Universidade Federal de Santa Catarina
Centro de Ciências Físicas e Matemáticas
Curso de Pós-Graduação em Matemática Pura e Aplicada

Método de Restauração Inexata Aplicado ao Problema de Minimização com Restrições de Ortogonalidade

Lila Lisbeth Tenorio Paredes

Orientador: Prof. Dr. Fermín Sinforiano Viloche Bazán

Coorientador: Prof. Dr. Juliano de Bem Francisco

Florianópolis, 15 de Março de 2018

Universidade Federal de Santa Catarina
Curso de Pós-Graduação em Matemática
Pura e Aplicada

Método de Restauração Inexata Aplicado ao Problema de Minimização com Restrições de Ortogonalidade

Tese apresentada ao Programa de Pós-Graduação em Matemática Pura e Aplicada, do Centro de Ciências Físicas e Matemáticas da Universidade Federal de Santa Catarina, para a obtenção do Grau de Doutora em Matemática Pura e Aplicada, com área de concentração em Otimização.
Orientador: Prof. Dr. Fermín S. Viloche Bazán
Coorientador: Prof. Dr. Juliano de Bem Francisco

Lila Lisbeth Tenorio Paredes

Florianópolis, 15 de Março de 2018

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Tenorio Paredes, Lila Lisbeth
Método de Restauração Inexata Aplicado ao Problema de Minimização com Restrições de Ortogonalidade / Lila Lisbeth Tenorio Paredes ; orientador, Fermín Sinforiano Viloche Bazán, coorientador, Juliano de Bem Francisco, 2018.
104 p.

Tese (doutorado) - Universidade Federal de Santa Catarina, Centro de Ciências Físicas e Matemáticas, Programa de Pós-Graduação em Matemática Pura e Aplicada, Florianópolis, 2018.

Inclui referências.

1. Matemática Pura e Aplicada. 2. Restauração Inexata. 3. não monótono. 4. restrições de ortogonalidade. 5. transformação de Cayley. I. Viloche Bazán, Fermín Sinforiano. II. Francisco, Juliano de Bem. III. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Matemática Pura e Aplicada. IV. Título.

Método de Restauração Inexata Aplicado ao Problema de Minimização com Restrições de Ortogonalidade

por

Lila Lisbeth Tenorio Paredes

Esta Tese foi julgada para a obtenção do Título de "Doutora em Matemática Pura e Aplicada", área de Concentração em Otimização, e aprovada em sua forma final pelo Curso de Pós-Graduação em Matemática Pura e Aplicada.

Prof. Dr. Ruy Coimbra Charão
Coordenador

Comissão Examinadora:

Prof. Dr. Fermín Sinforiano Viloche Bazán
(Orientador - UFSC)

Profa. Dra. Sandra Augusta Santos (UNICAMP)

Prof. Dr. Roger Behling (UFSC-Blumenau)

Prof. Dr. Douglas Soares Gonçalves (UFSC)

Prof. Dr. Luciano Bedin (UFSC)

Florianópolis, 15 de Março de 2018.

Agradecimentos

Agradeço em primeiro lugar a Deus, quem deu-me força e coragem para seguir em frente. Deu graças a Ele por ter colocado pessoas maravilhosas durante minha morada no Brasil.

Ao meu orientador, Prof. Dr. Fermín S. Viloche Bazán, pela dedicação, pelos ensinamentos e pela amizade construída ao longo destes anos.

Ao meu coorientador, Prof. Dr. Juliano de Bem Francisco, pela paciência, pela dedicação, pelas conversas e pelos ensinamentos que levarei pra o resto da minha vida.

Ao meus pais e minha irmã, Venancio, Bertha e Violeta, por sempre me apoiarem e motivaram para conquistar meus objetivos, mesmo estando tão longe de casa.

A meu noivo, Jonathan, pelo apoio em todo tempo, por sempre acreditar em mim e em minha capacidade.

A minha amiga e colega, Samara, pelos momentos compartilhados dentro e fora da universidade.

A todos os meus colegas da pós graduação pela hospitalidade, ajuda e amizade.

A secretária, Elisa, pela atenção recebida durante o meu período de estudos.

A CAPES pelo auxilio financeiro durante o doutorado.

Resumo

Neste trabalho, apresentamos e estudamos o Algoritmo de Restauração Inexata não monótono para resolver problemas de minimização com restrições de ortogonalidade, que combina o método de Restauração Inexata de Fischer e Friedlander [1] e o critério de não monotonia de Zhang e Hager [2]. Desenvolvemos as ferramentas teóricas para caracterizar o subespaço tangente do conjunto viável, o qual nos permite descrever o Algoritmo proposto. Mostramos, sob certas hipóteses, a boa definição do Algoritmo assim com a convergência global a pontos viáveis do problema. O método de Restauração Inexata é um método iterativo que consta de duas fases: viabilidade e otimalidade. Neste trabalho a fase de viabilidade será feita de forma exata utilizando a transformação de Cayley. Portanto, sequência de pontos restaurados pertencem ao conjunto viável. Na fase de otimalidade, as direções de descida podem ser obtidas da seguintes maneiras: o gradiente espectral projetado ou a minimização de uma aproximação quadrática para o Lagrangiano restrito ao subespaço tangente. Para resolver este ultimo problema utilizamos o método de gradiente conjugado [3]. A implementação computacional do algoritmo proposto é realizado no *software* MATLAB e é comparado com o método de Wen e Yin [4] e com o método Gradiente Conjugado do pacote ManOpt [5] para diferentes problemas testes da literatura.

Palavras-chave: Restauração Inexata, não monótono, restrições de ortogonalidade, transformação de Cayley.

Abstract

In this work, we present and study the non monotone algorithm inexact restoration to solve minimization problems with orthogonality constraints, which combines the Inexact Restoration Method of Fischer and Friedlander [1] and the nonmonotone criteria of Zhang and Hager [2]. We develop the theoretical tools to characterize the subspace tangent of the feasible set, which allows us to describe the proposed algorithm. We show, under certain hypotheses, the good definition of the Algorithm as well as the global convergence to viable points of the problem. The inexact restoration method is an iterative method that consists of two phases: viability and optimality. In this work, the feasibility phase will be obtained in an exact way by using Cayley transformation. Therefore, the sequence of restored points belong to the viable set. In the optimality phase, the descent directions can be obtained in two ways: projected spectral gradient or minimization of a quadratic approximation for the Lagrangian, both on the tangent subspace. To solve this minimization we use the conjugate gradient method [3]. The computational implementation of the proposed algorithm is performed on MATLAB *software* and is compared with the Wen and Yin method [4] and the Conjugated Gradient method from ManOpt [5] library for different test problems in the literature.

Keywords: Inexact Restoration, non monotone, orthogonality constraints, Cayley Transform.

Lista de Símbolos

$\mathbb{R}^n$	Conjunto dos vetores com n coordenadas reais
$\mathbb{R}^{n\times p}$	Conjunto das matrizes com entradas reais de n linhas e p colunas
$\nabla f(X)[Z]$	Derivada direcional de f em X na direção Z
$\lim_{k\in\mathcal{K}} x_k$	O limite de x_k restrito a $k \in \mathcal{K}$
$\text{vec}(X)$	Dada uma matriz $X \in \mathbb{R}^{m\times n}$, denotamos $\text{vec}(X) = (x_1^T, \ldots, x_n^T)^T \in \mathbb{R}^{mn}$, em que $x_i \in \mathbb{R}^m$ é a i-ésima coluna de X
$\langle A, B\rangle$	Produto Interno no espaço das matrizes $n \times p$
$\|A\|_F$	Norma de Frobenius
I_n	Matriz identidade de ordem n
$\text{diag}(d_1, \ldots, d_p)$	Matriz diagonal de ordem $m \times n$, onde $d_{ij} = 0$ para todo $i \neq j$, $d_i = d_{ii}$ e $p = \min\{m, n\}$
$\text{Ker}(A)$	Núcleo do operador A
$\mathcal{L}$	Função de Lagrange

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução

Neste trabalho, consideramos o problema de otimização com restrições de ortogonalidade,

$$\begin{aligned} \text{minimizar} \quad & F(X) \\ \text{s.a.} \quad & X^T X = I, \\ & X \in \Omega, \end{aligned} \tag{1.1}$$

em que $\Omega \subset \mathbb{R}^{n\times p}$ $(p \leq n)$ é um conjunto convexo e compacto e $F : \mathbb{R}^{n\times p} \to \mathbb{R}$ é uma função continuamente diferenciável. Muitos problemas podem ser formulados como (1.1). Dentre eles, podemos citar o problema de autovalor de polinômios matriciais [6, 7], o problema de Procrustes Ortogonal (simples e generalizado) [8, 9, 10], o problema de minimização da Energia Total de Kohn-Sham [11], a diagonalização conjunta [12, 13], a análise de componentes principais [14, 15, 16], o problema de reduzir o posto de uma matriz de correlação [17, 18, 19], etc.

O conjunto de restrições $\mathrm{St}(n,p) := \{X \in \mathbb{R}^{n\times p} : X^T X = I\}$ é conhecido, na literatura, como "*Variedade de Stiefel*", em homenagem a Eduard Stiefel, que estudou sua topologia [20]. Note que no caso em que $p = 1$, a variedade é uma esfera unitária; se $p = n$, ela é conhecida como grupo ortogonal. A dimensão da variedade $\mathrm{St}(n,p)$ é $np - \frac{1}{2}p(p+1)$ e pode ser vista como uma subvariedade de $\mathbb{R}^{n\times p}$. Devido a que este tipo de variedade aparecer em diferentes problemas das ciências aplicadas, vários métodos númericos para o problema (1.1) têm sido propostos. Bolla *et al.* [21] analisaram o problema do máximo da soma de formas quadráticas com restrições de Stiefel baseados na teoria das matrizes. Além disso, eles obtiveram um algoritmo que garante

convergência global para pontos estacionários. Os autores Edelman, Arias e Smith [22] desenvolveram os métodos de Newton e de Gradientes Conjugados na variedade de Stiefel.

