aligned}$$

Dessa forma, para este problema verificam-se resultados semelhantes aos vistos na subseção anterior. A seguir, enunciamos o mais importante deles.

Teorema 1.17 (*Condições KKT para (1.3)*) Se x^* é minimizador local regular de (1.3), então existem únicos $\mu_i^* \in \mathbb{R}$, $\mu_i^* \geq 0$, $i \in \mathcal{A}(x^*)$, tais que

$$\begin{cases} \nabla f(x^*) + \sum_{i=1}^q \mu_i^* \nabla h_i(x^*) = 0 \\ h_i(x^*) \leq 0 \\ (x^*)^T \mu_* = 0. \end{cases}$$

Demonstração: Ver Martínez e Santos [31]

Como no caso das restrições de igualdade, podemos obter condições necessárias e suficientes de segunda ordem. Mais detalhes podem ser encontrados em [29].

2 ALGORITMO NÃO MONÓTONO PARA MINIMIZAÇÃO IRRESTRITA

O Método de Newton, por ser um dos algoritmos de minimização mais usados e conhecidos na comunidade científica, desempenhou um papel fundamental como ponto de partida para o desenvolvimento de novos métodos para resolver o problema de minimização irrestrita

$$\min f(x), \text{ s.a. } x \in \mathbb{R}^n, \quad (2.1)$$

com $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Entretanto, apresenta um bom comportamento apenas em uma região próxima da solução, ou seja, é um método de convergência local. Para contornar esse problema costuma-se utilizar alguma estratégia de globalização [29, 33], das quais citamos região de confiança e busca linear. Esta última consiste em, a cada iteração, escolher uma direção e o quanto seguir através dela, combinada com uma condição de decréscimo suficiente. O próximo teorema apresenta uma condição bem conhecida na literatura, a saber, a Condição de Armijo.

Teorema 2.1 *Sejam $x, d \in \mathbb{R}^n$ tais que $\nabla f(x) \neq 0$, $\nabla f(x)^T d < 0$ e $\gamma \in (0, 1)$. Então, existe $\varepsilon > 0$ de forma que*

$$f(x + td) \leq f(x) + \gamma t \nabla f(x)^T d, \quad (2.2)$$

para todo $t \in (0, \varepsilon)$.

Demonstração: Ver Nocedal e Wright [33].

Quando incorporamos a Condição de Armijo na busca linear, a seqüência dos valores da função objetivo torna-se monótona decrescente. Porém, pedir monotonicidade pode deixar o algoritmo muito lento, no sentido de serem necessários muitos passos da busca linear e, conseqüentemente, muitas avaliações da função. Nesse contexto, Grippo, Lampariello e Lucidi [21] propuseram uma técnica de busca linear não monótona, apresentada na definição a seguir.

Definição 2.2 *Sejam $\{x_k\}_{k \in \mathbb{N}}$ uma seqüência definida por $x_{k+1} = x_k + t_k d_k$, $d_k \neq 0$, $t_k \in (0, 1)$, $\gamma \in (0, 1)$, $M \in \mathbb{Z}^+$ e $m(0) = 0$. Para cada $k \geq 1$, escolha $m(k)$ recursivamente por*

$$0 \leq m(k) \leq \min\{m(k-1) + 1, M\}, \quad (2.3)$$

e defina

$$f(x_{l(k)}) = \max\{f(x_{k-j}) : 0 \leq j \leq m(k)\}. \quad (2.4)$$

Dizemos que x_{k+1} satisfaz a **Condição de Armijo Não Monótona** se,

$$f(x_{k+1}) \leq f(x_{l(k)}) + \gamma t_k \nabla f(x_k)^T d_k. \quad (2.5)$$

Devido à promessa de tornar o algoritmo de alguma forma mais eficiente, principalmente na minimização de funções que apresentam uma bacia de atração estreita e curva [11], as ideias de não monotonia foram posteriormente utilizadas por Grippo, Lampariello e Lucidi [22], Sun [41], Fu e Sun [18] e Yuan e Yu [45]. Vale mencionar que existem outras abordagens não monótonas, dentre elas citamos a estratégia de Zhang e Hager [46], onde uma média dos valores já calculados da função objetivo substitui $f(x_{l(k)})$ em (2.4).

Ressaltamos que no Capítulo 4 descrevemos um algoritmo para resolver problemas de minimização com restrições de ortogonalidade que, a cada passo da busca linear, necessita calcular a Decomposição em Valores Singulares (SVD) de uma matriz. Tal procedimento exige um alto custo computacional e, por esta razão, é importante reduzirmos o número de buscas lineares no algoritmo o que torna o procedimento não monótono de grande valia.

Colocamos a seguir o algoritmo para minimização irrestrita com busca linear satisfazendo a Condição de Armijo Não Monótona (2.5).

Algoritmo 2.1: Algoritmo para minimização irrestrita com Armijo Não Monótono

Seja $x_0 \in \mathbb{R}^n$. Escolha $M \in \mathbb{Z}^+$, $a > 0$, $\sigma, \gamma \in (0, 1)$ e $c_1, c_2 > 0$.
Faça $k = 0$ e $m(0) = 0$.

Passo 1: Calcule $f(x_k)$, $g(x_k)$ e $f(x_{l(k)})$ como em (2.4). Se $g(x_k) = 0$ declare x_k ponto estacionário.

Passo 2: Escolha $d_k \in \mathbb{R}^n$ tal que $g(x_k)^T d_k < 0$ e

- (i) $\|d_k\| \leq c_1 \|g(x_k)\|^2$,
- (ii) $g(x_k)^T d_k \leq -c_2 \|g(x_k)\|^2$.

Passo 3: Faça $t_k = \sigma^{\tau_k} a$, em que τ_k é o primeiro inteiro não negativo τ que satisfaz

$$f(x_k + \sigma^\tau a d_k) \leq f(x_{l(k)}) + \gamma \sigma^\tau a g(x_k)^T d_k.$$

Passo 4: Faça $x_{k+1} = x_k + t_k d_k$ e $k = k + 1$.

No próximo capítulo falamos de um algoritmo que explora o esquema não monótono dado na Definição 2.2, onde apresentamos uma análise rigorosa de convergência. Por este motivo, os resultados que garantem a convergência do Algoritmo 2.1 para um ponto estacionário serão omitidos. No entanto, mais detalhes podem ser encontrados em [21].

Para efeito de validade desta técnica, apresentamos a seguir exemplos em problemas mal condicionados, a saber, em duas funções sugeridas por Grippo, Lampariello e Lucidi [21]. Resultados numéricos mais elaborados que comparam o algoritmo monótono com o não monótono podem ser encontrados em [42], onde o potencial numérico deste esquema é destacado.

A primeira função considerada, a função Rosenbrock, é definida por

$$f : \mathbb{R}^2 \rightarrow \mathbb{R}$$

$$f(x) = c(x_2 - x_1^2)^2 + (1 - x_1)^2.$$

Foi introduzida por Rosenbrock [38] e, devido a forma de suas curvas de nível, também é conhecida como função banana. Para $c = 100$ a função apresenta uma bacia de atração na direção da parábola $x_2 = x_1^2$, dificultando convergência para o mínimo global. Como ponto de partida vamos considerar $x_0 = [-1.2 \ 1]^T$.

Como segundo exemplo utilizamos a função Cubo

$$f : \mathbb{R}^2 \rightarrow \mathbb{R}$$

$$f(x) = c(x_2 - x_1^3)^2 + (1 - x_1)^2,$$

introduzida por Leon [28] esta função possui uma bacia de atração na direção da curva $x_2 = x_1^3$ e, por essa razão, também aparece frequentemente em testes que avaliam o desempenho de novos métodos de otimização. O ponto inicial considerado será $x_0 = [-1.2 \ -1]^T$.

Os gráficos das funções são colocados na Fig. 2.

A Tab. 1 mostra o potencial numérico da Condição de Armijo Não Monótona na busca linear, especialmente quando o problema é mal condicionado, conforme ilustrado nos casos acima ao aumentarmos o

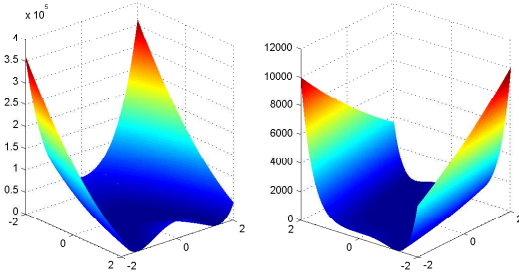


Figura 2: Função Rosenbrock (esquerda) e Função Cubo (direita)

Tabela 1: Resultados numéricos para o Método de Newton

Exemplo	c	M = 0			M = 10		
		N_i	N_a	$f(x^*)$	N_i	N_a	$f(x^*)$
1. Rosenbrock	10^2	21	29	$3.74\text{e-}21$	11	16	$7.42\text{e-}18$
	10^4	80	114	$9.10\text{e-}19$	11	17	0
	10^6	347	493	$3.83\text{e-}18$	9	15	$6.53\text{e-}13$
2. Cubo	10^2	26	37	$4.04\text{e-}14$	10	13	$2.83\text{e-}29$
	10^4	110	160	$3.66\text{e-}20$	8	11	$1.08\text{e-}11$
	10^6	485	712	$2.84\text{e-}16$	8	11	$1.72\text{e-}15$

valor da constante c . Denotamos o número de iterações por N_i e o número de avaliações da função por N_a . Os parâmetros $M = 0$ e $M = 10$ estão associados as Condição de Armijo Monótona e Não Monótona, respectivamente. Na Fig. 3 podemos observar o comportamento de $\ln f(x_k)$ para as funções Rosenbrock e Cubo, ambas definidas com $c = 100$.

Outros resultados numéricos envolvendo este esquema não monótono serão apresentados no Capítulo 5, que, em geral, confirmam sua eficiência mesmo em problemas com restrições não convexas.

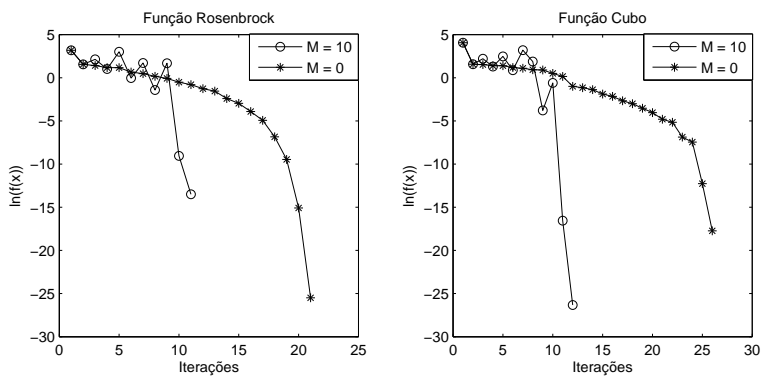


Figura 3: Comportamento de $\ln f(x_k)$ para a função de Rosenbrock (esquerda) e Cubo (direita)

3 ALGORITMO PARA MINIMIZAÇÃO EM CONJUNTOS FECHADOS

O objetivo desse capítulo é apresentar uma maneira eficiente de resolver o problema

$$\begin{aligned} \min \quad & f(x) \\ \text{s.a.} \quad & x \in \Omega, \end{aligned} \tag{3.1}$$

em que Ω é um conjunto fechado, e $f : \mathbb{R}^n \rightarrow \mathbb{R}$ é uma função continuamente diferenciável, definida num conjunto aberto e convexo Ω_c que contém Ω .

Desenvolver um algoritmo utilizando as ideias de região de confiança é um início promissor, conforme visto em [7, 13, 32, 33, 35]. Seguindo nessa direção, apresentamos o método desenvolvido por Francisco e Viloche Bazán [17], que acrescenta um parâmetro de regularização para penalizar a restrição da região de confiança, tornando-o um método do tipo Levenberg-Marquardt. O estudo de Francisco e Viloche Bazán [17] está baseado no método apresentado em [16] para resolver problemas do cálculo de estruturas eletrônicas, os autores estenderam as ideias de [16] para problemas mais gerais de otimização, acrescentaram um esquema não monótono, a condição de gradiente Lipschitz e retiraram a hipótese de regularidade fraca introduzida por [32].

Os resultados de convergência aqui apresentados também foram retirados de [17], onde técnicas de não monotonia [21] e de região de confiança são utilizadas para garantir a convergência global.

Assim como os demais algoritmos elaborados para resolver um problema de minimização com restrições, o aqui apresentado calcula, de maneira iterativa, uma sequência de pontos viáveis que se aproximam de algum ponto estacionário da função objetivo.

Considere os conjuntos:

$$\mathcal{S}_n = \{A \in \mathbb{R}^{n \times n} : A^T = A\},$$

$$\mathcal{S}_n^+ = \{A \in \mathcal{S}_n : x^T A x > 0 \ \forall x \in \mathbb{R}^n\},$$

e assumamos g Lipschitz contínua em Ω_c , ou seja, que existe $L_f > 0$ tal que

$$\|g(y) - g(x)\| \leq L_f \|y - x\|, \quad \forall x, y \in \Omega_c. \tag{3.2}$$

Então,

$$\begin{aligned}
 |f(y) - f(x) - \langle g(x), y - x \rangle| &= \left| \int_0^1 \langle g(x + t(y - x)) - g(x), y - x \rangle dt \right| \leq \\
 &\leq \int_0^1 \|g(x + t(y - x)) - g(x)\| \|y - x\| dt \leq \|y - x\| \int_0^1 L_f \|x + t(y - x) - x\| dt = \\
 &= \|y - x\|^2 L_f \int_0^1 |t| dt = \frac{L_f}{2} \|y - x\|^2,
 \end{aligned}$$

ou seja,

$$f(y) \leq f(x) + \langle g(x), y - x \rangle + \frac{L_f}{2} \|y - x\|^2, \quad \forall x, y \in \Omega_c. \quad (3.3)$$

Note que (3.2) é sempre válida para algum $L_f > 0$ quando Ω é compacto e $f \in \mathcal{C}^2(\Omega_c)$.

Conforme comentado anteriormente, utilizaremos as ideias de não monotonia [21] na elaboração do algoritmo, descrito na próxima seção.

3.1 DESCRIÇÃO DO ALGORITMO

Como usual em métodos de região de confiança, a ideia do algoritmo é calcular um ponto candidato, solução da minimização de um modelo quadrático da função objetivo em uma vizinhança do ponto anterior, e a decisão de aceitá-lo ou rejeitá-lo como próximo ponto da sequência gerada é tomada com base na redução da função objetivo quando comparada com a redução prevista pelo modelo.

Para tanto, considere o seguinte modelo quadrático:

$$Q_\rho(x) = \langle g(y), x - y \rangle + \frac{1}{2} (x - y)^T (B + \rho A) (x - y), \quad (3.4)$$

em que, $\rho > 0$ é um parâmetro de regularização, $A \in \mathcal{S}_n^+$ e $B \in \mathcal{S}_n$.

Dessa forma, em cada passo k do algoritmo calculamos o ponto

$$\bar{x}_\rho^k = \underset{s.a. \quad x \in \Omega,}{\arg \min} Q_\rho^k(x)$$

em que, Q_ρ^k é a quadrática definida em (3.4), com $y = x_k$ (iterado atual), $B = B_\rho^k \in \mathcal{S}_n$ e $A = A_\rho^k \in \mathcal{S}_n^+$. Feito isso, o ponto candidato será um ponto $x_\rho^k \in \Omega$ que satisfaz

$$Q_\rho^k(x_\rho^k) \leq \eta Q_\rho^k(\bar{x}_\rho^k),$$

em que $\eta \in (0, 1]$.

Se $Q_\rho^k(x_\rho^k) = 0$, paramos o algoritmo e declaramos x_k ponto estacionário de (3.1). Senão, calculamos a redução real da função objetivo, a saber,

$$ared_\rho^k = f(x_{l(k)}) - f(x_\rho^k),$$

e a redução prevista pelo modelo,

$$pred_\rho^k = -\Psi_k(x_\rho^k) = -[\langle g(x_k), x_\rho^k - x_k \rangle + \frac{1}{2}(x_\rho^k - x_k)^T B_\rho^k (x_\rho^k - x_k)],$$

e avaliamos se elas satisfazem

$$\frac{ared_\rho^k}{pred_\rho^k} \geq \beta_1, \quad (3.5)$$

com $\beta_1 \in (0, \frac{1}{2}]$. Note que a estratégia não monótona está incluída na definição de $ared_\rho^k$. A condição (3.5) é típica em algoritmos baseados em região de confiança e é adotada ao longo deste trabalho para globalizar a convergência da sequência de iterados. Se (3.5) for satisfeita, tomamos o próximo iterado como sendo o ponto candidato, isto é, $x_{k+1} = x_\rho^k$. Caso contrário, aumentamos o valor de ρ e reiniciamos o processo.

Apresentamos a seguir o algoritmo modelo.

Algoritmo 3.1: Algoritmo Modelo

Tome $x_0 \in \Omega$, $\eta \in (0, 1]$, $M \in \mathbb{Z}^+$, $\beta_1 \in (0, \frac{1}{2})$, $L_u \geq L_f$, $\rho_{lim} > 0$, $0 < \rho_{min} \leq \rho_{max} < +\infty$ e $1 < \xi_1 \leq \xi_2 < +\infty$.
Faça $k = 0$ e $m(0) = 0$.

Passo 1. Calcule $f(x_{l(k)})$ como em (2.4), $g(x_k)$ e tome $\rho \in [\rho_{min}, \rho_{max}]$.

Passo 2. Se $\rho \leq \rho_{lim}$, escolha $B_\rho^k \in \mathcal{S}_n$ e $A_\rho^k \in \mathcal{S}_n^+$. Senão, faça $B_\rho^k = L_u I$ e $A_\rho^k = I$.

Passo 3. Calcule

$$\bar{x}_\rho^k = \underset{s.a. \ x \in \Omega}{\operatorname{arg\,min}} \ Q_\rho^k(x) \quad (3.6)$$

em que Q_ρ^k denota a forma quadrática definida em (3.4) com $y = x_k$, $B = B_\rho^k$ e $A = A_\rho^k$.

Passo 4. Calcule $x_\rho^k \in \Omega$ tal que $Q_\rho^k(x_\rho^k) \leq \eta Q_\rho^k(\bar{x}_\rho^k)$.

Se $Q_\rho^k(x_\rho^k) = 0$, declare x^k um ponto estacionário de (3.1).

Passo 5. Defina

$$\Psi_k(x) = \langle g(x_k), x - x_k \rangle + \frac{1}{2}(x - x_k)^T B_\rho^k(x - x_k), \quad (3.7)$$

$$\operatorname{ared}_\rho^k = f(x_{l(k)}) - f(x_\rho^k) \quad \text{e} \quad \operatorname{pred}_\rho^k = -\Psi_k(x_\rho^k).$$

Se

$$\frac{\operatorname{ared}_\rho^k}{\operatorname{pred}_\rho^k} \geq \beta_1, \quad (3.8)$$

defina $\rho_k = \rho$, $x_{k+1} = x_\rho^k$, faça $k = k + 1$ e volte ao Passo 1.