Vale ressaltar que, embora a variedade de Stiefel seja um conjunto compacto, o que garante a existência de um mínimo global, a determinação do mínimo não é tarefa fácil devido ao fato da variedade ser um conjunto não convexo. Com isso, o problema (1.1) apresenta muitos pontos estacionários, o que torna este problema de otimização difícil de resolver. Alguns casos muito particulares apresentam soluções analíticas, como por exemplo o problema de autovalores lineares e o problema de Procrustes balanceado. No entanto, na maioria dos casos, uma solução do problema (1.1) é encontrada por técnicas iterativas computacionais. Porém, preservar a viabilidade da sequência gerada pode ser computacionalmente caro. A maioria dos algoritmos viáveis propostos na literatura, ou utilizam rotinas para ortogonalizar matrizes, ou geram pontos ao longo de geodésicas. No primeiro caso, é necessária uma decomposição em valores singulares em cada iteração ou decomposição QR, e no segundo, calcula-se a exponencial de matrizes ou solução de equações diferenciais parciais. Uma classe promissora de métodos usa a transformação de Cayley para manter a viabilidade, e portanto, em vez de uma decomposição SVD, estes métodos resolvem um sistema linear; isto leva a iterações mais econômicas do ponto de vista numérico [4, 23, 24].

Alguns métodos de otimização trabalham com pontos não necessariamente viáveis, como por exemplo o Método Lagrangiano Aumentado [25], a Programação Quadrática Sequencial [26] e o Método Splitting [27]. Mais recentemente, o método de Lagrangiano Aproximado Sequencial foi introduzido em Zhu *et al.* [28], que resolve problemas com restrições lineares e restrições de ortogonalidade generalizadas. Uma outra alternativa aos métodos de otimização do tipo não viável é o Método de Restauração Inexata, introduzido por Martínez e Pillota [29]. Em resumo, este esquema decompõe cada iteração em duas fases: uma de otimalidade (minimização no subespaço tangente) e a outra de viabilidade. Visto como uma técnica de otimização em $\mathbb{R}^n$, dado um ponto $y \in \mathbb{R}^n$, o método de Restauração Inexata gera uma aproximação para o conjunto tangente das restrições em y e, utilizando algum critério de minimização, encontra um ponto x neste conjunto tangente que melhora o ponto y, segundo uma função de mérito específica. Assim, o método procede calculando um ponto y^+ que está mais próximo do conjunto viável, quando comparado com y, e continua como descrito anteriormente, calculando um x^+. Os primeiros métodos de Restau-

ração Inexata [29, 30] foram motivados pelas condições de otimalidade sequencial AGP (approximate gradient projection) [31]. A convergência local do método foi provada pelos autores Birgin e Martínez [32]. Um novo método de Restauração Inexata foi desenvolvido pelos autores Fischer e Friedlander [1], o qual apresenta propriedades de convergência global a pontos estacionários do problema. O principal resultado deste artigo estabelece que, com certas hipóteses, as direções de descida obtidas no subespaço tangente convergem para zero. Em [33], os autores Gomes-Ruggiero *et al*. escolhem direções de descida no subespaço tangente baseado no método Gradiente Projetado Espectral. O resultado de convergência para esta nova escolha da direção é fundamentado por [29]. Recentemente, no trabalho [34], foi apresentada uma variação do método de restauração Inexata que propõe uma modificação na função de mérito. A convergência global desta nova abordagem é provada com certas hipóteses de condição de qualificação fracas.

Neste trabalho, embora utilizemos a teoria do algoritmo de Restauração Inexata, consideramos o ponto y sempre viável (a fase da restauração é exata). Para evitar cálculo da SVD na fase de viabilidade, usamos a transformação de Cayley, que tem a propriedade de preservar ortogonalidade. Além disso, na fase de otimização, uma das escolhas da direção de busca pode ser feita minimizando o Lagrangiano no conjunto tangente das restrições. Para tanto, usamos uma adaptação do método do gradiente conjugado para minimização de quadráticas com restrições lineares [3]. A fim de diminuir o número de avaliações da função de mérito no Algoritmo de Restauração Inexata, em comparação com a busca monótona, incorporamos o esquema não monótono de Zhang e Hager [2].

A fim de analisar o desempenho numérico do método proposto, usamos uma implementação computacional do método no ambiente MATLAB2017b. Para o cálculo da direção de descida, foi implementado o método dos gradientes conjugados. Foram realizados testes numéricos para o problema de autovalor linear, procrustes ortogonal, minimização da energia total e minimização de quadráticas.

O trabalho está estruturado da seguinte forma. No capítulo 2, apresentamos e descrevemos cada passo do método de Restauração Inexata não monótono, assim como os resultados teóricos para demonstrar a convergência do método no capítulo seguinte. No capítulo 3, propomos o algoritmo de Restauração Inexata não monótono para o problema de minimização matricial com restrições de ortogonalidade e, além disso, mostramos resultados matemáticos que fundamentam nosso algoritmo, os quais garantem a convergência global a pontos estacionários. No

capítulo 4, apresentamos os resultados dos testes numéricos para os problemas acima descritos. No capítulo 5, são colocadas as conclusões do trabalho.

Capítulo 2

Método não monótono de restauração inexata

Consideremos uma descrição do método de Restauração Inexata (RI) aplicado ao problema com restrições de igualdade:

$$\begin{array}{ll} \text{minimizar} & F(x) \\ \text{s.a} & H(x) = 0 \\ & x \in \Omega, \end{array} \tag{2.1}$$

em que $F : \mathbb{R}^n \to \mathbb{R}, H : \mathbb{R}^n \to \mathbb{R}^m$ são funções continuamente diferenciáveis e $\Omega \subset \mathbb{R}^n$, um conjunto convexo e compacto. Em alguns casos, Ω pode ser o espaço $\mathbb{R}^n$. O ponto $x \in \Omega$ que satisfaz todas as restrições é chamado de ponto viável.

O método RI foi introduzido por Martínez e Pilotta [29], com o objetivo de trabalhar com sequências de iterações não viáveis, sobretudo quando as restrições são fortemente não lineares, para resolver problemas de otimização restrita.

O método RI é um processo iterativo para calcular soluções para o problema (2.1) e consta de duas fases: restauração e minimização. A Figura 2.1 ilustra a ideia por trás do método de Restauração Inexata. Os passos característicos da metodologia de Restauração Inexata são descritos a seguir:

1. Dado o iterado atual $x_k \in \Omega$ geramos o ponto y_k mais próximo do conjunto viável, o qual é calculado usando um procedimento arbitário; na prática, é dependente das características do problema. Este passo é chamado de Passo de Restauração ou Viabilidade.

2. Calculamos o ponto $z_k = y_k + d_k$, que pertence a uma aproximação linear das restrições no ponto y_k, de tal forma que o valor da função de mérito que mede a otimalidade melhore no ponto z_k com relação a y_k. Este passo é conhecido como minimização.

3. Se o ponto z_k satisfaz um critério que combina viabilidade e otimalidade (função de mérito), definimos $x_{k+1} = z_k$. Caso contrário, realizamos uma busca linear ao longo da direção a fim de reduzir o valor da função de mérito.

Na fase de otimalidade, busca-se, um ponto $z_k = y_k + d_k$ que possa melhorar o valor da função objetivo [29] ou da função Lagrangiano como no artigo [30]. O ponto z_k será aceito como um novo iterando x_{k+1} se o valor da função de mérito, que combina a viabilidade e otimalidade, no ponto z_k é suficientemente menor do que em x_k [29, 30] ou se satisfaz uma estratégia de filtro [35].

Este trabalho está baseado no método de restauração inexata proposto por Fischer e Friedlander [1], onde o novo iterado é calculado numa aproximação linear das restrições e é aceito por um processo de busca linear que envolve a função de mérito. Neste capítulo, propomos algumas modifições no Algoritmo de Fischer e Friedlander: realizamos uma busca não monótona, inspirados em Zhang e Hager [2] para a função de mérito. O propósito de usar busca não monótona é obter melhores resultados numéricos para resolver problemas com restrições de ortogonalidade.

O resultado principal do algoritmo proposto por Fischer e Friedlander garante a convêrgencia para zero da sequência de direções de descida. Além disso, escolhendo direções apropriadas, implica que a sequência gerada $\{x_k\}$ possui uma subsequência convergente a pontos viáveis que satisfazem a condição de otimalidade AGP, proposta por Martínez e Svaiter [31]. A vantagem deste método é a liberdade na escolha das estratégias para realizar as duas fases, o que permite a escolha de algoritmos apropriados para cada problema, conforme as características do problema a ser resolvido.

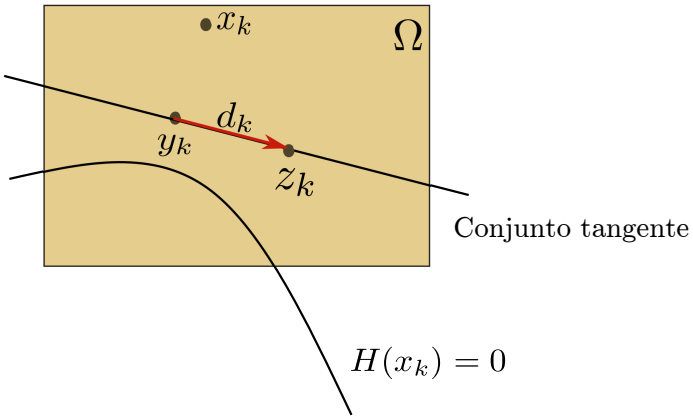

Figura 2.1: A ideia geral do método de Restauração Inexata.

A seguir, para todo $x \in \Omega$ e $\lambda \in \mathbb{R}^m$, definimos o Lagrangiano por:

$$\mathcal{L}(x, \lambda) := F(x) + \langle H(x), \lambda \rangle. \tag{2.2}$$

O ponto $(\bar{x}, \bar{\lambda}) \in \Omega \times \mathbb{R}^m$ é dito um ponto estacionario do problema (2.1) se

$$\mathrm{P}_\Omega(\bar{x} - \nabla \mathcal{L}(\bar{x}, \bar{\lambda})) - \bar{x} = 0, \tag{2.3}$$

$$H(\bar{x}) = 0, \tag{2.4}$$

onde $\mathrm{P}_\Omega : \mathbb{R}^n \to \Omega$ denota a projeção ortogonal em Ω. Além disso, se $(\bar{x}, \bar{\lambda})$ é um ponto estacionário de (2.1), dizemos que $\bar{x}$ é um ponto KKT.
Neste trabalho, consideramos uma função $h : \Omega \to [0, \infty)$ tal que

$$\|H(x)\| \leq h(x), \ \forall x \in \Omega, \tag{2.5}$$

que representa a medida da viabilidade de um ponto $x \in \Omega$.
Assim, definimos a função de mérito $\Phi : \Omega \times [0, \infty) \to \mathbb{R}$ por

$$\Phi(x, \theta) = \theta F(x) + (1 - \theta)h(x), \quad \forall (x, \theta) \in \Omega \times [0, 1], \tag{2.6}$$

que combina a relação entre a viabilidade e otimalidade. Esta relação será utilizada para decidir se o ponto intermediário z_k, que foi primeiramente definido em [1], será aceito como o novo iterado do algoritmo. A seguir, descreveremos os passos do Algoritmo de Restauração Inexata.

2.1 Fase de Restauração

Dado o ponto $x_k \in \Omega$, encontramos o ponto $y_k \in \Omega$, de modo que esteja mais próximo do conjunto viável do problema. Neste caso, o ponto y_k deve satisfazer duas condições:

$$h(y_k) \leq rh(x_k)$$
$$\|y_k - x_k\| \leq \beta h(x_k),$$

em que $r \in [0, 1)$ e $\beta > 0$. Se y_k é um ponto viável, então podemos escolher qualquer $r \in [0, 1)$ e, neste caso, dizemos que a "restauração é exata". A primeira condição acima estabelece a necessidade de obter, no passo de restauração, um ponto y_k que esteja mais próximo da viabilidade em comparação com o ponto x_k, conforme é ilustrado na Figura 2.2. A segunda condição acima estabelece que o passo de restauração não seja significativamente grande. Note que, se o ponto corrente x_k é viável, então y_k é igual a x_k.

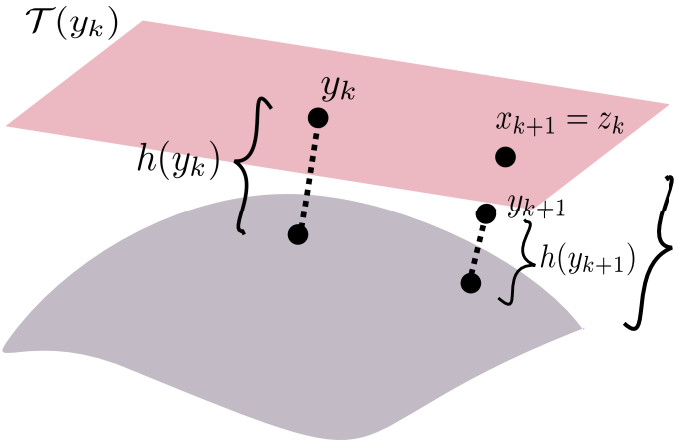

Figura 2.2: Passo de restauração.

2.2 Conjunto tangente e direção de descida

Seja $y_k \in \Omega$, definimos o subespaço

$$\mathcal{S}(y_k) := \text{Ker}(\nabla H(y_k)), \tag{2.7}$$

e $\mathcal{T}(y_k)$ o conjunto das direções tangentes em y_k

$$\mathcal{T}(y_k) := \{y_k + d \,|\, y_k + d \in \Omega \text{ e } d \in \mathcal{S}(y_k)\}. \tag{2.8}$$

Considere a direção de descida proposta por Fischer e Friedlander

$$\begin{aligned} d_k :&= \mathrm{P}_{\mathcal{T}(y_k)}(y_k - \nabla F(y_k)) - y_k \\ &= \mathrm{P}_{\mathcal{S}(y_k)}(-\nabla F(y_k)), \end{aligned} \tag{2.9}$$

em que $\mathrm{P}_{\mathcal{T}(y_k)}(z)$ é a projeção ortogonal de z em $\mathcal{T}(y_k)$, conforme é esquematizado na Figura 2.3.

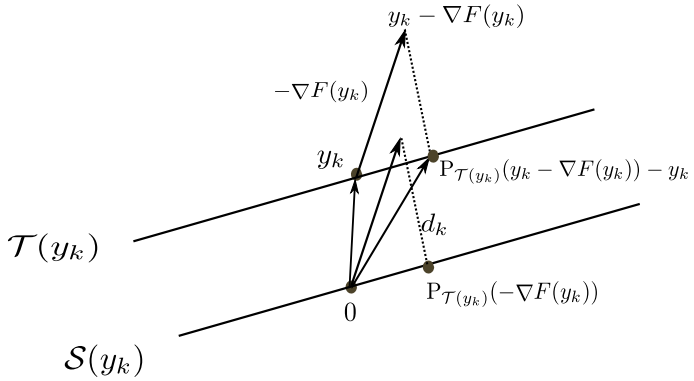

Figura 2.3: Direção de descida d_k.

Podemos verificar que d_k é uma direção de descida para F no subespaço $\mathcal{S}(y_k)$. De fato, pela definição da direção em (2.9), temos, pelo teorema de Pitágoras:

$$\begin{aligned} \|-\nabla F(y_k) - d_k\|^2 &\leq \|-\nabla F(y_k)\|^2, \\ \|\nabla F(y_k)\|^2 + 2\langle \nabla F(y_k), d_k\rangle + \|d_k\|^2 &\leq \|\nabla F(y_k)\|^2, \\ \langle \nabla F(y_k), d_k\rangle &\leq -\frac{1}{2}\|d_k\|^2, \end{aligned}$$

logo, d_k é uma direção de descida.
Note que, se $d_k := \mathrm{P}_{\mathcal{S}(y_k)}(-\nabla F(y_k)) = 0$, então y_k é um ponto estacionário do problema (2.1).

2.3 Passo de otimalidade

Nesta parte, levamos em consideração o problema de encontrar o valor do parâmetro t_k, que minimiza a função de mérito Φ, definida por (2.6), ao longo da reta que passa por x_k e tem direção d_k. De

modo geral, Fischer e Friedlander [1] exigem que a função de mérito no ponto $x_k + td_k$ seja menor que a função de mérito no ponto x_k. De fato, é necessária de uma redução suficiente da função de mérito, que é definida pela verificação do seguinte teste:

$$\Phi(x_k + td_k, \theta_{k+1}) - \Phi(x_k, \theta_{k+1}) \leq \frac{(1-r)}{2}(h(y_k) - h(x_k))$$
$$F(y_k + td_k) - F(y_k) \leq -\gamma t \|d_k\|^2, \quad \gamma \in (0,1)$$

em que a escolha de θ_k a cada iteração depende de considerações teóricas, por exemplo, se o tamaho de $h(x_k)$ é grande, então o pesso associado a $F(x_k)$ em (2.6) deve ser menor.
Em geral, é usado o *backtracking* para encontrar o valor do comprimento da direção de descida $t = t_k$, o qual é definido como o primeiro termo da sequência $\{1/2^j\}_{j\in\mathbb{N}}$ tal que satisfaz:

$$\Phi(x_k + t_k d_k, \theta_{k+1}) - \Phi(x_k, \theta_{k+1}) \leq \frac{(1-r)}{2}(h(y_k) - h(x_k))$$
$$F(y_k + t_k d_k) - F(y_k) \leq -\gamma t_k \|d_k\|^2.$$

Quando o critério acima é satisfeito, atualizamos $x_{k+1} = x_k + t_k d_k$. Caso contrário, fazemos uma nova busca linear na direção d_k, de tal forma que o novo ponto $x_k + t_k d_k$ verifica a condição acima.

2.4 Condição de otimalidade AGP

Definimos a direção $d(z)$ como a projeção ortogonal da direção $-\nabla F(z)$ sobre $\mathcal{T}(z)$, isto é,

$$d(z) = \mathrm{P}_{\mathcal{T}(z)}(z - \nabla F(z)) - z. \tag{2.10}$$

O ponto viável z tal que $d(z) = 0$ é chamado de ponto estacionário do Problema (2.1).