Senão, escolha $\rho_{\text{novo}} \in [\xi_1 \rho, \xi_2 \rho]$, faça $\rho = \rho_{\text{novo}}$ e volte ao Passo 2.

O modelo definido em (3.7) é padrão em métodos de região de confiança. A este modelo acrescentamos um termo de regularização, com o objetivo de controlar a proximidade entre x_{k+1} e x_k , obtendo assim (3.4).

No lema a seguir mostramos que o algoritmo anterior está bem definido, ou seja, que é necessário apenas um número finito de aumentos de ρ .

Lema 3.1 *Se $x_k \in \Omega$ não é um ponto estacionário de (3.1) e $\rho > \rho_{\text{lim}}$, então a condição (3.8) é satisfeita.*

Demonstração: Por hipótese, $\rho > \rho_{\text{lim}}$. Então, do Passo 2 do Algoritmo 3.1 segue que $B_\rho^k = L_u I$ e $A_\rho^k = I$. Assim,

$$Q_\rho^k(x) = \langle g(x_k), x - x_k \rangle + \frac{(L_u + \rho)}{2} \|x - x_k\|^2$$

e,

$$\Psi_k(x) = \langle g(x_k), x - x_k \rangle + \frac{L_u}{2} \|x - x_k\|^2.$$

Substituindo y por x_ρ^k e x por x_k em (3.3) e como $L_f \leq L_u$, tem-se

$$f(x_\rho^k) \leq f(x_k) + \Psi_k(x_\rho^k) \leq f(x_{l(k)}) + \Psi_k(x_\rho^k).$$

Além disso, $\Psi_k(x_\rho^k) < Q_\rho^k(x_\rho^k) \leq \eta Q_\rho^k(\bar{x}_\rho^k) \leq 0$, pois $Q_\rho^k(x_k) = 0$ e \bar{x}_ρ^k é solução global de (3.6). Dessa forma, $\Psi_k(x_\rho^k) < 0$ e como $\beta_1 \in (0, \frac{1}{2}]$, temos

$$f(x_\rho^k) \leq f(x_{l(k)}) + \beta_1 \Psi_k(x_\rho^k),$$

ou seja,

$$f(x_{l(k)}) - f(x_\rho^k) \geq -\beta_1 \Psi_k(x_\rho^k),$$

isto é,

$$ared_\rho^k \geq \beta_1 pred_\rho^k.$$

Portanto, a condição (3.8) é satisfeita sempre que $\rho > \rho_{lim}$. \square

3.2 ANÁLISE DE CONVERGÊNCIA

Nesta seção vamos discutir a convergência da sequência gerada pelo Algoritmo 3.1. Em particular, mostraremos que o algoritmo discutido apresenta convergência global, isto é, a sequência dos iterados converge para um ponto estacionário do problema (3.1) independentemente da escolha do ponto inicial x_0 .

Para tanto, além da condição (3.2), assumiremos os seguintes fatos:

H1: $\Omega_0 = \{x \in \Omega : f(x) \leq f(x_0)\}$ é um subconjunto limitado;

H2: Os subconjuntos \mathcal{S}_n e \mathcal{S}_n^+ são uniformemente limitados, isto é, existe $\overline{M} > 0$ tal que $\|B_\rho^k\|_F \leq \overline{M}$ e $\|A_\rho^k\|_F \leq \overline{M}$, para todos k e ρ ;

A hipótese H1 é típica de métodos de otimização e a hipótese H2 pode ser controlada no algoritmo.

Lema 3.2 *Se $A \in \mathcal{S}_n^+$, então A é simétrica definida positiva e*

$$\gamma \|x\|^2 \leq x^T A x \leq \overline{M} \|x\|^2,$$

para todo $x \in \mathbb{R}^n$.

Demonstração: De $A \in \mathcal{S}_n^+$ segue que A é simétrica e $x^T A x > 0$, para

todo $x \in \mathbb{R}^n$. Então, A é simétrica definida positiva e existe $\gamma > 0$ tal que $x^T A x \geq \gamma \|x\|^2$.

Por outro lado, $x^T A x = |\langle x, Ax \rangle| \leq \|A\| \|x\|^2 \leq \overline{M} \|x\|^2$, para todo $x \in \mathbb{R}^n$. O que conclui a demonstração. \square

O lema a seguir mostra que no Passo 4 do Algoritmo Modelo é coerente declarar x_k como sendo ponto estacionário de (3.1).

Lema 3.3 *Se $Q_\rho^k(x_\rho^k) = 0$, isto é, se o Algoritmo 3.1 parou no Passo 4, então x_k é ponto estacionário do problema*

$$\begin{aligned} \min \quad & f(x) \\ \text{s.a.} \quad & x \in \Omega. \end{aligned}$$

Ainda, se Ω é convexo e x_k é ponto estacionário de (3.1), então existe $\tilde{\rho} > 0$ tal que para todo $\rho \geq \tilde{\rho}$, $Q_\rho^k(x_\rho^k) = 0$.

Demonstração: Como $Q_\rho^k(x_\rho^k) = 0$ e \bar{x}_ρ^k é solução global de (3.6), tem-se que $Q_\rho^k(\bar{x}_\rho^k) \leq 0$. Além disso,

$$0 = Q_\rho^k(x_\rho^k) \leq \eta Q_\rho^k(\bar{x}_\rho^k) \leq 0,$$

ou seja,

$$Q_\rho^k(\bar{x}_\rho^k) = 0.$$

Então, x_k também é solução de (3.6), pois $Q_\rho^k(x_k) = 0$.

Portanto, $\langle \nabla Q_\rho^k(x_k), \alpha'(0) \rangle \geq 0$, para toda curva viável α partindo de x_k . Como $\nabla Q_\rho^k(x_k) = g(x_k)$, segue que $\langle g(x_k), \alpha'(0) \rangle \geq 0$, para toda curva viável α partindo de x_k , ou seja, x_k é ponto estacionário de (3.1).

Considere agora Ω convexo e x_k ponto estacionário de (3.1).

Seja α curva viável partindo de x_k , então $\langle g(x_k), \alpha'(0) \rangle \geq 0$. Além disso, como visto anteriormente, $Q_\rho^k(x_\rho^k) \leq 0$. Então,

$$0 \geq Q_\rho^k(x_\rho^k) \geq \frac{1}{2} (x_\rho^k - x_k)^T (B_\rho^k + \rho A_\rho^k) (x_\rho^k - x_k) \geq 0,$$

para todo $\rho \geq \tilde{\rho}$, em que $\tilde{\rho}$ é tal que $(B_\rho^k + \tilde{\rho} A_\rho^k) > 0$.

Logo, $Q_\rho^k(x_\rho^k) = 0$, para todo $\rho \geq \tilde{\rho}$. \square

Dessa forma, ou o Algoritmo 3.1 termina no Passo 4 em um ponto

estacionário do problema, ou gera uma sequência infinita de iterados. Assumiremos, sem perda de generalidade, que $\{x_k\}_{k \in \mathbb{N}}$ é uma sequência e mostraremos que a mesma admite uma subsequência convergente para um ponto estáacionário de (3.1).

Os lemas técnicos a seguir são relevantes para a demonstração do resultado de convergência global da sequência de iterados gerada pelo Algoritmo Modelo.

Lema 3.4 *A sequência $\{f(x_{l(k)})\}_{k \in \mathbb{N}}$ é monótona não crescente.*

Demonstração: Note que,

$$\begin{aligned} f(x_{l(k+1)}) &= \max\{f(x_{k+1-j}) : 0 \leq j \leq m(k+1)\} \\ &\leq \max\{f(x_{k+1-j}) : 0 \leq j \leq m(k)+1\} \\ &= \max\{f(x_{k+1}), f(x_k), \dots, f(x_{k-m(k)})\} \\ &= \max\{f(x_{k+1}), f(x_{l(k)})\}. \end{aligned}$$

Por outro lado, como $Q_\rho^k(x_{k+1}) = Q_\rho^k(x_k) \leq 0$ e $\Psi_k(x_{k+1}) < Q_\rho^k(x_{k+1})$, temos que $\Psi_k(x_{k+1}) < 0$. Assim, de

$$\frac{ared_\rho^k}{pred_\rho^k} = \frac{f(x_{l(k)}) - f(x_{k+1})}{-\Psi_k(x_{k+1})} \geq \beta_1 > 0,$$

segue que, $f(x_{k+1}) < f(x_{l(k)})$.

Portanto, $f(x_{l(k+1)}) \leq f(x_{l(k)})$ e o resultado segue. □

Lema 3.5 *A sequência $\{x_k\}_{k \in \mathbb{N}}$ é limitada.*

Demonstração: Na demonstração do lema anterior vimos que $f(x_{k+1}) < f(x_{l(k)})$, usaremos este fato fortemente nessa demonstração para provar, por indução, que $\{x_k\} \subseteq \Omega_0$.

Para $k = 0$, note que $f(x_1) < f(x_{l(0)}) = f(x_0)$, ou seja, $x_1 \in \Omega_0$. Suponha válido para todo $i \in \{0, 1, \dots, k\}$, então como $l(k) \leq k$ segue que, $f(x_{k+1}) < f(x_{l(k)}) \leq f(x_0)$. Pelo Segundo Princípio da Indução, $\{x_k\} \subseteq \Omega_0$. Como assumimos Ω_0 limitado, segue que a sequência é limitada. □

Portanto, pelo Lema 3.5, podemos assumir que $\{x_k\}_{k \in \mathbb{N}}$ admite subsequência convergente, ou seja, um ponto de acumulação.

Lema 3.6 *A sequência $\{\rho_k\}_{k \in \mathbb{N}}$ é limitada.*

Demonstração: Pelo Lema 3.1 para todo $\rho > \rho_{lim}$ a condição (3.8) é satisfeita, ou seja, o Passo 5 do algoritmo termina em um número finito de ciclos. Então, o maior valor possível para ρ no Passo 2 é $\max\{\xi_2 \rho_{lim}, \xi_2 \rho_{max}\}$ e, conseqüentemente

$$0 \leq \rho_k \leq \max\{\xi_2 \rho_{lim}, \xi_2 \rho_{max}\},$$

para todo $k \in \mathbb{N}$. □

Lema 3.7 *Seja $\mathcal{K} \subseteq \infty \mathbb{N}$ tal que $\lim_{k \in \mathcal{K}} \Psi_k(x_{k+1}) = 0$. Então, $\lim_{k \in \mathcal{K}} \|x_{k+1} - x_k\| = 0$.*

Demonstração: Vimos anteriormente que $Q_{\rho_k}^k(x_{k+1}) \leq 0$, ou seja,

$$\langle g(x_k), x_{k+1} - x_k \rangle + \frac{1}{2}(x_{k+1} - x_k)^T (B_{\rho_k}^k + \rho_k A_{\rho_k}^k)(x_{k+1} - x_k) \leq 0,$$

ou ainda,

$$\begin{aligned} & \frac{1}{2} \rho_k (x_{k+1} - x_k)^T A_{\rho_k}^k (x_{k+1} - x_k) \leq \\ & - \left[\langle g(x_k), x_{k+1} - x_k \rangle + \frac{1}{2} (x_{k+1} - x_k)^T B_{\rho_k}^k (x_{k+1} - x_k) \right]. \end{aligned} \quad (3.9)$$

Pelo Lema 3.2 temos que

$$0 < \gamma \|x_{k+1} - x_k\|^2 \leq (x_{k+1} - x_k)^T A_{\rho_k}^k (x_{k+1} - x_k).$$

Então, usando (3.9) segue que

$$\begin{aligned} 0 & < \frac{1}{2} \rho_{min} \gamma \|x_{k+1} - x_k\|^2 \leq \frac{1}{2} \rho_{min} (x_{k+1} - x_k)^T A_{\rho_k}^k (x_{k+1} - x_k) \\ & \leq \frac{1}{2} \rho_k (x_{k+1} - x_k)^T A_{\rho_k}^k (x_{k+1} - x_k) \\ & \leq - \left[\langle g(x_k), x_{k+1} - x_k \rangle + \frac{1}{2} (x_{k+1} - x_k)^T B_{\rho_k}^k (x_{k+1} - x_k) \right] \\ & = -\Psi_k(x_{k+1}). \end{aligned}$$

Tomando o limite quando $k \rightarrow \infty$ restrito a $k \in \mathcal{K}$ na desigualdade acima, conclui-se que

$$\lim_{k \in \mathcal{K}} \|x_{k+1} - x_k\| = 0.$$

□

Os dois lemas seguintes são fundamentais para a prova do teorema de convergência enunciado mais a frente.

Lema 3.8 *Seja $\mathcal{K}_1 \subseteq \infty \{l(k)\}_{k \in \mathbb{N}}$. Então, para qualquer $j \in \mathbb{N}$,*

$$\lim_{k \in \mathcal{K}_1} \|x_{k-j+1} - x_{k-j}\| = 0 \text{ e } \lim_{k \in \mathcal{K}_1} f(x_{k-j}) = \lim_{k \rightarrow \infty} f(x_{l(k)}). \quad (3.10)$$

Demonstração: Faremos a prova de acordo com Princípio da Indução sobre j .

Segue dos Lemas 3.4 e 3.5 que a sequência $\{f(x_{l(k)})\}_{k \in \mathbb{N}}$ é convergente. Então,

$$\lim_{k \in \mathcal{K}_1} f(x_k) = \lim_{k \rightarrow \infty} f(x_{l(k)}).$$

Além disso, da definição de redução real, $ared_{\rho}^k$, dada no Passo 5 do Algoritmo Modelo segue que, para todo $k \in \mathcal{K}_1$

$$\begin{aligned} f(x_k) &= f(x_{\rho_{k-1}}^{k-1}) = f(x_{l(k-1)}) - ared_{\rho_{k-1}}^{k-1} \leq f(x_{l(k-1)}) - \beta_1 pred_{\rho_{k-1}}^{k-1} \\ &= f(x_{l(k-1)}) + \beta_1 \Psi_{k-1}(x_{\rho_{k-1}}^{k-1}) = f(x_{l(k-1)}) + \beta_1 \Psi_{k-1}(x_k), \end{aligned}$$

ou seja,

$$\Psi_{k-1}(x_k) \geq \frac{1}{\beta_1} [f(x_k) - f(x_{l(k-1)})].$$

Tomando limite quando $k \rightarrow \infty$, $k \in \mathcal{K}_1$ e lembrando que $\Psi_{k-1}(x_k) < 0$, temos

$$0 \geq \lim_{k \in \mathcal{K}_1} \Psi_{k-1}(x_k) \geq \lim_{k \in \mathcal{K}_1} \frac{1}{\beta_1} [f(x_k) - f(x_{l(k-1)})] = 0,$$

isto é,

$$\lim_{k \in \mathcal{K}_1} \Psi_{k-1}(x_k) = 0.$$

Pelo Lema 3.7 conclui-se que $\lim_{k \in \mathcal{K}_1} \|x_k - x_{k-1}\| = 0$, e como f é uniformemente contínua em Ω_0 ,

$$\lim_{k \in \mathcal{K}_1} f(x_{k-1}) = \lim_{k \in \mathcal{K}_1} f(x_k) = \lim_{k \rightarrow \infty} f(x_{l(k)}).$$

Agora, assumindo que (3.10) vale para um j fixo, vamos mostrar que vale para $j + 1$.

Analogamente ao que foi feito para $j = 0$, e usando a hipótese de indução, conclui-se que

$$\lim_{k \in \mathcal{K}_1} \Psi_{k-j-1}(x_{k-j}) = 0.$$

Então, pelo Lema 3.7, $\lim_{k \in \mathcal{K}_1} \|x_{k-j} - x_{k-j-1}\| = 0$. Também, da continuidade uniforme da função f em Ω_0 e da hipótese de indução

$$\lim_{k \in \mathcal{K}_1} f(x_{k-j-1}) = \lim_{k \in \mathcal{K}_1} f(x_{k-j}) = \lim_{k \rightarrow \infty} f(x_{l(k)}).$$

Portanto, pelo Primeiro Princípio da Indução, o resultado segue. \square

Lema 3.9 $\lim_{k \rightarrow \infty} f(x_{k+1}) = \lim_{k \rightarrow \infty} f(x_{l(k)})$.

Demonstração: Defina $\widehat{l}(k) = l(k + M + 2)$ e $\widehat{\mathcal{K}} = \{\widehat{l}(k)\}_{k \in \mathbb{N}} \subseteq \infty$ $\{l(k)\}_{k \in \mathbb{N}}$. Note que podemos escrever

$$x_{k+1} = x_{\widehat{l}(k)} - \sum_{j=1}^{\widehat{l}(k)-k-1} \left(x_{\widehat{l}(k)-j+1} - x_{\widehat{l}(k)-j} \right).$$

Então,

$$\|x_{k+1} - x_{\widehat{l}(k)}\| \leq \sum_{j=1}^{\widehat{l}(k)-k-1} \|x_{\widehat{l}(k)-j+1} - x_{\widehat{l}(k)-j}\|.$$

Como $\widehat{l}(k) - k - 1 \leq M + 1$, segue que o somatório acima percorre um conjunto finito de índices. Então, tomando o limite quando $k \rightarrow \infty$ e usando o resultado do Lema 3.8, temos

$$0 \leq \lim_{k \rightarrow \infty} \|x_{k+1} - x_{\widehat{l}(k)}\| \leq \sum_{j=1}^{\widehat{l}(k)-k-1} \|x_{\widehat{l}(k)-j+1} - x_{\widehat{l}(k)-j}\| = 0,$$

ou seja,

$$\lim_{k \rightarrow \infty} \|x_{k+1} - x_{\widehat{l}(k)}\| = 0. \quad (3.11)$$

Logo, pela continuidade uniforme da f em Ω_0 ,

$$\lim_{k \rightarrow \infty} f(x_{k+1}) = \lim_{k \rightarrow \infty} f(x_{\widehat{l}(k)}) = \lim_{k \rightarrow \infty} f(x_{l(k)}).$$

□

O próximo teorema, o resultado mais importante do capítulo, estabelece a convergência global da sequência gerada pelo Algoritmo Modelo.

Teorema 3.10 *Seja x^* um ponto de acumulação da sequência gerada pelo Algoritmo (3.1). Então, x^* é um ponto estacionário do problema*

$$\begin{aligned} \min \quad & f(x) \\ \text{s.a.} \quad & x \in \Omega \end{aligned}$$

que não é ponto de máximo local. Ainda mais, se o conjunto dos pontos estacionários do problema for finito, então a sequência converge.

Demonstração: Seja x^* um ponto de acumulação da sequência $\{x_k\}_{k \in \mathbb{N}}$ gerada pelo Algoritmo 3.1. Então, existe $\mathcal{K} \subseteq \infty \mathbb{N}$ tal que $\lim_{k \in \mathcal{K}} x_k = x^*$.

Pelo Lema 3.6, existe $\bar{\rho}$ tal que $\rho_k \leq \bar{\rho}$, para todo $k \in \mathcal{K}$. Além disso, como assumimos \mathcal{S}_n e \mathcal{S}_n^+ uniformemente limitados, pelo Teorema de Bolzano-Weierstrass existe $\mathcal{K}_1 \subseteq \mathcal{K}$ tal que $\lim_{k \in \mathcal{K}_1} A_{\rho_k}^k = \bar{A}$ e $\lim_{k \in \mathcal{K}_1} B_{\rho_k}^k = \bar{B}$. Também, para todo $k \in \mathcal{K}_1$,

$$f(x_{k+1}) \leq f(x_{l(k)}) + \beta_1 \Psi_k(x_{k+1}),$$

e, pelo Lema 3.9,

$$\lim_{k \in \mathcal{K}_1} \Psi_k(x_{k+1}) = 0.$$

Ainda, como $\Psi_k(x_{k+1}) \leq Q_{\rho_k}^k(x_{\rho_k}^k) \leq \eta Q_{\rho_k}^k(\bar{x}_{\rho_k}^k) \leq 0$, segue que $\lim_{k \in \mathcal{K}_1} Q_{\rho_k}^k(\bar{x}_{\rho_k}^k) = 0$.

Defina

$$Q^*(x) = \langle g(x^*), x - x^* \rangle + \frac{1}{2}(x - x^*)^T (\bar{B} + \bar{\rho} \bar{A})(x - x^*),$$

e seja

$$\hat{x} = \underset{\text{s.a. } x \in \Omega}{\arg \min} Q^*(x)$$

Então,

$$\begin{aligned}
0 &\geq Q^*(\hat{x}) = \langle g(x^*), \hat{x} - x^* \rangle + \frac{1}{2}(\hat{x} - x^*)^T (\overline{B} + \overline{\rho A})(\hat{x} - x^*) \\
&\geq \lim_{k \in \mathcal{K}_1} \left[\langle g(x_k), \hat{x} - x_k \rangle + \frac{1}{2}(\hat{x} - x_k)^T (B_{\rho_k}^k + \rho_k A_{\rho_k}^k)(\hat{x} - x_k) \right] \\
&= \lim_{k \in \mathcal{K}_1} Q_{\rho_k}^k(\hat{x}) \geq \lim_{k \in \mathcal{K}_1} Q_{\rho_k}^k(\overline{x}_{\rho_k}^k) = 0,
\end{aligned}$$

ou seja, $Q^*(\hat{x}) = 0 = Q^*(x^*)$. Logo, x^* também é um minimizador de Q^* em Ω .

Como $\nabla Q^*(x) = g(x^*) + (\overline{B} + \overline{\rho A})(x - x^*)$, segue que $\nabla Q^*(x^*) = g(x^*)$. Então, $\langle g(x^*), \alpha'(0) \rangle = 0$ para toda curva viável α partindo de x^* e, portanto, x^* é ponto estacionário de

$$\begin{aligned}
&\min f(x) \\
&s.a. \quad x \in \Omega.
\end{aligned}$$

Vamos mostrar agora que x^* não é ponto de máximo local de (3.1). Para tanto, suponha por contradição que

$$f(x^*) \geq f(x) \tag{3.12}$$

para todo $x \in B(x^*, \varepsilon)$, $\varepsilon > 0$.

Por (3.11), segue que existe $\mathcal{K}_3 \subseteq^\infty \mathcal{K}$ tal que $\lim_{x \in \mathcal{K}_3} x = x^*$.

Dessa forma, como f é uniformemente contínua em Ω e $\{f(x_{l(k)})\}_{k \in \mathbb{N}}$ é monótona não crescente, tem-se

$$\lim_{k \rightarrow \infty} f(x_{l(k)}) = \lim_{k \in \mathcal{K}_3} f(x_{l(k)}) = f(x^*)$$

e

$$f(x_{l(k)}) \geq f(x^*), \quad \forall k \in \mathbb{N}. \tag{3.13}$$

Além disso, usando a condição (3.8) mostra-se que $f(x_{l(k)}) < f(x_{l(l(k)-1)})$ e, conseqüentemente, $f(x_{l(k')}) < f(x_{l(k)})$, para todo $k' > k + M + 1$. Então, para k suficientemente grande, em qualquer vizinhança de x^* podemos encontrar $x_{l(k)}$ tal que $f(x_{l(k)}) > f(x_{l(k')})$, e por (3.13) tem-se

$$f(x_{l(k)}) > f(x_{l(k')}) \geq f(x^*),$$

o que contradiz (3.12).

Para concluir a demonstração, resta mostrarmos que se o conjunto dos pontos estacionários do problema (3.1) é finito, então a sequência gerada pelo Algoritmo Modelo converge.

Seja Z^* o conjunto dos pontos de acumulação de $\{x_k\}_{k \in \mathbb{N}}$. Já foi provado que cada ponto de acumulação dessa sequência é também um ponto estacionário do problema. Logo, Z^* é finito. Digamos, $Z^* = \{z_1^*, z_2^*, \dots, z_m^*\}$.

Então, $\delta = \min\{\|z_i^* - z_j^*\| : i \neq j, i, j = 1, \dots, m\} > 0$, e existe $k_0 \geq 0$ tal que

$$x_k \in \bigcup_{i=1}^m B\left(z_i^*, \frac{\delta}{4}\right) \quad \text{e} \quad \|x_k - x_{k+1}\| \leq \frac{\delta}{4}, \quad \forall k \geq k_0,$$

pois cada z_i^* é ponto de acumulação e vale o Lema 3.7.

Seja $k_1 \geq k_0$ tal que $x_{k_1} \in B(z_1^*, \frac{\delta}{4})$. Então, para $i \geq 2$

$$\|z_i^* - z_1^*\| \leq \|z_i^* - x_{k_1+1}\| + \|x_{k_1+1} - x_{k_1}\| + \|x_{k_1} - z_1^*\|,$$

ou seja,

$$\|z_i^* - x_{k_1+1}\| \geq \|z_i^* - z_1^*\| - \|x_{k_1+1} - x_{k_1}\| - \|x_{k_1} - z_1^*\| \geq \frac{\delta}{4}.$$

Segue que $x_{k_1+1} \notin B(z_i^*, \frac{\delta}{4})$, $i \geq 2$ e, conseqüentemente $x_{k_1+1} \in B(z_1^*, \frac{\delta}{4})$.

Por um argumento indutivo, concluímos que $x_k \in B(z_1^*, \frac{\delta}{4})$, para todo $k \geq k_1$. Segue que $m = 1$ e, portanto, a sequência $\{x_k\}_{k \in \mathbb{N}}$ é convergente.

□

O próximo capítulo descreve algumas aplicações do Algoritmo 3.1 que serão essenciais para validar a teoria desenvolvida neste capítulo.

4 APLICAÇÃO EM MINIMIZAÇÃO COM RESTRIÇÕES DE ORTOGONALIDADE

O problema de minimização com respeito ao conjunto das matrizes com colunas ortonormais aparece frequentemente em importantes classes de problemas de otimização e, assim como as restrições linear e quadrática, possui uma estrutura especial a ser explorada. Como exemplo, citamos aplicações em otimização polinomial [12], otimização combinatorial [9], problemas de autovalores e análise de componentes principais (PCA) esparsos [2]. De acordo com Wen e Yin [44] é, em geral, um problema do tipo NP-hard. Além disso as restrições de ortogonalidade podem conduzir a muitos mínimos locais, não há garantia de convergência para o mínimo global, exceto em casos mais simples (como encontrar os autovalores extremos de uma matriz) e gerar uma sequência de pontos viáveis não é fácil, visto que preservar as restrições de ortogonalidade numericamente é muito caro, em geral, envolve uma decomposição SVD.

Seja $h : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{n \times n}$, $m \geq n$, definida por:

$$h(X) = X^T X - I,$$

e considere $\Omega = \{X \in \mathbb{R}^{m \times n} : h(X) = 0\}$.

O conjunto viável Ω é conhecido como variedade de Stiefel, em homenagem ao matemático suíço Eduard Stiefel (1909-1978), quem primeiro estudou sua topologia [40].

Neste capítulo aplicaremos o algoritmo apresentado no capítulo anterior, para resolver o seguinte problema de minimização com restrições de ortogonalidade,

$$\begin{aligned} \min \quad & f(X) \\ \text{s.a.} \quad & X \in \Omega, \end{aligned} \tag{4.1}$$

em que $f : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ é uma função de classe \mathcal{C}^2 .

Note que,

$$g(X) = \left(\frac{\partial f}{\partial X_1}(X) \quad \cdots \quad \frac{\partial f}{\partial X_n}(X) \right) \in \mathbb{R}^{m \times n},$$

em que X_i é a i -ésima coluna de X . Além disso, como $\|X\|_F = \sqrt{n}$, para todo $X \in \Omega$, segue que Ω é compacto e g é Lipschitz contínua em Ω , ou seja, a condição (3.2) é satisfeita para o problema (4.1). Note ainda que, apesar do problema (4.1) estar colocado em um contexto

matricial, o mesmo pode ser reescrito sob a forma de mn variáveis reais e, portanto, podemos empregar o Algoritmo 3.1 para resolvê-lo.

Em um esquema de otimização, é importante verificar a otimalidade de cada iterado para então decidir continuar o processo ou pará-lo em um ponto suficientemente próximo da solução. Nesse sentido, a seguinte seção estabelece as condições de otimalidade para o problema (4.1).

4.1 CONDIÇÕES DE OTIMALIDADE PARA MINIMIZAÇÃO COM RESTRIÇÕES DE ORTOGONALIDADE

Nesta seção, vamos caracterizar as Condições KKT para o problema (4.1), utilizando projeção ortogonal no espaço tangente de Ω .

Antes de iniciarmos a discussão, note que

$$\begin{aligned} h(X + tY) &= (X + tY)^T(X + tY) - I \\ &= X^T X - I + tX^T Y + tY^T X + t^2 Y^T Y \\ &= h(X) + t(X^T Y + Y^T X) + t^2 Y^T Y \end{aligned} \quad (4.2)$$

Por outro lado, pelo Polinômio de Taylor,

$$h(X + tY) = h(X) + t\langle \nabla h(X), Y \rangle + o(t^2). \quad (4.3)$$

De (4.2) e (4.3) segue que

$$\langle \nabla h(X), Y \rangle = X^T Y + Y^T X.$$

Portanto, $N(h'(X)) = \{Y \in \mathbb{R}^{m \times n} : X^T Y = -Y^T X\}$.

Lema 4.1 *Sejam $A, X \in \mathbb{R}^{n \times n}$. Então, $\hat{X} = \frac{A - A^T}{2}$ é solução do problema*

$$\begin{aligned} \min \quad & \|A - X\|_F^2 \\ \text{s.a.} \quad & X = -X^T. \end{aligned}$$

Demonstração: Seja $X \in \mathbb{R}^{n \times n}$ tal que $X^T = -X$. Então,

$$\begin{aligned}
\|A - \widehat{X}\|_F^2 &= \left\| A - \frac{A + A^T}{2} \right\|_F^2 = \left\| \frac{A + A^T}{2} \right\|_F^2 \\
&= \left\| \frac{A + A^T}{2} - \frac{X + X^T}{2} \right\|_F^2 = \left\| \frac{A - X}{2} + \frac{(A - X)^T}{2} \right\|_F^2 \\
&\leq \frac{1}{4} \|A - X\|_F^2 + \frac{1}{4} \|(A - X)^T\|_F^2 = \frac{1}{2} \|A - X\|_F^2 \\
&\leq \|A - X\|_F^2,
\end{aligned}$$

o que conclui a demonstração. \square

Lema 4.2 *Sejam $Y \in \Omega$ e $A \in \mathbb{R}^{m \times n}$. Então, $\widehat{X} = A - \frac{Y(Y^T A + A^T Y)}{2}$ é solução de*

$$\begin{aligned}
&\min \|A - X\|_F^2 \\
&s.a. \quad X \in N(h'(Y)).
\end{aligned}$$

Demonstração: Seja $Y^\perp \in \mathbb{R}^{m \times (m-n)}$ uma matriz com colunas ortogonais tal que

$$V = \begin{pmatrix} Y & Y^\perp \end{pmatrix} \in \mathbb{R}^{m \times m}$$

é ortogonal, isto é,

$$VV^T = \begin{pmatrix} Y & Y^\perp \end{pmatrix} \begin{pmatrix} Y^T \\ (Y^\perp)^T \end{pmatrix} = YY^T + Y^\perp(Y^\perp)^T = I.$$

Seja I_n a matriz identidade de ordem $n \times n$. Definindo $\widetilde{I} = \begin{pmatrix} I_n \\ 0 \end{pmatrix} \in \mathbb{R}^{m \times n}$, temos que

$$V\widetilde{I} = \begin{pmatrix} Y & Y^\perp \end{pmatrix} \begin{pmatrix} I_n \\ 0 \end{pmatrix} = Y.$$

Então,

$$\begin{aligned}
X \in N(h'(Y)) &\Leftrightarrow X^T Y + Y^T X = 0 \\
&\Leftrightarrow X^T V\widetilde{I} + (V\widetilde{I})^T X = 0 \\
&\Leftrightarrow X^T V\widetilde{I} + \widetilde{I}^T V^T X = 0.
\end{aligned} \tag{4.4}$$

Escreva $Z = \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} = V^T X$, com $Z_1 \in \mathbb{R}^{n \times n}$ e $Z_2 \in \mathbb{R}^{(m-n) \times n}$
e, assim

$$\begin{aligned} \tilde{I}^T V^T X + X^T V \tilde{I} &= 0 \\ \Leftrightarrow \tilde{I}^T \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} + \begin{pmatrix} Z_1^T & Z_2^T \end{pmatrix} \tilde{I} &= 0 \\ \Leftrightarrow Z_1 + Z_1^T &= 0 \quad \Leftrightarrow \quad Z_1 = -Z_1^T. \end{aligned}$$

Logo, (4.4) é equivalente a $Z_1 = -Z_1^T$, $Z_1 \in \mathbb{R}^{n \times n}$ e $Z_2 \in \mathbb{R}^{(m-n) \times n}$.

Além disso,

$$\begin{aligned} \|A - X\|_F^2 &= \|V^T(A - X)\|_F^2 = \|V^T A - Z\|_F^2 = \\ &= \left\| \begin{pmatrix} Y^T \\ (Y^\perp)^T \end{pmatrix} A - \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} \right\|_F^2 = \|Y^T A - Z_1\|_F^2 + \|(Y^\perp)^T A - Z_2\|_F^2. \end{aligned}$$

Portanto,

$$\begin{aligned} \min \quad & \|A - X\|_F^2 \\ \text{s.a.} \quad & X \in N(h'(Y)) \end{aligned}$$

é equivalente a

$$\begin{aligned} \min \quad & (\|Y^T A - Z_1\|_F^2 + \|(Y^\perp)^T A - Z_2\|_F^2) \\ \text{s.a.} \quad & Z_1 = -Z_1^T, Z_1 \in \mathbb{R}^{n \times n} \\ & Z_2 \in \mathbb{R}^{(m-n) \times n}. \end{aligned} \tag{4.5}$$

Note que a função objetivo em (4.5) é separável com relação à Z_1 e Z_2 , e portanto, pelo Lema 4.1 sua solução é $\widehat{Z}_1 = \frac{Y^T A - A^T Y}{2}$ e $\widehat{Z}_2 = (Y^\perp)^T A$. Agora, como Y tem colunas ortogonais e $X = VZ$, temos que

$$\begin{aligned} \widehat{X} &= V\widehat{Z} = \begin{pmatrix} Y & Y^\perp \end{pmatrix} \begin{pmatrix} \widehat{Z}_1 \\ \widehat{Z}_2 \end{pmatrix} = Y\widehat{Z}_1 + Y^\perp\widehat{Z}_2 \\ &= Y\widehat{Z}_1 + Y^\perp(Y^\perp)^T A = Y \frac{Y^T A - A^T Y}{2} + (I - YY^T)A \\ &= A - \frac{Y(Y^T A + A^T Y)}{2} \end{aligned}$$

o que conclui a demonstração. □

O próximo teorema estabelece uma equivalência entre as Condições KKT do problema (4.1) e a projeção sobre o espaço tangente de Ω , o que simplifica consideravelmente a verificação de ponto estacionário e, conseqüentemente, a decisão de continuar ou parar o algoritmo em um ponto estacionário aproximado.

Teorema 4.3 *Seja $X_* \in \Omega$. São equivalentes:*

- (i) $X_*(X_*^T g(X_*) + g(X_*)^T X_*) - 2g(X_*) = 0$;
- (ii) X_* satisfaz as Condições KKT de (4.1);
- (iii) $g(X_*) = X_* S$, para alguma matriz $S \in \mathbb{R}^{n \times n}$ simétrica.

Demonstração: Já vimos que as Condições KKT para um problema com restrições de igualdade são

$$g(X_*) - h'(X_*)^T \lambda^* = 0,$$

isto é,

$$g(X_*) = h'(X_*)^T \lambda^*.$$

Logo, as Condições KKT de (4.1) são equivalentes à $g(X_*) \in R(h'(X_*)^T)$, ou seja, $g(X_*) \perp N(h'(X_*))$, isto é, a projeção de $g(X_*)$ se anula em $N(h'(X_*))$.

Além disso, pelo Lema 4.2, temos que

$$\widehat{V} = \arg \min_{V \in N(h'(X_*))} \|g(X_*) - V\|_F^2 = g(X_*) - \frac{X_*(X_*^T g(X_*) + g(X_*)^T X_*)}{2}$$

Como $g(X_*) \perp N(h'(X_*))$, segue que $\widehat{V} = 0$ e, portanto, (i) é equivalente a (ii).

Para mostrarmos a equivalência entre (ii) e (iii), note que a função Lagrangeana de (4.1) é

$$l(X, \Theta) = f(X) + \text{traço}(\Theta(X^T X - I)),$$

em que, $\Theta \in \mathbb{R}^{n \times n}$ contém os multiplicadores de Lagrange do problema.

Logo, $\nabla_X l(X, \Theta) = 0$ implica em $g(X) + X(\Theta + \Theta^T) = 0$ e, portanto, (ii) é equivalente a (iii) com $S = -(\Theta + \Theta^T)$. □

Devido à equivalência $(i) \Leftrightarrow (ii)$ no teorema acima, a igualdade

$$\|X_*(X_*^T g(X_*) + g(X_*)^T X_*) - 2g(X_*)\|_F = 0,$$

será utilizada como critério de parada para o algoritmo colocado na seção seguinte. Esta condição, por necessitar apenas de multiplicações entre matrizes, é mais simples de verificar do que calcular os multiplicadores de Lagrange do problema.

4.2 UM ALGORITMO PARA MINIMIZAÇÃO COM RESTRIÇÕES DE ORTOGONALIDADE

Nesta seção aplicaremos o Algoritmo 3.1 para resolver o problema (4.1). O resultado será um método do tipo gradiente projetado, globalmente convergente, para resolver problemas de minimização com restrições de ortogonalidade.