Dizemos que o ponto viável x satisfaz a condição AGP, se existe uma sequência $\{x_k\} \subset \mathbb{R}^n$ que converge para x tal que

$$\lim_{k\to\infty} d(x_k) = 0.$$

Esta condição de otimalidade foi introduzida por Martínez e Svaiter [31], a qual surgiu a partir do estudo do método de restauração inexata. A seguir, apresentamos o Algoritmo de Restauração Inexata.

Algoritmo 1: Restauração Inexata.

Dados: Seja $r \in [0,1)$, $\beta > 0, \gamma > 0, \bar{\gamma} > 0$.
Escolha uma aproximação inicial arbitrária $x_0 \in \Omega$ e $\theta_0 \in (0,1)$.
Defina $k = 0$.
Passo 1. Passo de Restauração
Calcule $y_k \in \Omega$ tal que:

$$h(y_k) \leq r h(x_k) \tag{2.11}$$
$$F(y_k) \leq F(x_k) + \beta h(x_k). \tag{2.12}$$

Passo 2. Escolha do parâmetro de Penalidade
Calcule θ_{k+1} como o primeiro θ da sequência $\{\theta_k/2^j\}_{j\in\mathbb{N}}$ tal que

$$\Phi(y_k, \theta_{k+1}) \leq \Phi(x_k, \theta_{k+1}) + \frac{1}{2}(h(y_k) - h(x_k)). \tag{2.13}$$

Passo 3. Escolha da direção de busca
Calcule $d_k \in \mathbb{R}^n$ tal que $y_k + d_k \in \Omega$,

$$F(y_k + t d_k) \leq F(y_k) - \gamma t \|d_k\|^2, \tag{2.14}$$
$$h(y_k + t d_k) \leq h(y_k) + \bar{\gamma} t^2 \|d_k\|^2, \tag{2.15}$$

$\forall t \in [0, \tau]$.
Passo 4. Passo de minimização
Calcule t_k sendo o primeiro t da sequência $\{1/2^j\}_{j\in\mathbb{N}}$ tal que

$$\Phi(y_k + t_k d_k, \theta_{k+1}) \leq \Phi(x_k, \theta_{k+1}) + \frac{(1-r)}{2}(h(y_k) - h(x_k)) \tag{2.16}$$
$$F(y_k + t_k d_k) \leq F(y_k) - \gamma t_k \|d_k\|^2. \tag{2.17}$$

Passo 5. Atualização
Defina $x_{k+1} := y_k + t_k d_k$ e $k := k + 1$ volte para o Passo 1.

2.5 Resultados de convergência global do método de restauração inexata

Nesta seção, apresentamos os resultados teóricos para o Algoritmo 1. Abordamos aspectos relacionados com a boa definição, viabilidade

e convergência do algoritmo. Algumas demonstrações serão omitidas e podem ser encontradas em [1].
Vamos supor as seguintes hipóteses:

1. Ω é um conjunto convexo e compacto.

2. As funções F e H são continuamente diferenciáveis.

3. O gradiente de F e H satisfazem a condição de Lipschitz. Isto é, existe $L > 0$ tal que, para todo $x, y \in \Omega$,

$$\|\nabla F(y) - \nabla F(x)\| \leq L\|y - x\| \tag{2.18}$$
$$\|\nabla H(y) - \nabla H(x)\| \leq L\|y - x\|. \tag{2.19}$$

Lema 2.1. *Existem constantes γ, $\bar{\gamma}$, $\tau > 0$ tais que*

$$F(y_k + td_k) \leq F(y_k) - \gamma t\|d_k\|^2,$$
$$\|H(y_k + td_k)\| \leq \|H(y_k)\| + \bar{\gamma}t^2\|d_k\|^2,$$

para todo $y_k \in \Omega$, $t \in [0, \tau]$ e sendo d_k definido em (2.9).

Demonstração. A demonstração encontra-se no artigo de Fischer e Friedlander [1] (Lema 1). ■

Teorema 2.2. *O passo x_{k+1} do Algoritmo 1 está bem definido. Além disso, existe $k_0 \in \mathbb{N}$ e $\bar{t} > 0$ tal que*

$$\theta_k = \theta_{k_0} > 0, \forall k \geq k_0$$
$$t_k \geq \bar{t}, \quad \forall k \in \mathbb{N}.$$

Demonstração. A demonstração encontra-se no artigo de Fischer e Friedlander [1] (Lema 3). ■

Teorema 2.3. *Para qualquer sequência $\{x_k\}_{k\in\mathbb{N}}$ gerada pelo Algoritmo 1, existe $\sigma > 0$ tal que*

$$\sum_{k=0}^{\infty} h(x_k) = \sigma.$$

Logo, $\lim_{k\to\infty} h(x_k) = 0$.

Demonstração. A demonstração encontra-se no artigo de Fischer e Friedlander [1] (Teorema 1). ■

Teorema 2.4. *A sequência $\{d_k\}$ gerada pelo Algoritmo 1 satisfaz*

$$\lim_{k\to\infty} d_k = 0.$$

Demonstração. A demonstração encontra-se no artigo de Fischer e Friedlander [1] (Teorema 2). ■

Teorema 2.5. *Sejam as sequências $\{x_k\}$ e $\{y_k\}$ geradas pelo Algoritmo 1 e $\mu > 0$, então temos as seguintes propriedades:*

1. *O limite $\lim_{k\to\infty} h(x_k) = \lim_{k\to\infty} h(y_k) = 0$ e qualquer ponto de acumulação das sequências $\{x_k\}$ e $\{y_k\}$ é viável.*

2. *Se $\|d_k\| \geq \mu \|P_{\mathcal{S}(y_k)}(-\nabla F(y_k))\|$, todo ponto limite de $\{x_k\}$ satisfaz a condição de otimalidade AGP [31].*

3. *Se um ponto limite x^* satisfaz a condição AGP e alguma condição de qualificação, por exemplo: Mangasarian-Fromovitz (MFCQ) ou dependência linear positiva constante (CPLD), então a condição KKT é satisfeita em x^*.*

Demonstração. 1. Demonstraremos este item. Segundo o resultado 2.3 e a condição 2.11 temos que

$$\lim_{k\to\infty} h(y_k) \leq r \lim_{k\to\infty} h(x_k) = 0.$$

Segue da desigualdade acima e da continuidade de h, que todo ponto de acumulação das sequências $\{x_k\}$ e $\{y_k\}$ é viável.

2. Seja x^* um ponto de acumulação de $\{x_k\}$, então pelo item (3) existe $\mathbb{N}' \underset{\infty}{\subset} \mathbb{N}$ tal que

$$\lim_{k\subset\mathbb{N}'} y_k = x^*.$$

Segue do Teorema (2.4) e pela hipótese deste item que

$$\lim_{k\to\infty} \mathrm{P}_{\mathcal{S}(y_k)}(-\nabla F(y_k)) = 0.$$

Além disso, pelo item (1) tem-se que x^* é viável. Portanto, x^* satisfaz a condição AGP.

3. A prova para o caso da condição de qualificação MFCQ encontra-se em Martinez e Svaiter [31](Corolário 2) e para o caso de CPLD a prova encontra-se em Gomes-Ruggiero *et al.* [33](Teorema 3.11).

■

O próximo teorema afirma que, sob certas hipóteses, o Algoritmo 1, gera sequências com taxas de convergência local linear ou quadrática.

Teorema 2.6. *Assuma que os passos 1 e 3 do Algoritmo são satisfeitos para todo* $k \in \mathbb{N}$. *Suponhamos também que existem* $c > 0, \zeta \in [0,1)$, $\lambda_k \in \mathbb{R}^m, \zeta_k \in [0,\zeta], \forall k \in \mathbb{N}$, *tais que*

$$\begin{aligned} &\|P_\Omega(y_k + d_k - \nabla\mathcal{L}(y_k + d_k, \lambda_{k+1})) - (y_k + d_k)\| \\ &\leq \zeta_k \|P_\Omega(y_k - \nabla\mathcal{L}(y_k, \lambda_k)) - y_k)\| \end{aligned} \tag{2.20}$$

e

$$\|d_k\| + \|\lambda_{k+1} - \lambda_k\| \leq c\|P_\Omega(y_k - \nabla\mathcal{L}(y_k, \lambda_k)) - y_k\|. \tag{2.21}$$

Também, admita que $(\bar{x}, \bar{\lambda})$ *é um ponto estacionário e que* $t_k = 1$ *para* k *suficientemente grande. Então, existem um* $\delta, \epsilon > 0$ *tais que:*

- *Se para algum* $k_0 \in \mathbb{N}, \|x_{k_0} - \bar{x}\| \leq \epsilon$ *e* $\|\lambda_{k_0} - \bar{\lambda}\| \leq \delta$, *então a sequência* $\{(x_k, \lambda_k)\}$ *converge para algum ponto estacionário.*
- *Se* $r = \zeta = 0$, *a convergência é quadrática.*

A prova deste teorema pode ser encontrada em [32] (Teorema 2.3 e 2.5).

Observações

1. Na condição (2.20), o ponto $y_k + d_k$ é calculado como a solução aproximada para o problema

$$\begin{array}{cc} \text{minimizar} & \mathcal{L}(y_k + d, \lambda_k) \\ \text{s.a} & d \in \mathcal{S}(y_k) \cap \Omega. \end{array} \tag{2.22}$$

2. A condição (2.21) é uma hipótese de estabilidade que garante que a solução na fase de otimalidade está limitada.

2.6 Restauração inexata não monótona

A monotonia da sequência formada pelos valores da função objetivo, apesar de parecer vantajosa, apresenta algumas dificuldades. Entre elas, mencionaremos dois elementos importantes:

1. O algoritmo perde sua eficiência principalmente na minimização de funções que apresentam uma bacia de atração estreita e curva, o que causa passos bem pequenos ou trajetória zig-zag [36, 37].

2. A busca linear de Armijo pode não ser válida para passos t pequenos, pois $f(x_k + td_k) \simeq f(x_k)$. Em tal situação, x_k pode estar longe do valor minimizado por f, não obstante, a condição de Armijo pode não ser verificada devido aos valores da função $f(x_k + td_k)$ e $f(x_k)$ serem quase iguais na aritmética de ponto flutuante,

$$0 \simeq f(x_k + td_k) - f(x_k) > \sigma t f(x_k)^T d_k,$$

[38].

Portanto, a monotonicidade na busca linear de Armijo pode tornar o Algoritmo muito lento, pois em alguns casos pode precisar de muitos passos de busca linear e, consequentemente, muitas avaliações da função objetivo. Desse modo, Grippo, Lampariello e Lucidi [36] introduziram uma técnica de busca linear não monótona, que consiste em uma variante da condição de Armijo e exige um decréscimo da função a cada $\overline{M}$ iterações. Seja

$$f_{l(k)} = \max_{0 \leq j \leq m(k)} \{f(x_{k-j})\}. \tag{2.23}$$

Dado um ponto x_k, uma direção de descida d_k e os parâmetros $\sigma \in (0, 1)$ e $\overline{M} \in \mathbb{N}$, a estratégia de Grippo, Lampariello e Lucidi consiste em encontrar um comprimento de passo t_k tal que

$$f(x_k + t_k d_k) \leq f_{l(k)} + \sigma t_k \nabla f(x_k)^T d_k,$$

em que $m(0) = 0$ e $0 \leq m(k) \leq \min\{m(k-1) + 1, \overline{M}\}$.

Posteriormente, o termo não monótono $f_{l(k)}$ foi utilizado por Grippo *et al* [36] em algoritmos sofisticados, . Embora o método não monótono proposto por Grippo *et al.* funcione bem em muitos casos, apresenta algumas desvantagens: primeiro, bons valores da função objetivo são descartados pelo máximo em (2.23). Segundo, em muitos casos, o desempenho numérico é dependente da escolha do parâmetro $\overline{M}$ [39, 40, 37]. Buscando superar as dificuldades, Zhang e Hager [2] propõem outra abordagem para superar a dependência do parâmetro $\overline{M}$ e substituir o termo $f_{l(k)}$ por um valor C_k que consiste da média ponderada dos valores da função calculados anteriormente.

Assim, dados os parâmetros $0 \leq \eta_{\min} \leq \eta_{\max} < 1$, $\sigma \in (0, 1)$, o iterado atual x_k, a direção de descida d_k, definindo $C_0 = f(x_0)$ e $Q_0 = 1$, o método de Zhang e Hager consiste em encontrar o comprimento de passo t_k que satisfaz a condição

$$f(x_k + t_k d_k) \leq C_k + \sigma t_k \nabla f(x_k)^T d_k,$$

em que

$$C_{k+1} = (\eta_k Q_k C_k + f(x_{k+1}))/Q_{k+1}, \quad \text{e} \quad Q_{k+1} = \eta_k Q_k + 1,$$

com $\eta_k \in [\eta_{\min}, \eta_{\max}]$.

Note que C_{k+1} é uma combinação convexa de C_k e $f(x_{k+1})$. Como $C_0 = f(x_0)$, e pela definição de C_k, segue que C_k é uma combinação convexa dos valores da função $f(x_0)$, $f(x_1)$, $f(x_2), \ldots, f(x_k)$. Também repare que a escolha de η_k controla o grau de não monotonicidade. Por exemplo, se $\eta_k = 0$ para todo $k \in \mathbb{N}$, então a busca linear se reduz a uma busca monótona de Armijo. Por outro lado, se $\eta_k = 1$, para todo $k \in \mathbb{N}$, então o termo $C_k = A_k$ em que,

$$A_k = \frac{1}{k+1} \sum_{i=0}^{k} f(x_i)$$

é a média dos valores da função para as iterações 0 até $k+1$. Assim, quando η_k aproxima-se de 0, a busca aproxima-se de uma busca linear monótona; e quando η_k aproxima-se de 1, o esquema torna-se cada vez mais não monótono.

Recentemente, Mo *et al.* [41] introduziram outro termo baseado na combinação convexa dos valores da função objetivo

$$D_k = f(x_k) + \eta_{k-1}(D_{k-1} - f(x_k)),$$

em que $\eta_{k-1} \in [\eta_{\min}, \eta_{\max}]$ e $D_0 = f(x_0)$. Em um trabalho mais recente, Amini *et al.* [42] propõem um novo termo não monótono como sendo a combinação convexa de $f_{l(k)}$ e $f(x_k)$.

2.6.1 Algoritmo

Nesta seção, propomos um algoritmo de restauração inexata não monótono para resolver o problema matricial com restrições de ortogonalidade. A ideia principal é estabelecer um termo não monótono no passo de minimização determinado pelo média dos valores da função de mérito anteriores já calculados. Para tanto, aplicamos a estratégia de não monotonia inspirada em Zhang e Hager [2]. Dada uma aproximação inicial $x_0 \in \Omega$, $\theta_0 \in (0,1)$ e definindo $C_0 = \Phi(x_0, \theta_0)$, $Q_0 = 1$, o critério de não monotonia para a função de mérito consiste em encontrar o valor de t_k que satisfaz

$$\Phi(y_k + t_k d_k, \theta_{k+1}) \leq T_k + \frac{(1-r)}{2}(h(y_k) - h(x_k)),$$

em que

$$T_k := \max\{C_k, \Phi(x_k, \theta_{k+1})\},$$

e

$$\begin{aligned} Q_{k+1} &= \eta_k Q_k + 1, \\ C_{k+1} &= (\eta_k Q_k C_k + \Phi(x_{k+1}, \theta_{k+1}))/Q_{k+1}, \end{aligned}$$

com $\eta_k \in [\eta_{\min}, \eta_{\max}]$, de modo que $T_k \geq \Phi(x_k, \theta_{k+1})$.
Aqui, as atualizações de C_k e η_k são feitas no passo seguinte ao *backtracking*.
A seguir, no Algoritmo 2 descrevemos os passos bem como os parâmetros necessários para a implementação numérica.

Proposição 2.7. *Se ∇F é uma função contínua e d_k satisfaz a condição para $\mu > 0$,*

$$\langle \nabla F(y_k), d_k \rangle \leq -\mu \|d_k\|^2, \tag{2.31}$$

então d_k é uma direção de descida para F em y_k, e existe $\gamma > 0$ tal que

$$F(y_k + td_k) - F(y_k) \leq -\gamma t \|d_k\|^2.$$

Demonstração. Pelo Teorema fundamental do cálculo, pela hipótese ∇F contínua e pela hipótese (2.31), temos que

$$\begin{aligned} F(y_k + td_k) - F(y_k) &= t\langle F(y_k), d_k \rangle \\ &\quad + \int_0^1 (\nabla F(y_k + \xi t d_k) - \nabla F(y_k)) t d_k d\xi \\ &\leq -t\mu \|d_k\|^2 + \frac{t^2}{2} L \|d_k\|^2 \\ &= t(\mu - \frac{tL}{2}) \|d_k\|^2. \end{aligned}$$

Assim,

$$F(y_k + td_k) - F(y_k) \leq -\gamma t \|d_k\|^2, \quad \gamma = \mu/2 \tag{2.32}$$

para todo $t \leq \tau := \min\{1, \frac{\mu}{2L}\}$. ∎

2.7 Resultados teóricos

Nesta seção, fornecemos resultados importantes que serão utilizados no capítulo posterior para demonstrar a convergência global do Algoritmo aplicado ao problema matricial.
Assumiremos o seguinte:

Algoritmo 2: Restauração Inexata não monótona.

Dados: Seja $r \in [0,1)$ e β, γ, $\bar{\gamma}$, e $0 \leq \eta_{\min} \leq \eta_{\max} < 1$.
Escolha $x_0 \in \Omega$ e $\theta_0 \in (0,1)$. Defina $C_0 = \Phi(x_0, \theta_0)$, $Q_0 = 1$ e $k = 0$.

Passo 1. Passo de Restauração
Calcule $y_k \in \Omega$ tal que:

$$h(y_k) \leq r h(x_k) \tag{2.24}$$
$$F(y_k) \leq F(x_k) + \beta h(x_k). \tag{2.25}$$

Passo 2. Escolha do parâmetro de Penalidade
Calcule θ_{k+1} como o primeiro θ da sequência $\{\theta_k/2^j\}_{j\in\mathbb{N}}$ tal que

$$\Phi(y_k, \theta_{k+1}) - \Phi(x_k, \theta_{k+1}) \leq \frac{1}{2}(1-r)(h(y_k) - h(x_k)). \tag{2.26}$$

Passo 3. Escolha da direção de busca
Calcule $d_k \in \mathbb{R}^n$ tal que $y_k + d_k \in \Omega$,

$$F(y_k + t d_k) \leq F(y_k) - \gamma t \|d_k\|^2, \tag{2.27}$$
$$h(y_k + t d_k) \leq h(y_k) + \bar{\gamma} t \|d_k\|^2, \ \forall t \in [0, \tau].$$

Passo 4. Passo de otimalidade
Calcule t_k sendo o primeiro t da sequência $\{1/2^j\}_{j\in\mathbb{N}}$ tal que

$$\Phi(y_k + t_k d_k, \theta_{k+1}) - T_k \leq \frac{(1-r)}{2}(h(y_k) - h(x_k)), \tag{2.28}$$

em que $T_k := \max\{C_k, \Phi(x_k, \theta_{k+1})\}$.