Para tanto, vamos tomar constantes $0 < \sigma_{min} < \sigma_{max} < \infty$, $L_u \geq L_f^1$, para cada $k = 0, 1, 2, \dots$, escolher $\sigma_{spc}^k \in [\sigma_{min}, \sigma_{max}]$, e definir

$$\sigma_\rho^k = \begin{cases} \frac{\sigma_{spc}^k}{2}, & \text{se } 0 < \rho < L_u \\ L_u, & \text{caso contrário} \end{cases} \quad (4.6)$$

$A_\rho^k = I$ e $B_\rho^k = \sigma_\rho^k I$.

Dessa forma, o modelo quadrático $Q_\rho^k : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ é dado por:

$$Q_\rho^k(X) = \text{traço}(g(X_k)^T(X - X_k)) + \frac{\sigma_\rho^k + \rho}{2} \|X - X_k\|_F^2. \quad (4.7)$$

Para o parâmetro σ_{spc}^k vamos escolher, sempre que possível, o parâmetro espectral de Barzilai-Borwein [3].

Este parâmetro foi proposto por Barzilai e Borwein no contexto dos problemas de minimização irrestrita com iteração dada por: $x_{k+1} = x_k - \alpha_k g(x_k)$, em que $x_k \in \mathbb{R}^n$ e

$$\alpha_k = \frac{\langle x_k - x_{k-1}, g(x_k) - g(x_{k-1}) \rangle}{\|x_k - x_{k-1}\|^2}. \quad (4.8)$$

¹Lembramos que, como colocado em (3.2), L_f é a constante de Lipschitz associada ao gradiente de f

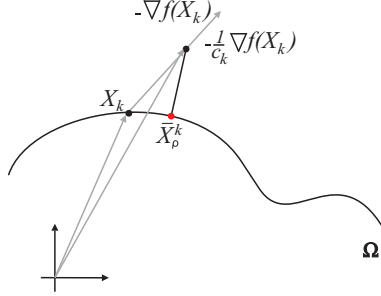


Figura 4: Iteração do Algoritmo Modelo para minimização com restrições de ortogonalidade, em que $c_k = \rho + \sigma_{\rho}^k$

Note que (4.8) é o quociente de Rayleigh relativo à média da matriz Hessiana $\int_0^1 \nabla^2 f(x_{k-1} + t(x_k - x_{k-1})) dt$ e, portanto, os valores α_k estão entre o menor e o maior autovalor desta matriz.

Ressaltamos que o uso do parâmetro de Barzilai-Borwein preserva as boas características do método utilizado e ainda reduz significativamente o número de buscas lineares, acelerando a convergência do mesmo [37]. Dessa forma, para dois iterados consecutivos, X_{k-1} e X_k , temos

$$\sigma_{bb}^k = \frac{\text{traço}((g(X_k)^T - g(X_{k-1})^T)(X_k - X_{k-1}))}{\|X_k - X_{k-1}\|_F^2}, \quad (4.9)$$

e definimos

$$\sigma_{spc}^k = \begin{cases} 1, & \text{para } k = 0 \\ \min\{\max\{\sigma_{bb}^k, \sigma_{min}\}, \sigma_{max}\}, & \text{para } k \geq 1 \end{cases} \quad (4.10)$$

Com estas escolhas, o Algoritmo Modelo torna-se uma variação do Método Gradiente Projetado Espectral não monótono de [5], para minimização com restrições de ortogonalidade. E pode ser representado geometricamente conforme a Fig. 4.

O teorema a seguir relaciona a minimização do modelo quadrático (4.7) com a minimização da função objetivo quando usamos o parâmetro definido em (4.9), justificando portanto sua escolha no Algoritmo 4.1 estabelecido mais à frente.

Teorema 4.4 *Suponha $f : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função quadrática, $x_{k-1}, x_k \in$*

Ω e defina $Q_k(x) = g(x_k)^T(x - x_k) + \frac{\sigma_{bb}^k}{2}\|x - x_k\|^2$. Então,

$$\begin{aligned} \min \quad & Q_k(x_k + t(x_k - x_{k-1})) \\ \text{s.a.} \quad & (x_k + t(x_k - x_{k-1})) \in \Omega \\ & t \in \mathbb{R}. \end{aligned}$$

é equivalente a

$$\begin{aligned} \min \quad & f(x_k + t(x_k - x_{k-1})) \\ \text{s.a.} \quad & (x_k + t(x_k - x_{k-1})) \in \Omega \\ & t \in \mathbb{R}. \end{aligned}$$

Demonstração: Considere $f(x) = \frac{1}{2}x^T Ax + b^T x + c$, com $A \in \mathbb{R}^{n \times n}$ simétrica. Então,

$$\begin{aligned} Q_k(x_k + t(x_k - x_{k-1})) &= tg(x_k)(x_k - x_{k-1}) + \frac{\sigma_{bb}^k}{2}t^2\|x_k - x_{k-1}\|^2 \\ &= t(Ax_k + b)^T(x_k - x_{k-1}) + \frac{t^2}{2} \frac{(Ax_k - Ax_{k-1})(x_k - x_{k-1})}{\|x_k - x_{k-1}\|^2} \|x_k - x_{k-1}\|^2 \\ &= t(Ax_k + b)^T(x_k - x_{k-1}) + \frac{t^2}{2}(x_k - x_{k-1})^T A(x_k - x_{k-1}) \\ &= f(x_k + t(x_k - x_{k-1})) - f(x_k) - 2c. \end{aligned}$$

Como $f(x_k)$ e c são constantes, o resultado segue. □

O próximo passo é encontrar a solução do subproblema

$$\begin{aligned} \min \quad & Q_\rho^k(X) \\ \text{s.a.} \quad & X^T X = I \\ & X \in \mathbb{R}^{m \times n}. \end{aligned} \tag{4.11}$$

Para tanto, defina para cada $k = 0, 1, 2, \dots$,

$$W_k = X_k - \frac{1}{\rho + \sigma_\rho^k} g(X_k),$$

e note que

$$\|X - W_k\|_F^2 = \|X - (X_k - \frac{1}{\rho + \sigma_\rho^k} g(X_k))\|_F^2$$

$$\begin{aligned}
&= 2\text{traço}(g(X_k)^T(X - X_k)) + (\rho + \sigma_\rho^k)\|X - X_k\|_F^2 + \\
&\quad + \frac{1}{\rho + \sigma_\rho^k}\|g(X_k)\|_F^2 \\
&= 2Q_\rho^k(X) + \frac{1}{\rho + \sigma_\rho^k}\|g(X_k)\|_F^2.
\end{aligned}$$

Ou seja, $Q_\rho^k(X)$ e $\|X - W_k\|_F^2$ diferenciam-se apenas por constantes. Dessa forma, resolver (4.11) é equivalente a encontrar a solução de

$$\begin{aligned}
\min \quad & \|X - W_k\|_F^2 \\
\text{s.a.} \quad & X^T X = I \\
& X \in \mathbb{R}^{m \times n},
\end{aligned}$$

cujas soluções são dadas no lema a seguir.

Lema 4.5 *Seja $A = U\Sigma V^T$ a Decomposição em Valores Singulares reduzida de $A \in \mathbb{R}^{m \times n}$. Então, $X^* = UV^T$ é solução de*

$$\begin{aligned}
\min \quad & \|X - A\|_F^2 \\
\text{s.a.} \quad & X^T X = I.
\end{aligned} \tag{4.12}$$

Demonstração: Seja $A = \overline{U}\overline{\Sigma}\overline{V}^T$ a decomposição SVD de A , em que $\overline{U} \in \mathbb{R}^{m \times m}$, $\overline{V} \in \mathbb{R}^{n \times n}$ são matrizes unitárias e $\overline{\Sigma} \in \mathbb{R}^{m \times n}$ é diagonal. Então,

$$\|X - A\|_F^2 = \|\overline{U}^T X \overline{V} - \overline{\Sigma}\|_F^2,$$

e assim, o problema (4.12) é equivalente a

$$\begin{aligned}
\min \quad & \|\overline{U}^T X \overline{V} - \overline{\Sigma}\|_F^2 \\
\text{s.a.} \quad & X^T X = I.
\end{aligned} \tag{4.13}$$

Defina $W = \overline{U}^T X \overline{V}$. Então, um cálculo simples mostra que $W^T W = I$ se, e somente se, $X^T X = I$, e portanto a solução do problema (4.13) é $X^* = \overline{U} W \overline{V}^T$, com W solução de

$$\begin{aligned}
\min \quad & \|W - \overline{\Sigma}\|_F^2 \\
\text{s.a.} \quad & W^T W = I.
\end{aligned}$$

isto é, $W = \begin{pmatrix} I_n & 0 \end{pmatrix}^T \in \mathbb{R}^{m \times n}$.

Logo, a solução X^* do problema (4.12) é

$$X^* = \overline{U} \begin{pmatrix} I_n \\ 0 \end{pmatrix} \overline{V}^T = UV^T,$$

em que $A = U\Sigma V^T$ é a decomposição SVD reduzida de A , concluindo assim a demonstração. \square

Podemos agora estabelecer o algoritmo para problemas com restrições de ortogonalidade, que denominamos Gradiente Projetado para minimização em variedades de Stiefel (GPST).

Algoritmo 4.1: Algoritmo para problemas com restrições de ortogonalidade (GPST)

Tome $X_0 \in \Omega$, $M \in \mathbb{Z}^+$, $\eta \in (0, 1]$, $\beta_1 \in (0, \frac{1}{2}]$, $0 < \sigma_{min} < \sigma_{max} < \infty$ e $1 < \xi_1 \leq \xi_2 < \infty$.

Faça $k = 0$ e $m(0) = 0$.

Passo 1. Calcule $g(X_k)$, $f(X_{l(k)})$ e faça $\rho = \frac{\sigma_{spc}^k}{2}$ como em (4.10);

Passo 2. Tome σ_ρ^k definido em (4.6) e calcule a SVD reduzida de W_k : $W_k = U\Sigma V^T$;

Passo 3. Defina $\overline{X}_\rho^k = UV^T$ e encontre $X_\rho^k \in \Omega$ tal que $Q_\rho^k(X_\rho^k) \leq \eta Q_\rho^k(\overline{X}_\rho^k)$. Se $Q_\rho^k(X_\rho^k) = 0$, declare X_k ponto estacionário;

Passo 4. Defina

$$\Psi_k(X) = \text{traço}(g(X_k)^T(X - X^k)) + \frac{\sigma_\rho^k}{2} \|X - X_k\|_F^2.$$

Se

$$f(X_\rho^k) \leq f(X_{l(k)}) + \beta_1 \Psi_k(X_\rho^k), \quad (4.14)$$

defina $\rho^k = \rho$, $X^{k+1} = X_\rho^k$, faça $k = k + 1$ e volte ao Passo 1.

Senão, escolha $\rho_{novo} \in [\xi_1 \rho, \xi_2 \rho]$, faça $\rho = \rho_{novo}$ e volte ao Passo 2.

O Teorema 3.10 garante a convergência global da sequência de iterados $\{X_k\}$, pois as hipóteses H1 e H2 assumidas na Seção 3.2 são trivialmente satisfeitas.

A escolha $\eta < 1$ é apropriada para problemas em que $m \gg n$.

Os pontos de Ω não são pontos regulares segundo a definição apresentada no Capítulo 1 (LICQ), porém estes pontos satisfazem uma outra condição de qualificação, a saber, a Condição de Qualificação de Posto Constante (CRCQ), conforme a proposição a seguir. Lembremos que condições de qualificação são exigências que um minimizador local deve cumprir para satisfazer as Condições KKT. É conhecido que todo minimizador local de (4.1) satisfaz as condições KKT do problema $\min_{X \in \Omega} f(X)$ [24]. Assim, podemos utilizar o item (i) do Teorema 4.3 como critério de parada e então declarar convergência do algoritmo apresentado neste capítulo. O próximo resultado mostra que ao retirarmos as restrições redundantes de Ω temos que a condição LICQ é satisfeita.

Lema 4.6 *Sejam $\Omega = \{X \in \mathbb{R}^{m \times n} : X^T S X = I\}$, $m \geq n$, $S > 0$ e $h_{ij}(X) = x_i^T S x_j - \delta_{ij}$, $i, j = 1, \dots, n$, em que $x_i \in \mathbb{R}^m$ é a i -ésima coluna de X e δ_{ij} é o delta de Kronecker. Então, o conjunto $\Gamma = \{\nabla h_{ij}(X)\}_{i \geq j}$ é linearmente independente (LI).*

Demonstração: Seja $X \in \Omega$. Organizando as variáveis na forma vetorial, temos:

$$\text{vec}(X) = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^{mn}.$$

Note que,

$$\nabla h_{jj}(\text{vec}(X)) = \begin{pmatrix} 0 \\ \vdots \\ 2Sx_j \\ \vdots \\ 0 \end{pmatrix} \leftarrow \text{posição } (j-1)m+1 \text{ até } jm$$

e

$$\nabla h_{ij}(\text{vec}(X)) = \begin{pmatrix} 0 \\ \vdots \\ Sx_i \\ 0 \\ \vdots \\ Sx_j \\ \vdots \\ 0 \end{pmatrix} \begin{array}{l} \leftarrow \text{posição } (j-1)m+1 \text{ até } jm \\ \\ \leftarrow \text{posição } (i-1)m+1 \text{ até } im \end{array}$$

Então, para j fixo, Γ é linearmente independente. De fato, considere a combinação linear

$$c_j \nabla h_{jj}(\text{vec}(X)) + c_{j+1} \nabla h_{(j+1)j}(\text{vec}(X)) + \cdots + c_n \nabla h_{nj}(\text{vec}(X)) = 0$$

$$\Rightarrow \begin{pmatrix} 0 \\ \vdots \\ 2c_j Sx_j + \cdots + c_n Sx_n \\ c_{j+1} Sx_j \\ \vdots \\ c_n Sx_j \end{pmatrix} = 0.$$

Note que, como S é definida positiva $Sx_j \neq 0$, para todo $j \in \{1, \dots, n\}$, pois, caso contrário, teríamos $x_l^T Sx_l = 0$ para algum $l \in \{1, \dots, n\}$. Segue que, $c_j = c_{j+1} = \cdots = c_n = 0$ e, para j fixo, Γ é LI.

O mesmo é válido ao deixarmos j livre, pois considerando a combinação linear

$$\sum_{j=1}^n \sum_{i \geq j} c_{ij} \nabla h_{ij}(\text{vec}(X)) = 0,$$

isto é,

$$\begin{pmatrix} 2c_{11}Sx_1 + c_{21}Sx_2 + \cdots + c_{n1}Sx_n \\ c_{21}Sx_1 + 2c_{22}Sx_2 + \cdots + c_{n2}Sx_n \\ \vdots \\ c_{n1}Sx_1 + c_{n2}Sx_2 + \cdots + 2c_{nn}Sx_n \end{pmatrix} = 0,$$

e usando o fato que para j fixo temos independência linear, concluímos que Γ é LI. □

Proposição 4.7 *Se $X \in \Omega$, então X satisfaz CRCQ. Isto é, se um conjunto Γ_X de gradientes das restrições h em X , $\Gamma_X = \{\nabla h_{ij}(\text{vec}(X)) : i, j \in \{1, \dots, n\}\}$, é tal que $\dim(\text{span}\{\Gamma_X\}) = p$, então para qualquer Y em uma vizinhança de X , $\dim(\text{span}\{\Gamma_Y\}) = p$.*

Demonstração: Pelo lema anterior, temos que, para todo $X \in \Omega$, $\Gamma_X = \{\nabla h_{ij}(\text{vec}(X))\}_{i \geq j}$ é um conjunto linearmente independente. Então, para todo $X \in \Omega$, qualquer subconjunto $\Gamma'_X \subseteq \Gamma_X$ é, também, um conjunto LI. Isto é, para qualquer subconjunto Γ'_X , $\dim(\text{span}\{\Gamma'_X\})$ é constante em Ω .

Se adicionarmos o gradiente de uma restrição $\nabla h_{kl}(\text{vec}(X))$ ao subconjunto Γ'_X , podemos ter dois casos:

- (i) $\nabla h_{kl}(\text{vec}(X))$ é LI com os demais elementos de Γ'_X e, neste caso, pelo argumento anterior, $\Gamma'_X \cup \{\nabla h_{kl}(\text{vec}(X))\}$ é LI para todo $X \in \Omega$ e, portanto, $\dim(\text{span}\{\Gamma'_X\})$ é constante em Ω ;
- (ii) $\nabla h_{kl}(\text{vec}(X))$ é linearmente dependente (LD) à Γ'_X . Neste caso, $k < l$ e $\nabla h_{kl}(\text{vec}(X)) = \nabla h_{lk}(\text{vec}(X))$ para algum gradiente $\nabla h_{lk}(\text{vec}(X)) \in \Gamma'_X$. Segue que,

$$\dim(\text{span}\{\Gamma'_X\}) = \dim(\text{span}\{\Gamma'_X \cup \{\nabla h_{kl}(\text{vec}(X))\}\}),$$

para todo $X \in \Omega$ e, portanto, a condição de posto constante é satisfeita.

Analogamente, ao adicionarmos um número finito de gradientes de restrições ao conjunto Γ'_X a dimensão do subespaço gerado por Γ'_X é constante para todo $X \in \Omega$.

Logo, todo $X \in \Omega$ satisfaz CRCQ. □

No próximo capítulo apresentamos os resultados obtidos.

5 RESULTADOS NUMÉRICOS

Neste capítulo vamos implementar o algoritmo apresentado no capítulo anterior e compará-lo com métodos tradicionais para resolver problemas de minimização com restrições de ortogonalidade, a saber um método de busca curvilínea [44], um método de minimização em geodésicas [1] e o Método da Iteração Ortogonal Inversa.

Os testes serão executados utilizando Matlab 7.10 em um processador intel CORE i3 com HD de 500 Gb e 4 Gb de memória Ram, sendo utilizadas os seguintes valores para as constantes do Algoritmo 4.1: $\beta_1 = 10^{-4}$, $\xi_1 = \xi_2 = 5$, $\sigma_{min} = 10^{-10}$, $L_u = L_f$ e $\sigma_{max} = L_u$. O parâmetro de não monotonia, quando não estiver devidamente explicitado, é considerado $M = 7$. Assumiremos convergência quando

$$\|X_k(X_k^T g(X_k) + g(X_k)^T X_k) - 2g(X_k)\|_F \leq 10^{-6},$$

limitando o tempo de execução (TMAX) em 600 segundos.

Lembramos que o número de condição de uma matriz $A \in \mathbb{R}^{n \times n}$ é dado por

$$\mathcal{K}(A) = \|A\| \|A^{-1}\|,$$

e que uma matriz com número de condição pequeno ($\mathcal{K}(A) \approx 1$) é dita bem condicionada, enquanto que uma matriz com um número de condição elevado é dita mal condicionada.

Apresentamos a seguir os métodos utilizados para comparar o desempenho do Algoritmo 4.1.

5.1 MÉTODO DA ITERAÇÃO ORTOGONAL INVERSA

O método da iteração ortogonal inversa é uma generalização do método da potência inverso, utilizado para calcular o subespaço invariante de dimensão r , $1 \leq r \leq n$, gerado pelos autovetores associados aos r menores autovalores da matriz $A \in \mathbb{R}^{n \times n}$.