Passo 5. Atualização
Defina $x_{k+1} := y_k + t_k d_k$.
Escolha $\eta_k \in [\eta_{\min}, \eta_{\max}]$ e defina

$$Q_{k+1} = \eta_k Q_k + 1, \tag{2.29}$$
$$C_{k+1} = (\eta_k Q_k C_k + \Phi(x_{k+1}, \theta_{k+1}))/Q_{k+1}, \tag{2.30}$$

$k := k + 1$ e volte ao Passo 1.

(**H1**) Os gradientes de F e H satisfazem as condições de Lipschitz em Ω, isto é, existe $L > 0$ tal que

$$\|\nabla F(x) - \nabla F(y)\|_F \leq L\|x - y\|_F$$
$$\|\nabla H(x) - \nabla H(y)\|_F \leq L\|x - y\|_F,$$

para todo $x, y \in \Omega$.

No próximo resultado, provamos que os parâmetros de penalidade θ_k, assim como t_k, são limitados acima de zero.

Teorema 2.8. *O iterando x_{k+1} do Algoritmo 2 está bem definido. Além disso, existe $k_0 \in \mathbb{N}$ e $\bar{t} > 0$ tal que*

$$\begin{aligned} \theta_k &= \theta_{k_0} > 0, \quad \forall k \geq k_0, \\ t_k &\geq \bar{t}, \qquad\qquad k \in \mathbb{N}. \end{aligned}$$

Demonstração. Seja x_k gerado pelo Algoritmo 2. Note que por (2.24) e (2.25) temos que

$$\begin{aligned} \Phi(y_k, \theta) - \Phi(x_k, \theta) &= \theta(F(y_k) - F(x_k)) + (1-\theta)(h(y_k) - h(x_k)) \\ &\leq \theta\beta h(x_k) - (1-\theta)(1-r)h(x_k) \\ &= h(x_k)\big(\theta(\beta + 1 - r) - (1-r)\big). \end{aligned}$$

Portanto, se $0 \leq \theta \leq \tilde{\theta} := \frac{1-r}{2(\beta+1-r)}$ então

$$\Phi(y_k, \theta) - \Phi(x_k, \theta) \leq -\frac{1}{2}(1-r)h(x_k) \leq \frac{1}{2}(1-r)(h(y_k) - h(x_k)).$$

Assim, θ_{k+1} pode ser escolhido como sendo

$$\theta_{k+1} \geq \bar{\theta} := \min\{\theta_0, \tilde{\theta}/2\}, \ \ \forall k \in \mathbb{N}.$$

Como θ_{k+1} é o maior valor na sequência $\{\theta_k/2^j\}_{j\in\mathbb{N}}$, então existe $k_0 \in \mathbb{N}$ tal que

$$\theta_k = \theta_{k_0} \geq \bar{\theta}, \ \ \forall k \geq k_0. \tag{2.33}$$

Veremos que o Passo 4 do Algoritmo 2 está bem definido. Note que pela definição $T_k \geq \Phi(x_k, \theta_{k+1})$,

$$\Phi(y_k + td_k, \theta_{k+1}) - T_k \leq \Phi(y_k + td_k, \theta_{k+1}) - \Phi(x_k, \theta_{k+1})$$

Logo, usando (2.26), (2.27) e (2.33) e para $k \geq k_0$

$$\begin{aligned}
&\Phi(y_k + td_k, \theta_{k+1}) - \Phi(x_k, \theta_{k+1}) \\
&\quad = \big(\Phi(y_k + td_k, \theta_{k+1}) - \Phi(y_k, \theta_{k+1})\big) + \big(\Phi(y_k, \theta_{k+1}) - \Phi(x_k, \theta_{k+1})\big) \\
&\quad \leq \theta_{k+1}(F(y_k + t_k d_k) - F(y_k)) + (1 - \theta_{k+1})\big(h(y_k + td_k) - h(y_k)\big) \\
&\qquad + \frac{1}{2}(1 - r)(h(y_k) - h(x_k)) \\
&\quad \leq -\bar{\theta}\gamma t\|d_k\|^2 + \bar{\gamma}t^2\|d_k\|^2 + \frac{1}{2}(1 - r)(h(y_k) - h(x_k)) \\
&\quad = t\|d_k\|^2(-\bar{\theta}\gamma + \bar{\gamma}t) + \frac{1}{2}(1 - r)(h(y_k) - h(x_k)),\ \forall t \in [0, \tau].
\end{aligned}$$

Então, se $0 \leq t \leq \tilde{t} := \min\{\tau, \bar{\theta}\gamma/\bar{\gamma}\}$, temos que $-\bar{\theta}\gamma + \bar{\gamma}t \leq 0$ e

$$\Phi(y_k + td_k, \theta_{k+1}) - \Phi(x_k, \theta_{k+1}) \leq \frac{1}{2}(1 - r)(h(y_k) - h(x_k)).$$

Portanto, o Passo 4 do Algoritmo está bem definido e gera a sequência t_k com

$$t_k \geq \bar{t} := \frac{\tilde{t}}{2},\ \forall k \in \mathbb{N}.$$

■

Lema 2.9. *Seja $\{Q_k\}$ a sequência gerada pelo Algoritmo 2, então*

$$Q_{k+1} \leq \sum_{j=0}^{k} \eta_{\max}^j, \quad \textit{para todo} \quad k \geq 0.$$

Além disso,

$$Q_{k+1} \leq \frac{1}{1 - \eta_{\max}}, \quad \textit{para todo} \quad k \geq 0.$$

Demonstração. Para $k = 0$ e usando o fato de que $Q_0 = 1$ e $Q_{k+1} = \eta_k Q_k + 1$ e $\eta_k \in [\eta_{\min}, \eta_{\max}]$

$$\begin{aligned}
Q_1 &= \eta_0 Q_0 + 1 \\
&\leq \eta_{\max} + 1.
\end{aligned}$$

Usando o principio de indução matemática, suponha válido para k

$$Q_k \leq \sum_{j=0}^{k-1} \eta_{\max}^j.$$

Logo, na iteração $k+1$, verifica-se

$$\begin{aligned} Q_{k+1} &= \eta_k Q_k + 1 \\ &\leq \eta_{\max}(\sum_{j=0}^{k-1} \eta_{\max}^j) + 1 \\ &= \sum_{j=0}^{k} \eta_{\max}^j \end{aligned}$$

Por outro lado, desde que $|\eta_{\max}| < 1$, segue o resultado

$$Q_{k+1} \leq \sum_{j=0}^{k} \eta_{\max}^j \leq \sum_{j=0}^{\infty} \eta_{\max}^j = \frac{1}{1-\eta_{\max}}, \quad \forall k \geq 0.$$

■

Lema 2.10. *A sequência $\{T_k\}_{k\geq k_0+1}$ é monótona não crescente e*

$$T_{k+1} - T_k \leq -(1-\eta_{\max})\frac{(1-r)^2}{2}h(x_k), \quad \forall k \geq k_0. \tag{2.34}$$

Demonstração. Pelo Teorema 2.8, temos que $\theta_k = \theta_{k_0}$, $\forall k \geq k_0$. Note que

$$T_{k+1} := \max\{C_{k+1}, \Phi(x_{k+1}, \theta_{k_0})\}, \quad \forall k \geq k_0.$$

Por (2.28) e (2.24), temos que

$$\begin{aligned} C_{k+1} &= \frac{\eta_k Q_k C_k + \Phi(x_{k+1}, \theta_{k_0})}{Q_{k+1}} \\ &\leq \frac{\eta_k Q_k C_k + T_k + ((1-r)/2)(h(y_k) - h(x_k))}{Q_{k+1}} \\ &\leq \frac{(\eta_k Q_k + 1)T_k - ((1-r)^2/2)h(x_k)}{Q_{k+1}} \\ &= T_k - \frac{(1-r)^2}{2Q_{k+1}}h(x_k), \quad k \geq k_0. \end{aligned}$$

E como $Q_{k+1} \leq \frac{1}{1-\eta_{\max}}$, então

$$C_{k+1} \leq T_k - (1-\eta_{\max})\frac{(1-r)^2}{2}h(x_k), \quad k \geq k_0. \tag{2.35}$$

Por outro lado, para todo $k \geq k_0$ pelo Passo 4 do Algoritmo 2

$$\begin{aligned}\Phi(x_{k+1}, k_0) &\leq T_k - \frac{(1-r)^2}{2} h(x_k) \\ &\leq T_k - (1-\eta_{\max})\frac{(1-r)^2}{2} h(x_k), \quad k \geq k_0. \end{aligned} \tag{2.36}$$

Pela definição de T_{k+1}, pelos resultados (2.35) e (2.36), obtemos

$$T_{k+1} \leq T_k - (1-\eta_{\max})\frac{(1-r)^2}{2} h(x_k), \quad \forall k \geq k_0.$$

Agora, como $\eta_{\max} \in [0, 1)$, então verifica-se a desigualdade

$$T_{k+1} \leq T_k, \quad \forall k \geq k_0.$$

■

O Teorema seguinte garante que todo ponto de acumulação das sequências $\{x_k\}$ e $\{y_k\}$ são pontos viáveis.

Teorema 2.11. *Seja $\{x_k\}_{k\in\mathbb{N}}$ a sequência gerada pelo Algoritmo 2. Então,*

$$\lim_{k\to\infty} h(x_k) = \lim_{k\to\infty} h(y_k) = 0.$$

Além disso, todo ponto limite $\bar{x}$ de $\{x_k\}$ satisfaz

$$h(\bar{x}) = 0.$$

Demonstração. Pelo Teorema 2.8, para todo $k \geq k_0$, o parâmetro $\theta_k = \theta_{k_0}$. Assim, para $l > k_0$ e por (2.34) segue que

$$\begin{aligned} T_{l+1} - T_{k_0} &= \sum_{k=k_0}^{l} (T_{k+1} - T_k) \\ &\leq -(1-\eta_{\max})\frac{(1-r)^2}{2} \sum_{k=k_0}^{l} h(x_k), \quad \forall l \geq k_0. \end{aligned}$$

Logo,

$$(1-\eta_{\max})\frac{(1-r)^2}{2} \sum_{k=k_0}^{l} h(x_k) \leq T_{k_0} - T_{l+1} \leq T_{k_0} - \Phi(x_{l+2}, \theta_{k_0}) \tag{2.37}$$

e desde que $\Phi(\cdot, \theta_{k_0})$ é continua em Ω e $\{x_k\} \subset \Omega$, sendo Ω um conjunto compacto, então a sequência $\{\tilde{\sigma}_l\}_{l \geq k_0}$ é limitada, em que

$$\tilde{\sigma}_l := \sum_{k=k_0}^{l} h(x_k), \, \forall l \geq k_0.$$

Note que, a sequência $\{\tilde{\sigma}_l\}_{l \in \mathbb{N}}$ é monótona crescente e limitada. Logo, a série $\sum_{k=0}^{\infty} h(x_k)$ é convergente, e consequentemente $\lim_{k \to \infty} h(x_k) = 0$. Além disso tomando o limite na desigualdade (2.24), temos que

$$\lim_{k \to \infty} h(y_k) = 0.$$

■

Em resumo, o teorema acima garante a existência de uma subsequência de $\{x_k\}$ que converge para um ponto viável x^*. Além disso, se o ponto x^* satisfaz a condição de otimalidade AGP e a condição de qualificação CRCQ (posto constante), então x^* é um ponto KKT.
No capítulo seguinte provaremos que todo problema com restrições de ortogonalidade satisfaz CRCQ. Ainda, utilizando hipóteses adicionais, demonstraremos que todo ponto limite é AGP e, portanto, a convergência global do algoritmo proposto para minimização com restrições de ortogonalidade está assegurada.

Capítulo 3

Método de restauração inexata não monótono no problema de minimização em variedades de Stiefel

Nesta seção, aplicaremos o método de restauração inexata para resolver problemas de minimização com restrições de ortogonalidade

$$\begin{aligned} \text{minimizar} \quad & F(X) \\ \text{s.a.} \quad & X^T X = I, \\ & X \in \Omega, \end{aligned} \tag{3.1}$$

em que $F : \mathbb{R}^{n\times p} \to \mathbb{R}$ é uma função continuamente diferenciável e $\Omega \subset \mathbb{R}^{n\times p}$, um conjunto convexo e compacto. Denotaremos o conjunto viável do problema por $\mathcal{V} := \{X \in \Omega : X^T X - I = 0\}$. O conjunto $\mathrm{St}(n,p) := \{X \in \mathbb{R}^{n\times p} : X^T X - I = 0\}$ é conhecido como *variedade de Stiefel* [20]. Para o caso $p = 1$, a variedade de Stiefel identifica-se como a esfera unitária e para o caso $p = n$ identifica-se com o grupo ortogonal $O(n)$. A dimensão da variedade de Stiefel $\mathrm{St}(n,p)$ é $np - \frac{p(p+1)}{2}$ [43] e pode ser vista como uma subvariedade imersa em $\mathbb{R}^{n\times p}$.

O Problema (3.1) tem diversas aplicações, dentre as quais citamos o problema do autovalor linear [6, 7, 23], o problema de Procrustes [44],

o problema de diagonalização conjunta [13], o problema de alocação quadrática, o problema de minimização da energia total de Kohn-Sham [45] e a decomposição em valores singulares [46, 47]. Em geral, é difícil encontrar solução global para o Problema (3.1), pois as restrições são não convexas. Além disso, manter a ortogonalidade a cada iteração, para p grande, é numericamente caro.

A maioria dos métodos existentes para resolver este tipo de problema usa fatoração de matrizes, ou decomposição em valores singulares, ou QR. Outros requerem o cálculo de geodésicas, o que resulta em algoritmos computacionalmente caros. Recentemente, Wen e Yin [4] introduziram um método em que constroem curvas a partir da transformação de Cayley o que pode ser visto como uma forma especial do método de Crank-Nicolson. A cada busca linear, é necessário resolver apenas sistemas lineares de tamanho $2p \times 2p$.

Inspirados em Wen e Yin [4], aplicamos o método de Restauração Inexata não monótono da Seção 2.7.1 ao Problema (3.1), que inclui um procedimento específico na fase de restauração. Neste caso, aproveitamos as propriedades da transformação de Cayley para obter um método eficiente que preserva as restrições de ortogonalidade. Para tanto, defina a função

$$H(X) := X^T X - I, \, \forall X \in \Omega \tag{3.2}$$

e a função $h : \Omega \to [0, \infty)$ tal que

$$\|H(X)\| \leq h(X), \, \forall X \in \Omega. \tag{3.3}$$

Assim, temos a função de mérito

$$\Phi(X, \theta) = \theta F(X) + (1 - \theta)h(X), \; \forall (X, \theta) \in \Omega \times [0, 1].$$

Além disso, defina a função de Lagrange associada ao Problema (3.1):

$$\mathcal{L}(X, \Lambda) = F(X) + \frac{1}{2}\mathrm{tr}(\Lambda(X^T X - I)), \tag{3.4}$$

em que $\Lambda \in \mathbb{R}^{p \times p}$ é a matriz com os multiplicadores de Lagrange.

3.1 Subespaço tangente ao conjunto viável

Nesta seção, verificaremos resultados importantes referentes ao subespaço tangente a $\mathcal{V}$, e iremos obter o operador projeção neste subespaço.

Dado $Y \in \mathcal{V}$ definimos o subespaço

$$\mathcal{S}(Y) = \mathrm{Ker}(\nabla H(Y)), \tag{3.5}$$

e o conjunto tangente

$$\mathcal{T}(Y) = \{X \in \Omega : X - Y \in \mathcal{S}(Y)\}. \tag{3.6}$$

Note que $\mathcal{T}(Y)$ é um conjunto afim paralelo a $\mathcal{S}(Y)$ e restrito a Ω, isto é,

$$\mathcal{T}(Y) = (Y + \mathcal{S}(Y)) \cap \Omega. \tag{3.7}$$

A seguir, apresentamos caracterizações do subespaço $\mathcal{S}(Y)$. Com efeito, calculando a variação de primeira ordem de (3.2) obtemos o seguinte resultado.

Lema 3.1. *Seja $Y \in \mathcal{V}$. Então,*

$$\begin{aligned}\mathcal{S}(Y) =& \left\{ Z \in \mathbb{R}^{n\times p} : Y^T Z + Z^T Y = 0 \right\} \\ =& \left\{ YW + Y^{\perp} K : W^T = -W \in \mathbb{R}^{p\times p},\, K \in \mathbb{R}^{(n-p)\times p} \right\},\end{aligned}$$

em que $Y^{\perp} \in \mathbb{R}^{n\times n-p}$ é tal que $YY^T + Y^{\perp}(Y^{\perp})^T = I$. Além disso, a dimensão de $\mathcal{S}(Y)$ é $np - p(p+1)/2$.

A prova deste lema pode ser encontrada em [22] (Seção 2.2.1).

No seguinte resultado, encontramos uma forma de representar os elementos de $\mathcal{S}(Y)$. O resultado diz que $Z = AY$ pertence a $\mathcal{S}(Y)$ se e somente se A é antissimétrica.

Lema 3.2. *Seja $Y \in \mathcal{V}$. Definimos*

$$\mathcal{M}(Y) := \left\{ AY \in \mathbb{R}^{n\times p} : A \in \mathbb{R}^{n\times n}, \quad A^T = -A \right\}, \tag{3.8}$$

então $\mathcal{S}(Y) = \mathcal{M}(Y)$.