Dados $A \in \mathbb{R}^{n \times n}$ não singular e $Q_0 \in \mathbb{R}^{n \times r}$ ortogonal, o método consiste em, a cada passo k resolver o sistema

$$AZ_k = Q_{k-1},$$

seguinte de uma fatoração QR de Z_k , $Z_k = Q_k R_k$. Note que, o caso em que $r = 1$, coincide com o método da potência inverso.

É conhecido que este método tem uma taxa de convergência proporcional a $\left| \frac{\lambda_r}{\lambda_{r+1}} \right|^k$, em que $|\lambda_1| \leq |\lambda_2| \leq \dots \leq |\lambda_n|$ são os autovalores de A . Mais informações sobre o método podem ser encontradas em [20].

Algoritmo 5.1: Método da iteração ortogonal inversa (Potência)

Seja $Q_0 \in \Omega$ e $1 \leq r \leq n$.

Faça $k = 0$.

Passo 1. Resolva $AZ_k = Q_{k-1}$.

Passo 2. Calcule $Z_k = Q_k R_k$, a decomposição QR de Z_k , e volte ao Passo 1.

5.2 MÉTODO DE MINIMIZAÇÃO ATRAVÉS DE GEODÉSICAS

Utilizaremos uma variação do Método de Newton desenvolvido por Arias, Edelman e Smith [1] para minimização em variedades de Stiefel e de Grassmann¹, que aproveita a geometria dessas variedades.

Em problemas de minimização irrestritos, o Método de Newton consiste em gerar novos iterados subtraindo do iterado atual um múltiplo do vetor $H^{-1}\nabla f$, em que H é a matriz Hessiana da função f avaliada no iterado atual. Arias, Edelman e Smith [1] substituíram essa subtração restringindo a função ao longo de um caminho geodésico na variedade. O gradiente continua sendo o usual (tangente à superfície das restrições), e a matriz Hessiana é obtida através da segunda diferencial da função restrita à geodésica.

A seguir apresentamos a notação utilizada e colocamos um resumo do modelo do algoritmo. Para mais detalhes ver [1, 14].

Considere o problema de minimização

$$\begin{aligned} \min \quad & f(X) \\ \text{s.a.} \quad & X \in \Omega, \end{aligned}$$

em que, $f : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ e $\Omega = \{X \in \mathbb{R}^{m \times n} : X^T X = I\}$.

¹Uma variedade de Grassmann pode ser vista como um espaço de matrizes simétricas, idempotentes ($D^2 = D$) e com traço constante. Recebeu esse nome em homenagem ao matemático polonês Hermann Grassmann (1809-1877).

Vamos denotar por $F_Y \in \mathbb{R}^{m \times n}$ a derivada parcial de f com relação aos elementos de Y , isto é,

$$(F_Y)_{ij} = \frac{\partial f}{\partial Y_{ij}},$$

e por $F_{YY}(\Delta_1, \Delta_2)$ o valor,

$$F_{YY}(\Delta_1, \Delta_2) = \sum_{ij,kl} (F_{YY})_{ij,kl} (\Delta_1)_{ij} (\Delta_2)_{kl},$$

em que $\Delta_1, \Delta_2 \in \mathbb{R}^{m \times n}$ e

$$(F_{YY})_{ij,kl} = \frac{\partial^2 f}{\partial Y_{ij} \partial Y_{kl}}.$$

Seja Δ a solução do sistema linear

$$\begin{aligned} F_{YY}(\Delta) - Y \text{skew}(F_Y^T \Delta) - \text{skew}(\Delta F_Y^T) - \frac{1}{2} \Pi \Delta Y^T F_Y &= -G \\ (Y^T \Delta)^T &= -Y^T \Delta, \end{aligned} \quad (5.1)$$

em que $\text{skew}(X) = \frac{X - X^T}{2}$, $\Pi = I - YY^T$ e $F_{YY}(\Delta)$ representa o único vetor tangente que satisfaz

$$F_{YY}(\Delta, Z) = \langle F_{YY}(\Delta), Z \rangle, \quad \forall Z \in \mathbb{R}^{m \times n}.$$

Abaixo apresentamos, em termos gerais, o algoritmo de [1], que será denominado daqui em diante por `sg_min`.

Algoritmo 5.2: Método de Newton para minimização em variedades de Stiefel (`sg_min`)

Seja $Y \in \Omega$.

Passo 1. Calcule $G = F_Y - F_Y^T Y$.

Passo 2. Encontre Δ solução de (5.1).

Passo 3. Mova de Y para $Y(1)$ na direção Δ usando a fórmula geodésica

$$Y(t) = YM(t) + QN(t), \quad (5.2)$$

em que Q vem da decomposição QR magra de $(I - YY^T)\Delta$, $M(t)$ e $N(t)$ são matrizes dadas pela equação

$$\begin{pmatrix} M(t) \\ N(t) \end{pmatrix} = \exp t \begin{pmatrix} Y^T \Delta & -R^T \\ R & 0 \end{pmatrix} \begin{pmatrix} I_n \\ 0 \end{pmatrix},$$

e volte ao Passo 1.

Nos experimentos apresentados à frente usamos a opção `dogleg` do pacote `sg_min2` para ajustar o parâmetro da busca linear ao longo da geodésica. Neste caso, ao invés de $t = 1$, obtemos um parâmetro t_k em (5.2) tal que $f(Y(t_k)) < f(Y)$.

5.3 MÉTODO DE BUSCA CURVILINEAR

Nesta seção apresentamos um método de busca curvilinear não monótono com comprimento do passo ajustado pelo parâmetro espectral de Barzilai-Borwein, introduzido em [44]. Para tanto, vamos considerar o problema de otimização com restrições de ortogonalidade

$$\begin{aligned} \min \quad & f(X) \\ \text{s.a.} \quad & X \in \Omega, \end{aligned}$$

em que, como usual, $\Omega = \{X \in \mathbb{R}^{m \times n} : X^T X = I\}$.

O método consiste em construir uma sequência de iterados $X_k \in \Omega$ conforme descrito a seguir.

Para cada $k \in \mathbb{N}$, defina

$$W_k = \nabla f(X_k) X_k^T - X_k \nabla f(X_k)^T \quad (5.3)$$

e

$$X_{k+1} = Y(t_k) = \left(I + \frac{t_k}{2} W_k \right)^{-1} \left(I - \frac{t_k}{2} W_k \right) X_k.$$

Note que podemos escrever $Y(t) = QX$, em que

$$Q := \left(I + \frac{t}{2} W_k \right)^{-1} \left(I - \frac{t}{2} W_k \right),$$

²O solver utilizado está disponível em <http://web.mit.edu/ripper/www/sgmin.html> (acessado em 29/01/2012)

conhecida como transformada de Cayley, leva a diversas propriedades interessantes: a curva $Y(t)$ é suave em t , $Y(0) = X$ e, a mais importante, $Y(t)^T Y(t) = X^T X$ para todo $t \in \mathbb{R}$. Esta última decorre do teorema a seguir.

Teorema 5.1 *Sejam $W \in \mathbb{R}^{m \times m}$ anti-simétrica e $Q : \mathbb{R} \rightarrow \mathbb{R}^{m \times n}$ dada por $Q(t) = (I + \frac{t}{2}W)^{-1} (I - \frac{t}{2}W)$. Então, para todo $t \in \mathbb{R}$, $Q(t)$ é ortogonal.*

Demonstração: Note que

$$\begin{aligned} Z &= (I + \frac{t}{2}W)^{-1} (I - \frac{t}{2}W)^{-1} = ((I - \frac{t}{2}W) (I + \frac{t}{2}W))^{-1} \\ &= \left(I^2 + \frac{t}{2}W - \frac{t}{2}W - \frac{t^2}{4}W \right)^{-1} = ((I + \frac{t}{2}W) (I - \frac{t}{2}W))^{-1} \\ &= (I - \frac{t}{2}W)^{-1} (I + \frac{t}{2}W)^{-1}. \end{aligned}$$

Então,

$$\begin{aligned} Q(t)^T Q(t) &= (I + \frac{t}{2}W^T) (I - \frac{t}{2}W^T)^{-1} (I - \frac{t}{2}W)^{-1} (I + \frac{t}{2}W) \\ &= (I - \frac{t}{2}W) (I + \frac{t}{2}W)^{-1} (I - \frac{t}{2}W)^{-1} (I + \frac{t}{2}W) \\ &= (I - \frac{t}{2}W) (I - \frac{t}{2}W)^{-1} (I + \frac{t}{2}W)^{-1} (I + \frac{t}{2}W) = I \end{aligned}$$

o que conclui a demonstração. □

O passo da busca linear sob a curva Y é definido pelo parâmetro espectral de Barzilai-Borwein, ou seja,

$$t_k = \frac{\text{traço}(S_{k-1}^T S_{k-1})}{|\text{traço}(S_{k-1}^T Y_{k-1})|},$$

em que $S_{k-1} = X_k - X_{k-1}$ e $Y_{k-1} = \nabla f(X_k) - \nabla f(X_{k-1})$. Para garantir a convergência global do método Wen e Yin [44] adotaram a estratégia de não monotonia de [46], assim o passo t_k será dado por $t_k = t_k \delta^{\tau_k}$, em que τ_k é o menor inteiro que satisfaz

$$f(Y(t_k)) \leq C_k + \rho_1 t_k \delta^{\tau_k} \langle \nabla f(X_k), W_k X_k \rangle,$$

com $C_{k+1} = \frac{\eta q_k C_k + f(X_{k+1})}{q_{k+1}}$, $q_{k+1} = \eta q_k + 1$ e $q_0 = 1$.

Nos experimentos utilizamos um solver³ baseado no Algoritmo 2 de [44]. A seguir apresentamos o algoritmo que, daqui em diante, será denominado OptStiefel.

Algoritmo 5.3: Método de busca curvilinear com Barzilai-Borwein (OptStiefel)

Sejam $X_0 \in \Omega$, $t_0 > 0$, $0 < t_m < t_M$, $\rho_1, \delta, \eta, \varepsilon \in (0, 1)$.

Faça $k = 0$.

Passo 1. Calcule $f(X_k)$, $g(X_k)$ e defina W_k como em (5.3). Se $\|g(X_k)\| < \varepsilon$, pare. Senão, vá para o próximo passo.

Passo 2. Calcule $f(Y(t_k))$. Enquanto $f(Y(t_k)) \geq C_k + \rho_1 t_k \langle \nabla f(X_k), W_k X_k \rangle$, faça $t_k = \delta t_k$.

Passo 3. Defina $X_{k+1} = Y(t_k)$, $q_{k+1} = \eta q_{k+1}$, $C_{k+1} = \frac{\eta q_k C_k + f(X_{k+1})}{q_{k+1}}$. Tome $t_{k+1} = \max\{\min\{t_{k+1}, t_M\}, t_m\}$, faça $k = k + 1$ e volte ao Passo 1.

5.4 TESTE NUMÉRICOS E PROBLEMAS

Vamos testar os algoritmos no Problema de Procrustes Ortogonal Ponderado e no cálculo de uma base ortonormal para o subespaço gerado pelos menores autovalores de uma matriz simétrica. A seguir apresentamos cada um dos problemas e comentamos os resultados obtidos ao comparar os algoritmos testados.

5.4.1 Problema de Procrustes Ortogonal

Na mitologia grega, Procrustes foi um vilão e torturador que possuía uma cama de ferro onde colocava suas vítimas, cortando-lhes ou esticando alguns membros para que se ajustassem ao tamanho da cama. Em matemática, um problema do tipo Procrustes consiste em analisar o processo de preservação de uma transformação a um conjunto de formas. Tal processo envolve rotação e mudança de escala em um

³O solver utilizado está disponível em <http://optman.blogs.rice.edu/> (acessado em 29/01/2012)

certo conjunto de dados, a fim de movê-los para um quadro comum de referência onde possam ser comparados. Os dados conhecidos são dispostos em matrizes e a transformação de rotação a ser determinada é uma matriz com colunas ortonormais.

Vamos trabalhar com duas formulações desse problema, a saber, Procrustes Ortogonal e Procrustes Ortogonal Ponderado, conforme as definições a seguir.

Definição 5.2 *Seja $X \in \mathbb{R}^{m \times n}$, $m \geq n$. Dadas as matrizes $A \in \mathbb{R}^{p \times m}$, $C \in \mathbb{R}^{n \times q}$ e $B \in \mathbb{R}^{p \times q}$, o problema denominado **Problema de Procrustes Ortogonal Ponderado (WOPP)** consiste em resolver*

$$\begin{aligned} \min \quad & \|AXC - B\|_F^2 \\ \text{s.a.} \quad & X^T X = I. \end{aligned} \quad (5.4)$$

*Quando $C = I$ diremos que (5.4) é o **Problema de Procrustes Ortogonal (OPP)**.*

De acordo com Eldén e Park [15], o problema OPP com $m = n$ é dito balanceado e o caso em que $n < m$ é chamado de não balanceado. Para o caso balanceado, decorre do Lema 4.5 uma fórmula fechada para a solução, a saber $X_* = UV^T$, em que $A^T B = U \Sigma V^T$ é a decomposição SVD reduzida de $A^T B$. Para o caso não balanceado e para WOPP a solução é encontrada através de métodos iterativos e, portanto, não é necessariamente o mínimo global.

Problemas do tipo OPP e WOPP surgem em aplicações relacionadas com, por exemplo, movimento de corpos rígidos [39], psicometria [23], escalonamento multidimensional [10] e Sistema de Posicionamento Global (GPS) [4].

Mostraremos que esse problema satisfaz as condições impostas no desenvolvimento do Algoritmo 4.1. Para tanto, vamos considerar a função

$$f : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$$

$$f(X) = \|AXC - B\|_F^2.$$

Note que $f \in \mathcal{C}^2$ e como $g(X) = 2A^T(AXC - B)C^T$, temos

$$\begin{aligned} \|g(X) - g(Y)\|_F &= 2\|A^T A(X - Y)CC^T\|_F \\ &\leq 2\|A^T A\|_F \|CC^T\|_F \|X - Y\|_F, \end{aligned}$$

isto é, $g = \nabla f$ é Lipschitz contínua, com constante de Lipschitz satisfazendo

$$L_f \leq 2\|A^T A\|_F \|CC^T\|_F.$$

Ressaltamos que a utilização do parâmetro espectral de Barzilai-Borwein [3] é conveniente, visto que a função é uma quadrática e, portanto, pelo Teorema 4.4 resolver o subproblema (4.7) equivale a minimizar a função f ao longo da direção $(X_k - X_{k-1})$.

Os problemas 1, 2, 3, e 4 foram retirados de [47] e são descritos ao longo do texto. Vamos considerar $n = q$, $p = m$, $A = PSR^T$ e $C = Q\Lambda Q^T$, com P e R matrizes ortogonais com entradas randômicas, $Q \in \mathbb{R}^{n \times n}$ matriz de Householder, $\Lambda \in \mathbb{R}^{n \times n}$ diagonal com entradas uniformemente distribuídas no intervalo $[\frac{1}{2}, 2]$ e S diagonal definida de modo diferente para cada um dos problemas. Como ponto inicial X_0 vamos gerar uma matriz randômica satisfazendo $X_0 \in \Omega$. Todos os valores randômicos foram gerados com a função `randn` do Matlab.

Além disso, para monitorar o comportamento dos iterados X_k com respeito a solução exata do problema (5.4), vamos criar uma solução Q_* tomando $B = AQ_*C$, em que $Q_* \in \mathbb{R}^{m \times n}$ é matriz de Householder randômica.

Problema 1. Os elementos da diagonal de S são gerados através de uma distribuição normal no intervalo $[10, 12]$.

Note que a matriz A para este problema, por conter valores singulares muito próximos, é bem condicionada. Portanto, o uso da não monotonia pode não apresentar uma redução significativa no número de buscas lineares, conforme verificado na Tab. 2 e na Fig. 5, onde não há muitos acréscimos no valor da função. Uma característica do Algoritmo 4.1 que também pode ser observada na Fig. 5 é o decréscimo do resíduo já nas primeiras iterações. Esse fato é provavelmente uma consequência do ρ inicial satisfazer a condição (4.14) devido ao uso do parâmetro espectral.

Tabela 2: Comparativo entre os casos monótono ($M = 0$) e não monótono ($M = 7$) do Algoritmo 4.1 para o Problema 1 com $m = 500$

q	Monótono		Não Monótono	
	N_i	N_a	N_i	N_a
10	18	24	18	23
20	22	30	19	24
50	24	33	19	24
70	24	33	20	25

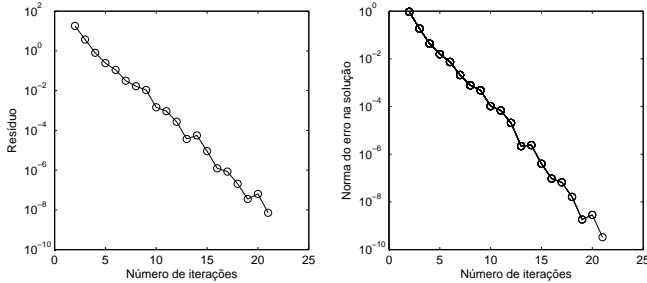


Figura 5: Comportamento do Algoritmo 4.1 não monótono para o Problema 1 com $m = 500$ e $q = 50$

A Tab. 3 expõe os resultados alcançados para o Problema 1 ao variarmos a dimensão do mesmo. Denotamos o número de iterações por N_i e o número de avaliações da função por N_a . O resíduo $\|AX_kC - B\|_F^2$, a norma do erro na solução, $\|X_k - Q_*\|_F$, e o tempo (em segundos) também são apresentados.

Em resumo, o Algoritmo 5.2 (sg_min) não apresentou um bom desempenho, pois, apesar de realizar poucas iterações, necessitou avaliar muitas vezes a função. Os Algoritmos 4.1 (GPST) e 5.3 (OptStiefel) exibiram comportamento semelhante, tanto no número de iterações quanto no número de avaliações da função, não sendo necessário muitos passos de busca linear e curvilínea, respectivamente. O tempo gasto também foi próximo.

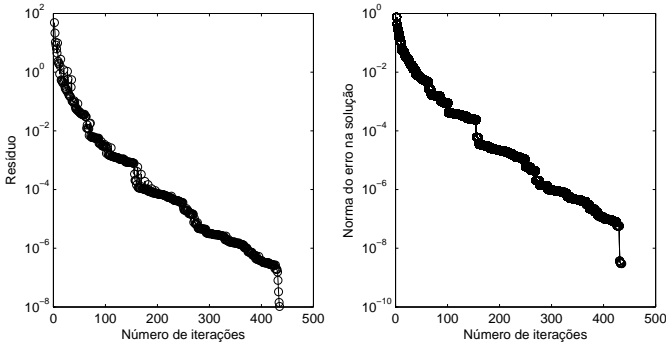
Problema 2. A diagonal de S é dada por: $S_{ii} = i + 2r_i$, em que r_i são números aleatórios distribuídos uniformemente no intervalo $[0, 1]$.

Note que, neste caso, os valores singulares de A tornam-se mais distantes uns dos outros à medida que m aumenta, deixando a matriz mal condicionada. Para este problema, a estratégia de não monotonia mostra seu potencial, conforme pode ser observado na Fig. 6. É notável a redução do número de avaliações da função e, conseqüentemente, do número de buscas lineares. Conforme apresentado na Tab. 4.

Podemos observar na Tab. 5 que o método OptStiefel foi o que apresentou os melhores resultados para os casos analisados. Lembramos que este algoritmo está consolidado e sua implementação otimizada, enquanto que o Algoritmo 4.1 (GPST) teve uma implementação simples,

Tabela 3: Desempenho dos métodos para o Problema 1

$m = 50$ e $q = 10$					
Método	N_i	N_a	Tempo	Resíduo	Erro
GPST	18	23	0.156	2.42e-08	1.44e-09
OptStiefel	27	27	0.015	1.45e-08	7.80e-10
sg_min	29	118	1.310	1.83e-08	1.33e-09
$m = 100$ e $q = 20$					
GPST	20	24	0.124	1.47e-08	6.80e-10
OptStiefel	28	28	0.124	1.85e-08	1.36e-09
sg_min	29	118	2.823	2.36e-08	1.80e-09
$m = 500$ e $q = 70$					
GPST	20	25	2.230	3.52e-08	2.59e-09
OptStiefel	31	31	1.716	3.67e-08	2.96e-09
sg_min	33	134	3.631	1.88e-08	1.41e-09
$m = 1000$ e $q = 100$					
GPST	20	25	4.711	1.27e-08	9.39e-10
OptStiefel	31	31	4.680	3.19e-08	2.58e-09
sg_min	32	130	164.8	1.76e-08	1.34e-09

Figura 6: Comportamento do Algoritmo 4.1 não monótono para o Problema 2 com $m = 50$ e $q = 10$

onde a escolha dos parâmetros pode ser melhorada. No entanto, o desempenho do método GPST foi adequado. Para dimensões maiores o método sg_min atingiu o tempo limite TMAX fixado e, mesmo em

Tabela 4: Comparativo entre os casos monótono e não monótono do Algoritmo 4.1 para o Problema 2 com $m = 100$

Parâmetro	q=10		q=50	
	N_i	N_a	N_i	N_a
M = 0	737	1218	1014	1664
M = 5	720	1015	1392	2050
M = 7	572	782	1474	2088
M = 10	446	558	1151	1566
M = 15	703	855	1300	1661

dimensões menores este método não apresentou bons resultados.

Tabela 5: Desempenho dos métodos para o Problema 2

$m = 50$ e $q = 10$					
Método	N_i	N_a	Tempo	Resíduo	Erro
GPST	434	587	0.889	1.04e-08	2.99e-09
OptStiefel	208	209	0.140	2.65e-08	6.80e-09
sg_min	4511	18046	72.44	1.16e-07	3.51e-08
$m = 100$ e $q = 10$					
GPST	572	782	1.294	4.13e-08	7.93e-09
OptStiefel	346	370	0.421	2.04e-08	3.80e-09
sg_min	5000	20002	135.0	1.66e-06	3.21e-07
$m = 100$ e $q = 50$					
GPST	1474	2088	12.09	1.39e-07	4.01e-08
OptStiefel	534	569	1.934	3.70e-08	1.02e-08
sg_min	5000	20002	586.2	1.74e-03	4.99e-04
$m = 500$ e $q = 20$					
GPST	8244	11927	98.73	1.05e-07	2.24e-08
OptStiefel	1364	1441	11.09	8.04e-06	1.66e-06
sg_min	-	-	TMAX	-	-
$m = 500$ e $q = 50$					
GPST	7931	11414	282.1	1.24e-07	3.10e-08
OptStiefel	2014	2131	52.55	7.06e-06	1.76e-06
sg_min	-	-	TMAX	-	-

Problema 3. Cada entrada da diagonal de S é gerada da seguinte maneira: $S_{ii} = 1 + \frac{99(i-1)}{m-1} + 2r_i$, com r_i distribuídos uniformemente no intervalo $[0, 1]$.

Podemos observar na Tab. 6 que o uso da não monotonia reduziu consideravelmente o número de iterações e avaliações da função, isto é, de buscas lineares.

Tabela 6: Comparativo entre os casos monótono e não monótono do Algoritmo 4.1 para o Problema 3 com $m = 100$

Parâmetro	q=10		q=50	
	N_i	N_a	N_i	N_a
M = 0	746	1215	1030	1677
M = 5	524	708	858	1207
M = 7	523	681	772	1069
M = 10	648	869	1048	1386
M = 15	795	984	885	1102

A Tab. 7 mostra que, para este problema, o método OptStiefel também foi o que apresentou os melhores resultados, com uma redução significativa em relação aos valores necessários para a convergência dos métodos `sg_min` e `GPST`.

Problema 4. A matriz S é definida utilizando as funções ‘ones’ e ‘rand’ do Matlab, da seguinte maneira:

$$S = \text{diag}([10 * \text{ones}(1, m_1) + \text{rand}(1, m_1), 5 * \text{ones}(1, m_2) + \text{rand}(1, m_2), 2 * \text{ones}(1, m_3) + \text{rand}(1, m_3), \text{rand}(1, m_4)/1000]),$$

com $m_1 + m_2 + m_3 + m_4 = m$. Neste problema definimos $C = I$.

Vamos utilizar os valores de m_1, m_2, m_3 e m_4 sugeridos em [47], ou seja

4.1. $m_1 = m_2 = 15, m_3 = 12$ e $m_4 = 0$;

4.2. $m_1 = m_2 = m_3 = 30$ e $m_4 = 5$.

Note que A possui valores singulares muito pequenos quando comparados com o maior valor singular, e portanto é mal condicionada (nos exemplos testados, $\mathcal{K}(A) > 10^5$).

Tabela 7: Desempenho dos métodos para o Problema 3

$m = 50$ e $q = 10$					
Método	N_i	N_a	Tempo	Resíduo	Erro
GPST	715	988	1.357	1.06e-07	2.41e-08
OptStiefel	564	605	0.764	1.24e-07	2.82e-08
sg_min	11263	45054	215.9	9.95e-08	2.26e-08
$m = 100$ e $q = 10$					
GPST	523	681	0.998	8.50e-08	1.73e-08
OptStiefel	350	377	0.421	8.33e-08	1.39e-08
sg_min	7466	29866	182.5	8.99e-08	1.84e-08
$m = 100$ e $q = 50$					
GPST	772	1069	5.725	6.27e-08	1.59e-08
OptStiefel	562	601	2.074	1.63e-07	4.17e-08
sg_min	-	-	TMAX	-	-
$m = 500$ e $q = 20$					
GPST	1080	1505	10.85	1.42e-07	5.19e-08
OptStiefel	662	715	4.882	4.23e-07	1.53e-07
sg_min	-	-	TMAX	-	-
$m = 500$ e $q = 70$					
GPST	1403	1965	63.77	1.55e-07	5.76e-08
OptStiefel	522	556	16.64	5.10e-07	1.88e-07
sg_min	-	-	TMAX	-	-
$m = 1000$ e $q = 10$					
GPST	1423	1991	34.95	1.41e-07	4.33e-08
OptStiefel	622	649	10.01	7.93e-07	2.43e-07
sg_min	-	-	TMAX	-	-

A Tab. 8 ilustra que o uso da não monotonia foi eficaz, reduzindo significativamente o número de passos da busca linear. O mesmo pode ser observado na Fig. 7.

Nos problemas testados o algoritmo GPST foi o mais eficiente, como podemos observar na Tab. 9.

Para reduzir o esforço computacional exigido no Algoritmo 4.1 durante o cálculo da SVD, propomos encontrar essa decomposição por meio de um algoritmo alternativo. Apresentamos essa ideia na subseção seguinte.

Tabela 8: Comparativo entre os casos monótono e não monótono do Algoritmo 4.1 para os Problemas 4.1 e 4.2

Parâmetro	m=50 e q=10		m=95 e q=10	
	N_i	N_a	N_i	N_a
M = 0	174	274	284	451
M = 5	123	162	261	350
M = 7	151	173	254	324
M = 10	145	165	292	367
M = 15	131	146	326	385

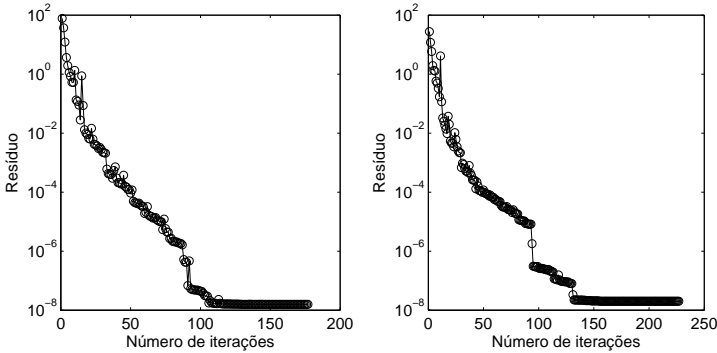


Figura 7: Comportamento do Algoritmo 4.1 não monótono para os Problemas 4.1 e 4.2 com $q = 10$

5.4.2 Usando a SVD Inexata

Descrevemos o método apresentado por Chahlaoui, Gallivan e Van Dooren [8] para calcular, recursivamente, a decomposição SVD de uma matriz. Isso será feito através do cálculo dos subespaços singulares dominantes, de dimensão k ($k \leq n$), à esquerda e à direita da matriz. O processo descrito é adequado para matrizes em que o número de linhas é muito maior do que o número de colunas. A seguir apresentamos o método em linhas gerais.

Tabela 9: Desempenho dos métodos para os Problemas 4.1 e 4.2

$m = 50$ e $q = 2$					
Método	N_i	N_a	Tempo	Resíduo	Erro
GPST	226	307	0.218	2.14e-08	1.90e-01
OptStiefel	206	213	0.124	1.46e-04	1.90e-01
sg_min	1777	7110	24.64	1.45e-04	1.90e-01
$m = 50$ e $q = 10$					
GPST	151	173	0.218	6.18e-08	3.86e-01
OptStiefel	208	210	0.374	2.54e-04	3.97e-01
sg_min	1246	4986	27.92	2.75e-04	4.64e-01
$m = 95$ e $q = 2$					
GPST	399	558	0.717	5.28e-10	3.78e-02
OptStiefel	394	401	5.148033e-001	2.30e-05	3.78e-02
sg_min	8181	32726	11.44	2.50e-05	3.98e-02
$m = 95$ e $q = 10$					
GPST	284	451	0.826	8.12e-09	1.95e-01
OptStiefel	236	244	0.452	9.00e-05	1.95e-01
sg_min	1511	6046	41.91	1.34e-04	3.17e-01

Dada $A \in \mathbb{R}^{m \times n}$, escreva

$$A = \begin{pmatrix} | & | & \cdots & | \\ a_1 & a_2 & \cdots & a_n \\ | & | & & | \end{pmatrix},$$

com $a_i \in \mathbb{R}^m$, $i = 1, \dots, n$, e considere

$$A_i = \begin{pmatrix} | & | & \cdots & | \\ a_1 & a_2 & \cdots & a_i \\ | & | & & | \end{pmatrix},$$

isto é, $A_i \in \mathbb{R}^{m \times i}$ é formada pelas i primeiras colunas de A .

O processo inicia com a decomposição QR de A_k , que será escrita da forma $A_k V_k = Q_k R_k$, com $V_k = I_k$. Passamos então a atualizar essa decomposição, conforme se segue.

Para $i = k + 1, \dots, n$, definimos: $r_i = Q_{i-1}^T a_i$, $\hat{a}_i = a_i - Q_{i-1} r_i$, $\rho_i = \|\hat{a}_i\|$ e $\hat{q}_i = \frac{\hat{a}_i}{\rho_i}$. Note que \hat{a}_i é ortogonal a Q_{i-1} . Então,

$$\left(\begin{array}{cc} Q_{i-1}R_{i-1} & a_i \end{array} \right) = \left(\begin{array}{c} Q_{i-1} \\ \hat{q}_i \end{array} \right) \left(\begin{array}{cc} R_{i-1} & r_i \\ 0 & \rho_i \end{array} \right). \quad (5.5)$$

Calculamos o vetor singular associado ao menor valor singular de

$$\hat{R}_i = \left(\begin{array}{cc} R_{i-1} & r_i \\ 0 & \rho_i \end{array} \right),$$

isto é, o vetor u_i que satisfaz $\hat{R}_i v_i = u_i \mu_i$, e construímos, através de transformações de Householder, uma transformação ortogonal G_u tal que

$$G_u^T u_i = e_i. \quad (5.6)$$

Em seguida, aplicamos G_u^T nas linhas de \hat{R}_i e construímos G_v ortogonal, também por transformações de Householder, tal que

$$G_u^T \hat{R}_i G_v = \hat{R}_i^{up} \quad (5.7)$$

é uma matriz triangular superior. Então, se $\hat{R}_i = U\Sigma V^T$ é a decomposição SVD de \hat{R}_i , $\hat{R}_i^{up} = G_u^T U\Sigma V^T G_v$ é a decomposição SVD de \hat{R}_i^{up} . Segue que, para \hat{R}_i^{up} , u_i torna-se e_i e v_i é transformado em um novo vetor \hat{v}_i , e assim

$$\hat{R}_i^{up} e_i = \mu_i \hat{v}_i$$

e

$$\hat{R}_i^{up} \hat{v}_i = \mu_i e_i.$$

Dessa forma, podemos escrever

$$\hat{R}_i^{up} = \left(\begin{array}{cc} R_i & 0 \\ 0 & \mu_i \end{array} \right)$$

e a atualização da decomposição QR de A_k é dada por:

$$\left(\begin{array}{cc} Q_{i-1} & \hat{q}_i \end{array} \right) G_u G_u^T \left(\begin{array}{cc} R_{i-1} & r_i \\ 0 & \rho_i \end{array} \right) G_v = \left(\begin{array}{cc} Q_i & q_i \end{array} \right) \left(\begin{array}{cc} R_i & 0 \\ 0 & \mu_i \end{array} \right). \quad (5.8)$$

Proposição 5.3 *Para cada passo $i \geq k$ do método descrito acima, temos que $A_i V_i = Q_i R_i$ e $V_i^T V_i = I_k$.*

Demonstração: Faremos por indução sobre i . Note que para $i = k$ o

resultado é verdadeiro, e assumamos válido pra $i - 1$, isto é, $A_{i-1}V_{i-1} = Q_{i-1}R_{i-1}$ e $V_{i-1}^T V_{i-1} = I_k$.

Acrescentando a coluna a_i na matriz A_{i-1} temos,

$$A_i \begin{pmatrix} V_{i-1} & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} Q_{i-1}R_{i-1} & a_i \end{pmatrix},$$

além disso, usando (5.8) e (5.5) segue que,

$$A_i \begin{pmatrix} V_{i-1} & 0 \\ 0 & 1 \end{pmatrix} G_v = \begin{pmatrix} Q_i R_i & q_i \mu_i \end{pmatrix}.$$

Tomando as primeiras k colunas em ambos os lados da igualdade acima, tem-se $A_i V_i = Q_i R_i$ com

$$V_i = \begin{pmatrix} V_{i-1} & 0 \\ 0 & 1 \end{pmatrix} G_v \begin{pmatrix} I_k \\ 0 \end{pmatrix} \in \mathbb{R}^{i \times k}.$$

Pelo Primeiro Princípio de Indução, o resultado segue. □

Note que, a cada passo do algoritmo, construímos uma decomposição que exclui o vetor singular associado ao menor valor singular da matriz de ordem $m \times (k + 1)$ considerada, e acrescentamos a próxima coluna de A para ser analisada. Dessa forma, todas as colunas de A são consideradas, gerando apenas os vetores singulares associados aos maiores valores singulares de A , isto é, calculamos os subespaços singulares dominantes de ordem k de A . Mais informações sobre o método, incluindo análise de erros podem ser encontradas em [8].

O algoritmo é colocado a seguir.

Algoritmo 5.4: SVD Inexata

Sejam $A \in \mathbb{R}^{m \times n}$ e $k \leq n$ a dimensão do subespaço dominante procurado.

Passo 1. Calcule $A_k = Q_k R_k$ a decomposição QR de A_k e defina $V_k = I_k$. Faça $i = k$.

Passo 2. Defina $r_i = Q_i^T A_i$, $\hat{a}_i = a_i - Q_i r_i$, $\rho = \|\hat{a}_i\|$, $q_i = \frac{\hat{a}_i}{\rho}$ e faça

$$Q_i = (Q_{i-1} \ q_i),$$

e

$$R_i = \begin{pmatrix} R_{i-1} & r_i \\ 0 & \rho \end{pmatrix}.$$

Passo 3. Calcule u_i , o vetor singular associado ao menor valor singular de R_i .

Passo 4. Encontre as transformações ortogonais G_u e G_v conforme (5.6) e (5.7), respectivamente.

Passo 5. Defina

$$(\ Q_{i+1} \ q_{i+1} \) \begin{pmatrix} R_{i+1} & 0 \\ 0 & \mu_{i+1} \end{pmatrix} = (\ Q_i \ q_i \) G_u G_u^T \begin{pmatrix} R_i & r_i \\ 0 & \rho_i \end{pmatrix} G_v,$$

e

$$V_{i+1} = \begin{pmatrix} V_i & 0 \\ 0 & 1 \end{pmatrix} G_v I_k.$$

Faça $i = i + 1$ e volte ao Passo 2.

Na Tab. 10 exibimos o tempo, em segundos, gasto pelo Algoritmo 4.1 em sua forma original e ao calcularmos a SVD através do Algoritmo 5.4, este último será referenciado como SISVD. Para o Problema 1 não há uma redução significativa, pois são necessárias poucas avaliações da função, no entanto para os Problemas 2 e 3 podemos perceber uma melhora no desempenho de, em média, 21%.

Tabela 10: Comparativo entre os Algoritmos 4.1 e 5.4 para os Problemas 1, 2 e 3

q	Problema 1 m=5000		Problema 2 m=500		Problema 3 m=1000	
	GPST	SISVD	GPST	SISVD	GPST	SISVD
1	1.294	0.951	14.33	8.751	9.718	6.614
5	4.726	4.477	37.97	28.78	19.85	17.11
10	7.940	7.285	61.29	56.64	24.03	21.32

Considere agora o problema OPP com $q = 1$, isto é,

$$\begin{aligned} \min \quad & \|Ax - b\|_F \\ \text{s.a.} \quad & x^T x = 1, \end{aligned} \tag{5.9}$$

em que $A \in \mathbb{R}^{m \times m}$ e $x, b \in \mathbb{R}^m$. Note que este é um problema de mínimos quadrados com x satisfazendo a condição $\|x\| = 1$. Um algoritmo conhecido para resolver o problema (5.9) no caso irrestrito é o LSQR [34], que baseia-se na bidiagonalização de Golub-Kahan. Vamos aproveitar essa ideia e incorporar o processo de bidiagonalização ao Algoritmo 4.1 com o intuito de resolver problemas de grande porte. Neste caso, projetamos o problema (5.9) em um subespaço obtendo um problema equivalente, porém com dimensão menor.

A seguir apresentamos o algoritmo de bidiagonalização Golub-Kahan, conhecido na literatura por GKB.

5.4.3 Usando Bidiagonalização de Golub-Kahan

Nesta seção vamos colocar o processo de bidiagonalização de Golub-Kahan apresentado em [34]. Este processo baseia-se na tridiagonalização de Lanczos [27] e no processo de bidiagonalização de Golub e Kahan [19].

O algoritmo que vamos utilizar foi escrito baseado no método *Bidiag 1* de [34]. Apresentamos a seguir as ideias fundamentais para o desenvolvimento do mesmo.

O processo de Lanczos consiste em gerar, a partir de uma matriz simétrica $A \in \mathbb{R}^{n \times n}$ e um vetor $b \in \mathbb{R}^n$, seqüências de vetores $\{v_k\}$ e de escalares $\{\alpha_k\}$ e $\{\beta_k\}$ tais que A é reduzida à forma tridiagonal, isto é, $V^T A V = T$ com $T \in \mathbb{R}^{n \times n}$ tridiagonal e $V \in \mathbb{R}^{n \times n}$ satisfazendo $V^T V = I$. Apresentamos o processo no algoritmo seguinte.

Algoritmo 5.5: Processo de Lanczos

Tome $b \in \mathbb{R}^n$.

Passo 1. Faça $v_0 = 0$, $v_1 = b$, $\beta_1 = \|b\|_2$ e $k = 1$;

Passo 2. Defina $w_k = Av_k - \beta_k v_{k-1}$, $\alpha_k = v_k^T w_k$ e $v = w_k - \alpha_k v_k$;

Passo 3. Defina $\beta_{k+1} = \|v\|_2$, $v_{k+1} = v$, faça $k = k + 1$ e volte ao Passo 2.

Dessa forma, após k passos do Algoritmo 5.5 temos:

$$AV_k = V_k T_k + \beta_{k+1} v_{k+1} e_k^T, \quad (5.10)$$

em que, e_k é o k -ésimo vetor canônico de \mathbb{R}^n ,

$$V_k = [v_1, v_2, \dots, v_k] \in \mathbb{R}^{n \times k}$$

e

$$T_k = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ & \beta_2 & \alpha_2 & \ddots & \\ & & \ddots & \ddots & \beta_k \\ & & & \beta_k & \alpha_k \end{pmatrix} \in \mathbb{R}^{k \times k}.$$

Desconsiderando erros de precisão, o processo termina quando $k = n - 1$. Porém, se a dimensão do subespaço de Krylov de ordem n gerado por A e v_1 for menor do que n , o método pode terminar sem que a tridiagonalização esteja completa. A definição de subespaço de Krylov é colocada a seguir.

Definição 5.4 *O subespaço de Krylov de ordem r gerado por $A \in \mathbb{R}^{n \times n}$ e $b \in \mathbb{R}^n$ é o subespaço gerado pelas imagens de b sob as r primeiras potências de A , isto é,*

$$\mathcal{K}_r(A, b) = \text{span}\{b, Ab, A^2b, \dots, A^{r-1}b\}.$$

O próximo teorema, retirado de [20], estabelece o término do processo de Lanczos.

Teorema 5.5 *Seja $A \in \mathbb{R}^{n \times n}$ simétrica e suponha que $v_1 \in \mathbb{R}^n$ é um vetor unitário. Então, o processo de Lanczos termina quando $k = m$, em que $m = \dim(\mathcal{K}_n\{A, v_1\})$, e, para $k = 1, \dots, m$ vale (5.10).*

Demonstração: Ver Golub e Van Loan [20].

O processo de bidiagonalização de Golub-Kahan consiste em aplicar o processo de Lanczos a um sistema simétrico particular, a saber

$$\begin{pmatrix} I & A \\ A^T & -\lambda^2 I \end{pmatrix} \begin{pmatrix} b - Ax \\ x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix},$$

em que $\lambda \in \mathbb{R}$. O algoritmo de Golub-Kahan é apresentado a seguir.

Algoritmo 5.6: Bidiagonalização (GKB)

Tome $b \in \mathbb{R}^n$.

Passo 1. Faça $\beta_1 = \|b\|_2$, $\alpha_1 = \|A^T b\|_2$, $u_1 = \frac{b}{\beta_1}$ e $v_1 = \frac{A^T b}{\beta_1}$.

Passo 2. Defina $u = Av_k - \alpha_k u_k$, $\beta_{k+1} = \|u\|_2$, $u_{k+1} = \frac{u}{\beta_{k+1}}$.

Passo 3. Defina $v = A^T u_{k+1} - \beta_{k+1} v_k$, $\alpha_{k+1} = \|v\|_2$ e $v_{k+1} = \frac{v}{\alpha_{k+1}}$. Faça $k = k + 1$ e volte ao Passo 2.

Após k passos do Algoritmo 5.6 temos,

$$AV_k = U_{k+1}B_k \quad (5.11)$$

e

$$A^T U_{k+1} = V_k B_k + \alpha_{k+1} v_{k+1} e_{k+1}^T, \quad (5.12)$$

em que,

$$U_k = [u_1, u_2, \dots, u_k] \in \mathbb{R}^{m \times k},$$

$$V_k = [v_1, v_2, \dots, v_k] \in \mathbb{R}^{n \times k}$$

e

$$B_k = \begin{pmatrix} \alpha_1 & & & & \\ \beta_2 & \alpha_2 & & & \\ & \beta_3 & \ddots & & \\ & & \ddots & \alpha_k & \\ & & & \beta_{k+1} & \end{pmatrix} \in \mathbb{R}^{k+1 \times k},$$

satisfazendo $U_{k+1}^T U_{k+1} = I$ e $V_k^T V_k = I$.

Vamos aplicar esse processo de bidiagonalização para resolver o problema de Procrustes (5.9).

Sejam $B_k \in \mathbb{R}^{k+1 \times k}$, $V_k \in \mathbb{R}^{n \times k}$ e $U_{k+1} \in \mathbb{R}^{m \times k+1}$ geradas após k iterações do Algoritmo 5.6, então

$$AV_k = U_{k+1}B_k.$$

Definindo $x = V_k y$, temos que

$$\|Ax - b\|_F = \|AV_k y - b\|_F = \|U_{k+1} B_k y - b\|_F = \|B_k y - U_{k+1}^T b\|_F,$$

com $y = V_k^T x \in \mathbb{R}^k$.

Além disso,

$$y^T y = 1 \Leftrightarrow x^T x = 1.$$

Portanto,

$$\begin{aligned} \min \quad & \|Ax - b\|_F \\ \text{s.a.} \quad & \|x\| = 1 \\ & x \in \text{span}\{V_k^1, \dots, V_k^k\} \end{aligned}$$

em que V_k^i é a i -ésima coluna de V_k , é equivalente a

$$\begin{aligned} \min \quad & \|B_k y - U_{k+1}^T b\|_F \\ \text{s.a.} \quad & \|y\| = 1. \end{aligned} \tag{5.13}$$

Aplicamos então o Algoritmo 4.1 ao problema (5.13), cuja dimensão é menor do que a dimensão de (5.9).

Note que podemos considerar também restrições do tipo $\|x\| = \omega$, com $\omega > 0$, no problema (5.9). Esse problema aparece frequentemente quando trabalhamos com métodos de região de confiança. Para resolvê-lo basta fazermos a mudança de variável $z = \frac{x}{\omega}$ e considerar o problema

$$\begin{aligned} \min \quad & \|Az - b\| \\ \text{s.a.} \quad & \|z\| = 1. \end{aligned}$$

Faremos os testes para as matrizes dos Problemas 1 e 4 considerando os valores: $\omega = 10^{-1}$, $\omega = 1$ e $\omega = 2$. As constantes m_1 , m_2 , m_3 e m_4 do Problema 4 foram consideradas as do item 4.2 multiplicadas por 10. Omitimos os testes para os Problemas 2 e 3 pois o processo de bidiagonalização não reduziu a dimensão do problema.

O Algoritmo 4.1 com o processo de bidiagonalização será denominado SGKB. Na Tab. 11 expomos os resultados obtidos ao aplicarmos o SGKB com parâmetro de não monotonia $M = 7$, DSP representa a dimensão do subespaço projetado, dado pelo número de iterações do Algoritmo 5.6. Para cada um dos problemas testados, apresentamos apenas os resultados obtidos para um valor de m , visto que o número de iterações do processo de bidiagonalização não sofreu alteração relevante.

Para o caso $\omega = 1$, isto é, para o problema OPP, o método SGKB apresentou resultados promissores. Foram necessários poucos passos

do processo de bidiagonalização e, portanto, a redução na dimensão do problema foi significativa. Além disso, o método convergiu rapidamente para um ponto muito próximo da solução. Para o caso $\omega = 2$ o número de bidiagonalizações também foi satisfatório, no entanto a precisão do algoritmo foi prejudicada. O caso $\omega = 10^{-1}$ utilizou mais iterações do processo de bidiagonalização o que atrapalhou o desempenho do algoritmo.

Tabela 11: Desempenho do método SGKB utilizando as matrizes dos Problema 1 e 4

Problema 1 com $m = 1000$						
ω	N_i	N_a	Tempo	Resíduo	Erro	DSP
1e-1	571	753	0.468	2.96e-02	2.61e-03	21
1	67	110	0.187	7.12e-09	6.85e-10	8
2	40	68	0.046	1.45e-02	1.27e-03	6
Problema 4 com $m = 950$						
1e-1	4861	5599	9.734	9.42e-01	1.33e-01	63
1	646	729	1.52	1.87e-003	1.87e-02	18
2	47	63	0.436	2.16e-01	6.23e-02	7

5.4.4 Cálculo dos menores autovalores de uma matriz simétrica

Nesta seção, mostramos que o problema de encontrar o subespaço invariante associado aos menores autovalores de uma dada matriz pode ser escrito como um problema de programação não linear.

A necessidade de calcular o subespaço invariante principal de uma matriz simétrica (ou, no caso complexo, Hermitiana) aparece em problemas de processamento de sinal (por exemplo, em radares de vigilância aérea [43]), dinâmica estrutural (por exemplo, problemas de flambagem [36]), entre outros.

Dada uma matriz $A \in \mathbb{R}^{n \times n}$ simétrica, defina $f : \mathbb{R}^{n \times q} \rightarrow \mathbb{R}$ por

$$f(X) = \frac{1}{2} \text{traço}(X^T A X)$$

e considere o problema de programação não linear

$$\begin{aligned} \min \quad & f(X) \\ \text{s.a.} \quad & X \in \Omega, \end{aligned} \tag{5.14}$$

em que, $\Omega = \{X \in \mathbb{R}^{n \times q} : X^T X = I\}$.

Como A é simétrica, $g(X) = \nabla f(X) = AX$ e, portanto, as condições KKT do problema (5.14) são

$$\begin{cases} AX - X\Theta = 0 \\ X \in \Omega \end{cases}, \quad (5.15)$$

em que $\Theta \in \mathbb{R}^{q \times q}$ é a matriz formada pelos multiplicadores de Lagrange θ_{ij} do problema.

No teorema a seguir mostramos que encontrar uma base para o subespaço gerado pelos autovetores associados aos q menores autovalores de A equivale a encontrar a solução global do problema (5.14).

Teorema 5.6 *Seja $A \in \mathbb{R}^{n \times n}$ simétrica e $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ seus autovalores. Então, uma solução do problema (5.14) é,*

$$X = \begin{pmatrix} | & | & \cdots & | \\ v_1 & v_2 & \cdots & v_q \\ | & | & \cdots & | \end{pmatrix},$$

em que cada v_i é o autovetor associado ao autovalor λ_i .

Demonstração: Como A é simétrica, seus autovetores são ortogonais e assim

$$\begin{cases} AX = X\Lambda \\ X \in \Omega \end{cases},$$

em que $\Lambda \in \mathbb{R}^{q \times q}$ é matriz diagonal formada pelos autovalores λ_i , $i = 1, \dots, q$. Ou seja, X satisfaz as condições KKT de (5.14).

Suponha, por contradição, que X não é minimizador global de (5.14). Então, existe $Y \in \mathbb{R}^{n \times q}$ satisfazendo (5.15) tal que

$$\frac{1}{2}\text{traço}(Y^T AY) < \frac{1}{2}\text{traço}(X^T AX). \quad (5.16)$$

Logo, existe $\Theta \in \mathbb{R}^{q \times q}$ formada pelos multiplicadores de Lagrange θ_{ij} tal que

$$\begin{cases} AY = Y\Theta \\ Y \in \Omega \end{cases}. \quad (5.17)$$

Note que Θ é simétrica e, portanto, diagonalizável. Isto é, existem matrizes $U \in \mathbb{R}^{q \times q}$ ortogonal e $M \in \mathbb{R}^{q \times q}$ diagonal tais que

$$U^T \Theta U = M.$$

Considere $Y' = YU$. É fácil ver que $f(Y') = f(Y)$ e $Y' \in \Omega$. Multiplicando (5.17) por U , lembrando que U é ortogonal, tem-se

$$AYU = Y\Theta U = YUU^T\Theta U = Y'M.$$

Logo,

$$\begin{cases} AY' = Y'M \\ Y' \in \Omega \end{cases},$$

ou seja, Y' é ponto estacionário de (5.14). Note que as colunas de Y' são formadas por q autovetores de A , e que para $i = 1, \dots, q$, $X_i^T A X_i = \lambda_i$ e $Y_i'^T A Y_i' = \mu_i$, com μ_i e λ_i autovalores de A sendo $\{\lambda_i\}_{i=1}^q$ os menores. Logo,

$$f(X) = \frac{1}{2} \sum_{i=1}^q \lambda_i \leq \frac{1}{2} \sum_{i=1}^q \mu_i = f(Y') = f(Y),$$

o que contradiz (5.16). Portanto, X é um minimizador global. \square

Note que,

$$\|g(Y) - g(X)\|_F = \|A(Y - X)\|_F \leq \|A\|_F \|Y - X\|_F,$$

isto é, g é Lipschitz contínua com constante de Lipschitz satisfazendo

$$L_f \leq \|A\|_F.$$

Portanto, podemos aplicar o Algoritmo 4.1 para resolver o problema (5.14). Além disso, como f é uma quadrática pelo Teorema 4.4 é conveniente usarmos o parâmetro espectral de Barzilai-Borwein.

Para os Problemas 5, 6 e 7 a matriz A será definida da seguinte maneira, $A = UDU^T$ com U ortogonal gerada de maneira randômica e D diagonal, definida de modo diferente para cada um dos problemas.

Problema 5. Os elementos da diagonal de D são gerados através de uma distribuição normal no intervalo $[10, 12]$.

Note que este problema é bem condicionado. Além disso, por conter autovalores muito próximos, esperamos um mau desempenho do método da iteração ortogonal inversa (nas tabelas, citado como Método da Potência).

A Tab. 12 apresenta o desempenho dos métodos na busca dos q menores autovalores de A para diferentes valores de n . Novamente,

reservamos a notação N_i e N_a para o número de iterações e avaliações da função, respectivamente. Denotamos por $\text{Erro}(\lambda)$ a diferença $|f(X) - f(X^*)|$, e por $\text{Erro}(X)$ a norma $\|X - X^*\|_F$, em que X representa a solução encontrada e X^* os autovetores de A .

Tabela 12: Comparativo entre os métodos para o Problema 5

$n = 100$ e $q = 1$					
Método	N_i	N_a	Tempo	$\text{Erro}(\lambda)$	$\text{Erro}(X)$
GPST	66	75	0.031	1.19e-11	2.51e-05
OptStiefel	65	66	0.046	1.59e-14	8.22e-07
sg_min	6	13	0.234	7.10e-14	1.57e-06
Potência	2872	2872	5.335	1.32e-11	2.64e-05
$n = 500$ e $q = 1$					
GPST	120	142	0.265	7.28e-11	1.90e-04
OptStiefel	99	103	0.234	1.59e-10	2.44e-04
sg_min	6	13	0.936	3.53e-13	1.05e-05
Potência	6526	6526	223.1	7.85e-11	1.82e-04
$n = 1000$ e $q = 6$					
GPST	323	380	3.775	3.23e-10	6.57e-04
OptStiefel	110	114	1.232	3.80e-09	2.22e-03
sg_min	8	16	7.550	8.88e-13	3.28e-05
Potência	-	-	TMAX	-	-
$n = 3000$ e $q = 6$					
GPST	284	345	25.89	2.46e-10	6.04e-04
OptStiefel	110	116	8.174	2.24e-09	1.33e-03
sg_min	7	14	54.55	5.38e-12	1.21e-04
Potência	-	-	TMAX	-	-

Tanto na Fig. 8 quanto na Tab. 13 podemos verificar as vantagens de utilizar um esquema não monótono, neste caso com uma redução significativa no número de iterações e de passos da busca linear.

Problema 6. Os elementos da diagonal de D são definidos por:

$$D_{ii} = 2 \cos\left(\frac{i\pi}{10}\right).$$

Este problema, além de ser extremamente mal condicionado (nos

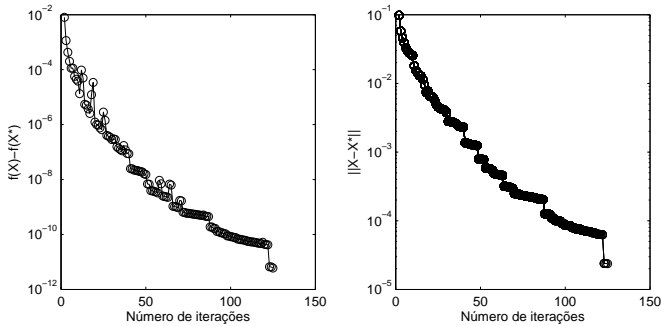


Figura 8: Comportamento do Algoritmo 4.1 para o Problema 5 com $n = 500$ e $q = 1$

Tabela 13: Comparativo entre os casos monótono e não monótono do Algoritmo 4.1 para o Problema 5

Parâmetro	n=100 e q=1		n=500 e q=1	
	N_i	N_a	N_i	N_a
M = 0	250	302	514	627
M = 5	78	93	200	248
M = 7	66	75	120	142
M = 10	63	69	106	123
M = 15	63	66	117	133

casos testados, $\mathcal{K}(A) > 10^{15}$), possui autovalores repetidos. Este último impediu a convergência do Método da Potência. No entanto, podemos verificar na Tab. 14 que os demais métodos testados apresentaram bons resultados e, para dimensões elevadas o método GPST foi o que apresentou o melhor desempenho.

Como podemos notar na Fig. 9 o método GPST gerou uma sequência monótona de valores da função objetivo, excluindo portanto a eficácia de uma técnica não monótona. Na mesma figura expomos o comportamento da sequência gerada pelo método OptStiefel, note que para este algoritmo a não monotonia apresentou algum resultado.

Problema 7. Os elementos da diagonal de D são gerados através de

Tabela 14: Comparativo entre os métodos para o Problema 6

$n = 100$ e $q = 1$					
Método	N_i	N_a	Tempo	Erro(λ)	Erro(X)
GPST	191	192	0.093	1.38e-12	1.67e-02
OptStiefel	44	46	0.031	1.85e-13	1.67e-02
sg_min	6	13	0.109	4.44e-16	1.67e-02
Potência	1	1	1.076	2	1.41
$n = 500$ e $q = 1$					
GPST	178	179	0.452	1.23e-12	1.23e-02
OptStiefel	37	38	0.140	3.82e-13	1.23e-02
sg_min	5	11	0.436	0	1.23e-02
Potência	1	1	0.374	2	1.41
$n = 1000$ e $q = 6$					
GPST	200	201	2.152	1.35e-12	3.51e-02
OptStiefel	41	43	0.483	5.70e-12	3.65e-02
sg_min	6	13	2.636	7.10e-15	3.67e-02
Potência	1	1	0.499	1.20e+01	3.46
$n = 3000$ e $q = 6$					
GPST	192	193	1.383	1.18e-12	3.06e-02
OptStiefel	40	42	3.222	6.01e-11	2.64e-02
sg_min	6	12	15.35	5.32e-15	3.15e-02
Potência	1	1	3.712	1.20e+01	3.46

uma distribuição normal no intervalo $[-10, 10]$.

Novamente, os autovalores da matriz A encontram-se muito próximos um dos outros, prejudicando a convergência do Método da Potência. Omitimos a tabela comparativa entre o caso monótono e não monótono do método GPST, pois para este problema a técnica não apresentou resultado relevante.

A Fig. 10 mostra o comportamento dos métodos OptStiefel e sg_min para este problema com $n = 1000$ e $q = 6$. Note que o uso do esquema não monótono no Algoritmo 5.3 mostrou-se eficaz, o que pode ser novamente confirmado nos dados da Tab. 15. Na mesma tabela, percebemos que o método sg_min, apesar de convergir em poucas iterações, teve um custo computacional superior ao OptStiefel.

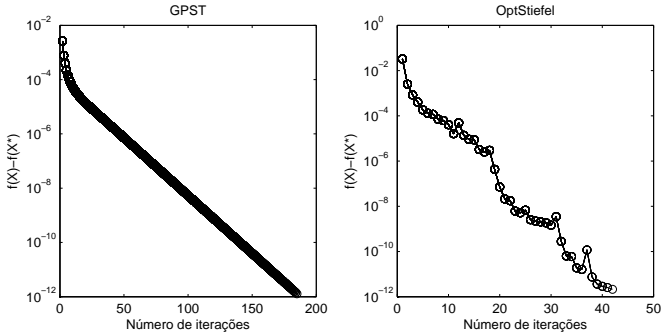


Figura 9: Comportamento dos Algoritmos 4.1 e 5.3 para o Problema 6 com $n = 100$ e $q = 1$

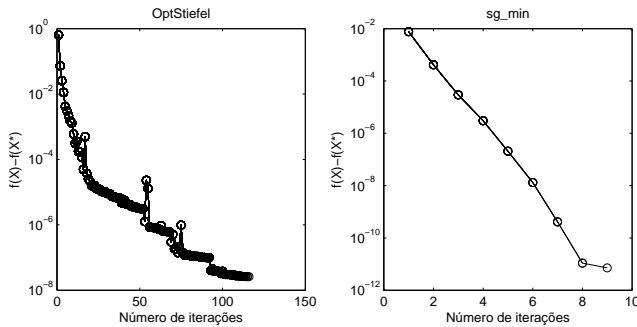


Figura 10: Comportamento dos Algoritmos 5.3 e 5.2 para o Problema 7 com $n = 1000$ e $q = 6$

Problema 8. Considere a matriz $A \in \mathbb{R}^{n \times n}$ dada por,

$$A = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & & -1 & 2 \\ & & & -1 & 2 \end{pmatrix}.$$

Esta matriz aparece frequentemente em problemas aplicados, especialmente nos que envolvem equações diferenciais. Seus autovalores

Tabela 15: Comparativo entre os métodos para o Problema 7

$n = 100$ e $q = 1$					
Método	N_i	N_a	Tempo	Erro(λ)	Erro(X)
GPST	305	306	0.124	5.29e-13	1.46e-06
OptStiefel	57	59	0.062	1.04e-11	6.05e-06
sg_min	6	12	0.156	3.90e-14	3.90e-07
Potência	14	14	0.031	9.65	1.414
$n = 500$ e $q = 1$					
GPST	1264	1265	2.386	2.07e-12	5.84e-06
OptStiefel	96	98	0.187	4.28e-11	2.64e-05
sg_min	7	15	1.092	1.26e-13	1.28e-06
Potência	21	21	1.014	9.96	1.41
$n = 1000$ e $q = 6$					
GPST	975	976	11.49	1.98e-12	5.48e-06
OptStiefel	98	100	1.060	1.50e-09	3.46e-06
sg_min	8	16	5.538	7.10e-13	3.14e-06
Potência	324	324	64.03	59.57	3.46
$n = 3000$ e $q = 6$					
GPST	5008	5009	446.3	2.10e-11	5.97e-05
OptStiefel	88	90	12.69	1.73e-08	1.17e-03
sg_min	8	16	88.76	2.20e-12	1.94e-05
Potência	74	74	271.4	5.99e+01	3.46

e autovetores podem ser expressos da seguinte maneira:

$$\lambda_i = 2 - 2 \cos\left(\frac{i\pi}{n+1}\right)$$

e

$$X_{ij} = \sin\left(\frac{ij\pi}{n+1}\right),$$

em que $i, j = 1, \dots, n$.

Os resultados envolvendo os esquemas monótono e não monótono no método GPST podem ser verificados na Tab. 16. Note que, para este problema a técnica de não monotonia reduziu consideravelmente o número de passos da busca linear.

O método da Potência apresentou ótima convergência, mesmo ao aumentarmos a dimensão do problema, caso onde os demais métodos

Tabela 16: Comparativo entre os casos monótono e não monótono do Algoritmo 4.1 para o Problema 8

Parâmetro	m=100 e q=1		m=500 e q=1	
	\tilde{N}_i	N_a	\tilde{N}_i	N_a
M = 0	484	781	2811	4910
M = 5	396	543	2287	3485
M = 7	353	483	1716	2492
M = 10	442	586	2173	3013
M = 15	514	648	3103	4101

ou atingiram o tempo máximo permitido, ou exigiram elevado custo computacional. Este e outros resultados podem ser verificados na Tab. 17.

Exibimos na Fig. 11 o comportamento dos métodos GPST e Potência para o problema com $n = 100$ e $q = 1$. Note que, a cada passo do método da Potência o valor da função reduz consideravelmente, enquanto que, para atingir o mesmo valor o método GPST necessita um número maior de iterações.

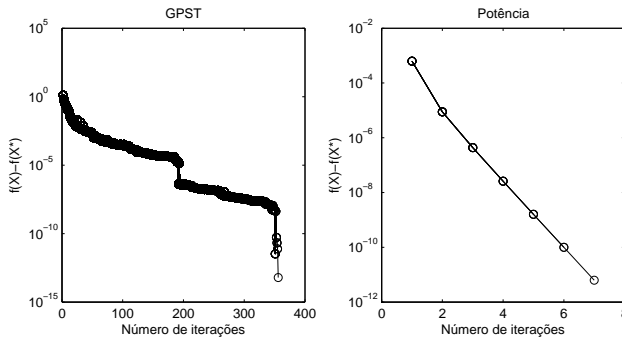


Figura 11: Comportamento dos Algoritmos 4.1 e 5.1 para o Problema 8 com $n = 100$ e $q = 1$

A seguir comparamos o método GPST e SISVD para os problemas 5, 6 e 7 com $n = 1000$ e $q = 1, 3$, ou 6. O cálculo da SVD via Algoritmo 5.4 apresentou uma redução de, em média, 23% do custo computacional, conforme pode ser observado na Tab. 18.

Tabela 17: Comparativo entre os métodos para o Problema 8

$n = 100$ e $q = 1$					
Método	N_i	N_a	Tempo	Erro(λ)	Erro(X)
GPST	353	483	0.171	7.87e-11	1.64e-04
OptStiefel	296	311	0.171	4.14e-10	3.76e-04
sg_min	15	31	0.592	8.41e-13	1.33e-05
Potência	8	8	0.031	6.68e-12	4.79e-05
$n = 500$ e $q = 1$					
GPST	1716	2492	2.683	1.84e-09	3.94e-03
OptStiefel	1020	1061	1.341	8.59e-08	2.70e-02
sg_min	17	35	3.120	1.50e-11	2.62e-04
Potência	4	4	0.109	3.90e-10	1.81e-03
$n = 1000$ e $q = 6$					
GPST	3675	5316	61.52	4.39e-10	1.85e-03
OptStiefel	1876	1959	22.16	1.62e-07	3.14e-02
sg_min	37	75	527.0	3.34e-11	3.46e-02
Potência	24	24	2.995	1.46e-09	3.38e-03
$n = 3000$ e $q = 6$					
GPST	-	-	TMAX	-	-
OptStiefel	4014	4142	318.61	2.02e-05	1.48
sg_min	-	-	TMAX	-	-
Potência	14	14	30.34	1.73e-08	3.49e-02

Tabela 18: Comparativo entre os Algoritmos 4.1 e SISVD para os Problemas 5, 6 e 7

k	Problema 5		Problema 6		Problema 7	
	GPST	SISVD	GPST	SISVD	GPST	SISVD
1	1.092	0.514	1.279	0.686	7.098	5.070
3	5.163	4.680	2.527	2.121	33.07	31.65
6	4.804	3.962	3.229	2.854	11.57	9.235

6 CONSIDERAÇÕES FINAIS

Neste trabalho, apresentamos um método para resolver problemas de minimização em conjuntos fechados com convergência global para pontos estacionários. Em seguida, aplicamos o algoritmo no problema de minimização com restrições de ortogonalidade, dando origem a um método do tipo gradiente projetado para minimização com restrições de ortogonalidade. Ressaltamos que esse tipo de restrição aparece em importantes problemas de otimização.

Com intuito de verificar os resultados teóricos e analisar o desempenho do algoritmo apresentado, aplicamos o mesmo nos problemas WOPP e no cálculo de uma base para o subespaço dominante de uma matriz simétrica. Apesar da programação ter sido feita sem rigor numérico, os resultados alcançados foram promissores quando comparados com os obtidos via métodos usuais, mostrando que o método estudado é competitivo com métodos já consolidados.

Um estudo detalhado da técnica de busca linear não monótona foi importante para uma compreensão inicial de seu funcionamento e eficácia. Verificamos que, para os problemas em que a Condição de Armijo não é satisfeita no primeiro passo, utilizar não monotonia implica em uma redução considerável no número de iterações e, conseqüentemente, de avaliações da função.

Ao algoritmo, incorporamos ideias de bidiagonalização e cálculo da SVD de uma maneira alternativa o que mostrou uma melhora significativa no desempenho do algoritmo, quando utilizada para problemas de mínimos quadrados, ou problemas em que o número de linhas é muito maior do que o número de colunas, respectivamente. Ressaltamos que pode-se aproveitar melhor os benefícios do Algoritmo 5.4, por exemplo, calculando os valores singulares dominantes da matriz, ao invés de calcular todos eles como foi feito no trabalho. Essa estratégia gera pontos não viáveis, porém pode-se restaurar a viabilidade dos pontos com o complemento ortogonal dos mesmos. Incorporar essa técnica no algoritmo fica como inspiração para estudos futuros.

Estamos cientes que problemas práticos podem chegar a dimensões muito maiores do que as aqui consideradas, mas lembramos que os exemplos testados correspondem à limitação dos computadores utilizados.

Planos futuros estabelecem-se com intuito de:

- Aplicar o Algoritmo 3.1 (Algoritmo Modelo) para resolver outros problemas, como por exemplo

$$\begin{aligned} \min \quad & f(X) \\ \text{s.a.} \quad & X^T = X \\ & X \in \mathbb{R}^{n \times n}; \end{aligned}$$

- Usar o parâmetro $\eta < 1$ no Algoritmo 4.1 para reduzir o esforço computacional;
- Aplicar o método GKB para casos onde o número de colunas da matriz X é maior do que um.

REFERÊNCIAS

- [1] T. A. Arias, A. Edelman, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.
- [2] A. Aspremont, L. E. Ghaoui, M. Jordan, and G. R. Lanckriet. A direct formulation for sparse pca using semidefinite programming. *SIAM REVIEW*, (49):434–448, 2007.
- [3] J. Barzilai and J. M. Borwein. Two point step size gradient methods. *IMA Journal of Numerical Analysis*, 8:141–148, 1988.
- [4] T. Bell. Global positioning system-based attitude determination and the orthogonal procrustes problem. *Journal of guidance, control and dynamics*, 26(5):820–822, 2003.
- [5] E. G. Birgin, J. M. Martínez, and M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, 10:1196–1211, 2000.
- [6] A. W. Bojanczyk and A. Lutoborski. The procrustes problem for orthogonal stiefel matrices. *SIAM Journal on Scientific Computing*, 21:1291–1304, 1998.
- [7] M. R. Celis, J. E. Dennis, and R. A. Tapia. A trust region strategy for nonlinear equality constrained optimization. *Numerical Optimization*, pages 71–82, 1984.
- [8] Y. Chahlaoui, K. Gallivan, and P. Van Dooren. Recursive calculation of dominant singular subspaces. *SIAM Journal on Matrix Analysis and Applications*, 25(2):445–463, 1999.
- [9] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver. *Combinatorial optimization*. John Wiley & Sons, Canadá, 1997.
- [10] T. F. Cox and M. A. A. Cox. *Multidimensional scaling*. Chapman and Hall, Londres, 2000.
- [11] Y. H. Dai. On the nonmonotone line search. *Journal of Optimization Theory and Application*, 112(2):315–330, 2002.

- [12] E. de Klerk, M. Laurent, and P. A. Parrilo. A ptas for the minimization of polynomials of fixed degree over the simplex. *Theoretical Computer Science*, pages 210–225, 2006.
- [13] J. E. Dennis. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, Philadelphia, 1996.
- [14] A. Edelman and R. Lippert. Nonlinear eigenvalue problems with orthogonality constraints. In *Templates for the Solution of Algebraic Eigenvalue Problems, A Practical Guide*, pages 290–314, Philadelphia, 2000.
- [15] L. Eldén and H. Park. A procrustes problem on stiefel manifolds. *Numerische Mathematik*, 82:599–619, 1999.
- [16] J. B. Francisco, J. M. Martínez, and L. Martínez. Globally convergent trust-region methods for self-consistent fiel electronic structure calculations. *Journal of Chemical Physics*, 121:10863–10878, 2004.
- [17] J. B. Francisco and F. S. Viloche Bazán. Nonmonotone algorithm for minimization on closed sets with application to minimization on stiefel manifolds. *Journal of Computational and Applied Mathematics*, No prelo 2012.
- [18] J. Fu and W. Sun. Nonmonotone adaptive trust-region method for unconstrained optimization problems. *Applied Mathematics and Computation*, 163:489–504, 2005.
- [19] G. H. Golub and W. Kahan. Calculating the singular values and pseudoinverse of a matrix. *SIAM Journal on Numerical Analysis*, 2:205–224, 1965.
- [20] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Londres, 1996.
- [21] L. Grippo, F. Lampariello, and S. Lucidi. A nonmonotone line search technique for newton’s method. *SIAM Journal on Numerical Analysis*, 23(4):707–716, 1986.
- [22] L. Grippo, F. Lampariello, and S. Lucidi. A truncated newton method with nonmonotone line search for unconstrained optimization. *Journal of Optimization Theory and Applications*, 60(3):401–419, 1989.

- [23] J. R. Hurley and R. B. Cattell. The procrustes program: Producing direct rotation to test a hypothesized factor structure. *Behavioral Science*, (7):258–262, 1962.
- [24] R. Janin. Direction derivative of the marginal function in nonlinear programming. *Mathematical Programming Study*, 21:127–138, 1984.
- [25] W. Karush. Minima of functions of several variables with inequalities as side constraints. *Universidade de Chicago*, 1939.
- [26] H.W. Kuhn and A.W. Tucker. Nonlinear programming. *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, 1951.
- [27] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45(4):255–282, 1950.
- [28] A. Leon. A comparison among eight known optimizing procedures. *Internal Working Paper, Space Sciences Laboratory, University of California*, (20), 1964.
- [29] D. G. Luenberger. *Introduction to linear and nonlinear programming*. Addison-Wesley Publishing, 1973.
- [30] Jerrold E. Marsden. *Elementary Classical Analysis*. W.H. Freeman and Company, San Francisco, 1974.
- [31] J. M. Martínez and S. A. Santos. *Métodos Computacionais de Otimização*. IMPA, Rio de Janeiro, 1995.
- [32] J. M. Martínez and S. A. Santos. A trust region strategy for minimization on arbitrary domains. *Mathematical Programming*, (68):267–302, 1995.
- [33] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2006.
- [34] C. C. Paige and M. A. Saunders. Lsqr: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, 8(1):43–71, 1982.

- [35] M. J. D. Powell and Y. Yuan. A trust region algorithm for equality constrained optimization. *Mathematical Programming*, (49):189–211, 1990.
- [36] J. Purbolaksono and M. H. Aliabadi. Application of drbem for evaluating buckling problems of perforated thin plates. *European Journal of Scientific Research*, 31(3):398–408, 2009.
- [37] M. Raydan. The barzilai and borwein gradient method for the large scale unconstrained minimization problem. *SIAM Journal Optimization*, 7(1):26–33, 1997.
- [38] H. H. Rosenbrock. An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3:175–184, 1960.
- [39] I. Söderkvist and P. Wedin. On condition numbers and algorithms for determining a rigid body movement. *BIT*, 34:424–436, 1994.
- [40] E.L. Stiefel. *Richtungsfelder und Fernparallelismus in n-dimensionalen Mannigfaltigkeiten*. Eidgen. technischen Hochschule, 1935.
- [41] Wenyu Sun. Nonmonotone trust region method for solving optimization problems. *Applied Mathematics and Computations*, 156:159–174, 2004.
- [42] P. L. Toint. An assessment of non-monotone linesearch techniques for unconstrained optimization. *SIAM Journal on Scientific Computing*, 17:725–739, 1996.
- [43] J. Ward. Space-time adaptive processing for airborne radar. *Technical report of MIT*, 1994.
- [44] Z. Wen and W. Yin. A feasible method for optimization with orthogonality constraints. *Optimization Online*, pages 1–30, 2010.
- [45] G. Yuan and L. Yu. A nonmonotone line search method for symmetric nonlinear equations. *Intelligent Control and Automation*, 1:28–35, 2010.
- [46] H. Zhang and W. W. Hager. A nonmonotone line search technique and its application to unconstrained optimization. *SIAM Journal Optimization*, 14(4):1043–1056, 2004.

- [47] Z. Zhang and K. Du. Successive projection method for solving the unbalanced procrustes problem. *Science in China: Series A Mathematics*, 49(7):971–986, 2006.