Demonstração. Seja $AY \in \mathcal{M}(Y)$, então

$$Y^T(AY) + (AY)^T Y = Y^T(A + A^T)Y = 0,$$

isto é, $AY \in \mathcal{S}(Y)$. Portanto,

$$\mathcal{M}(Y) \subset \mathcal{S}(Y). \tag{3.9}$$

Demonstraremos que $\dim \mathcal{M}(Y) = \dim \mathcal{S}(Y)$, para todo $Y \in \mathcal{V}$.
Denotamos por $\bar{I} = \begin{bmatrix} I \\ 0 \end{bmatrix} \in \mathcal{V}$, em que I é a matriz identidade $p \times p$, então

$$\mathcal{M}(\bar{I}) = \left\{ \bar{A} = \begin{bmatrix} A_{11} \\ A_{12} \end{bmatrix},\ A_{11}^T = -A_{11} \in \mathbb{R}^{p\times p},\ A_{12} \in \mathbb{R}^{(n-p)\times p} \right\}.$$

Logo,

$$\dim \mathcal{M}(\bar{I}) = (n-p)p + \frac{p(p-1)}{2}.$$

Dado $Y \in \mathcal{V}$, então esse elemento pode ser expressado como $Y = Q\bar{I}$, em que $Q = \begin{bmatrix} Y & Y^{\perp} \end{bmatrix} \in \mathbb{R}^{n\times n}$.
Verificaremos a seguinte igualdade de conjuntos

$$\mathcal{M}(\bar{I}) = \mathcal{M}(Q^T Y) = Q^T \mathcal{M}(Y). \tag{3.10}$$

De fato, mostraremos as seguintes inclusões

$$\mathcal{M}(\bar{I}) \subset Q^T \mathcal{M}(Y) \quad \text{e} \quad Q^T \mathcal{M}(Y) \subset \mathcal{M}(\bar{I})$$

1. Seja $Z \in \mathcal{M}(\bar{I})$, então Z é da forma $Z = \bar{A}\bar{I}$. Pela igualdade $\bar{I} = Q^T Y$ temos que $Z = \bar{A}Q^T Y$. Agora, multiplicando pela matriz Q, obtemos $QZ = Q\bar{A}Q^T Y$. Seja $A = Q\bar{A}Q^T$, então A é anti-simétrica e $Z = Q^T AY$,
   portanto, $Z \in Q^T \mathcal{M}(Y)$.

2. Seja $Z \in Q^T \mathcal{M}(Y)$, então $QZ \in \mathcal{M}(Y)$, isto é, $QZ = AY$ e A anti-simétrica. Logo, multiplicando por Q^T e usando o fato que Q é ortogonal, obtemos $Z = Q^T AQ\bar{I}$. Segue que $Z \in \mathcal{M}(\bar{I})$, pois $Q^T AQ$ é antissimétrica.

Por (3.10), segue que

$$\begin{aligned} \dim \mathcal{M}(Y) &= \dim Q^T \mathcal{M}(Y) \\ &= \dim \mathcal{M}(\bar{I}) \\ &= (n-p)p + \frac{p(p-1)}{2} \\ &= np - \frac{p(p+1)}{2}. \end{aligned}$$

Pelo Lema 3.1 temos que $\dim \mathcal{S}(Y) = np - \frac{p(p+1)}{2}$. Assim,

$$\dim \mathcal{M}(Y) = \dim \mathcal{S}(Y) \tag{3.11}$$

Portanto, de (3.9) e (3.11), temos que,

$$\mathcal{S}(Y) = \mathcal{M}(Y).$$

■

Observação De acordo com o Lema 3.2, o problema

$$\begin{aligned} &\text{minimizar } \|Z - X\|_F \\ &\text{s.a. } X \in \mathcal{S}(Y) \end{aligned} \tag{3.12}$$

pode ser substituído pelo problema equivalente

$$\begin{aligned} &\text{minimizar } \|Z - AY\|_F \\ &\text{s.a. } A^T + A = 0. \end{aligned} \tag{3.13}$$

No seguinte teorema, obtemos a fórmula fechada para a projeção de qualquer matriz $Z \in \mathbb{R}^{n\times p}$ no subespaço $\mathcal{S}(Y)$.

Teorema 3.3. *Sejam $Y \in \mathcal{V}$ e $Z \in \mathbb{R}^{n\times p}$, então a solução do problema*

$$\begin{aligned} &\textit{minimizar } \|Z - AY\|_F^2 \\ &\textit{s.a. } A^T + A = 0. \end{aligned} \tag{3.14}$$

é dada por

$$A^* = (I - \frac{1}{2}YY^T)ZY^T - YZ^T(I - \frac{1}{2}YY^T)$$

ou

$$A^* = UV^T, \tag{3.15}$$

em que

$$U = \begin{bmatrix}(I - \frac{1}{2}YY^T)Z & Y\end{bmatrix} \text{ e } V = \begin{bmatrix}Y & -(I - \frac{1}{2}YY^T)Z\end{bmatrix}. \tag{3.16}$$

Além disso, a projeção de Z em $\mathcal{S}(Y)$ é

$$P_{\mathcal{S}(Y)}Z = A^*Y = Z - \frac{1}{2}Y(Z^TY + Y^TZ). \tag{3.17}$$

Demonstração. A matriz $Y \in \mathcal{V}$, pode ser escrito da forma

$$Y = \begin{bmatrix} Y & Y^{\perp} \end{bmatrix} \begin{bmatrix} I \\ 0 \end{bmatrix} = Q\bar{I}.$$

Então temos que

$$\begin{aligned} \|A^*Y - Z\|_F^2 &= \|A^*Q\bar{I} - Z\|_F^2 \\ &= \|Q^TA^*Q\bar{I} - Q^TZ\|_F^2 \\ &= \|\bar{A}\bar{I} - Q^TZ\|_F^2, \end{aligned}$$

onde $\bar{A} = Q^TA^*Q$. Denote

$$\bar{A} = \begin{bmatrix} \bar{A}_{11} & -\bar{A}_{12}^T \\ \bar{A}_{12} & \bar{A}_{22} \end{bmatrix},\ \bar{A}_{11}^T = -\bar{A}_{11}.$$

A norma

$$\begin{aligned} \|\bar{A}\bar{I} - Q^TZ\|_F^2 &= \| \begin{bmatrix} \bar{A}_{11} & -\bar{A}_{12}^T \\ \bar{A}_{12} & \bar{A}_{22} \end{bmatrix} \begin{bmatrix} I \\ 0 \end{bmatrix} - \begin{bmatrix} Y^T \\ (Y^{\perp})^T \end{bmatrix} Z\|_F^2 \\ &= \| \begin{bmatrix} \bar{A}_{11} - Y^TZ \\ \bar{A}_{12} - (Y^{\perp})^TZ \end{bmatrix} \|_F^2 \\ &= \|\bar{A}_{11} - Y^TZ\|_F^2 + \|\bar{A}_{12} - (Y^{\perp})^TZ\|_F^2. \end{aligned}$$

Assim, o problema (3.14) é equivalente ao problema

$$\begin{aligned} &\text{minimizar } \|\bar{A}_{11} - Y^TZ\|_F^2 + \|\bar{A}_{12} - (Y^{\perp})^TZ\|_F^2 \\ &\text{s.a. } \bar{A}_{11}^T = -\bar{A}_{11},\ \bar{A}_{12} \in \mathbb{R}^{(n-p)\times p}. \end{aligned} \tag{3.18}$$

Logo, a solução do problema (3.18) é $\bar{A}_{11}^* = \frac{1}{2}(Y^TZ - Z^TY)$, $\bar{A}_{12}^* = (Y^{\perp})^TZ$ e $\bar{A}_{22}^* \in \mathbb{R}^{(n-p)\times(n-p)}$. Tomaremos

$$\bar{A}^* = \begin{bmatrix} \bar{A}_{11}^* & -(\bar{A}_{12}^*)^{\mathrm{T}} \\ \bar{A}_{12}^* & 0 \end{bmatrix}.$$

Assim, a matriz A^* é

$$\begin{aligned} A^* &= Q\bar{A}^*Q^T \\ &= \begin{bmatrix} Y & Y^{\perp} \end{bmatrix} \begin{bmatrix} \frac{1}{2}(Y^TZ - Z^TY) & -Z^TY^{\perp} \\ (Y^{\perp})^TZ & 0 \end{bmatrix} \begin{bmatrix} Y^T \\ (Y^{\perp})^T \end{bmatrix} \\ &= \frac{1}{2}(YY^TZY^T - YZ^TYY^T) - YZ^TY^{\perp}(Y^{\perp})^T + Y^{\perp}(Y^{\perp})^TZY^T. \end{aligned}$$

E como $Y^{\perp}(Y^{\perp})^T = I - YY^T$

$$\begin{aligned} A^* =& (ZY^T - YZ^T) - \frac{1}{2}(YY^TZY^T - YZ^TYY^T) \\ =& (ZY^T - YZ^T) + \frac{1}{2}Y(Z^TY - Y^TZ)Y^T. \end{aligned}$$

Portanto, como

$$\begin{aligned} &\text{minimizar } \|Z - X\|_F \\ &\text{s.a. } X \in \mathcal{S}(Y) \end{aligned}$$

é equivalente a (3.14),

$$\begin{aligned} \mathrm{P}_{\mathcal{S}(Y)}Z &= A^*Y \\ &= Z - YZ^TY + \frac{1}{2}Y(Z^TY - Y^TZ) \\ &= Z - Y\frac{1}{2}(Z^TY + Y^TZ). \end{aligned}$$

Note que como A é antissimétrica, então $\mathrm{P}_{\mathcal{S}(Y)}Z$ pertence a $\mathcal{S}(Y)$. ■

Corolário 3.4. *Se $Z = G(Y)Y$ em que $G : \Omega \to \mathbb{R}^{n\times n}$, e $G(Y)^T = G(Y)$, então a solução do problema de minimização 3.14 é dado por*

$$A^* = ZY^T - YZ^T = \begin{bmatrix} Z & Y \end{bmatrix}_{n\times 2p} \begin{bmatrix} Y^T \\ -Z^T \end{bmatrix}_{2p\times n}$$

e

$$P_{\mathcal{S}(Y)}Z = A^*Y = Z - YZ^TY.$$

A seguir, descrevemos as etapas do Algoritmo para resolver o problema (3.1).

3.2 Passo de restauração

O passo de restauração consiste em obter um ponto Y_k que esteja mais próximo do conjunto viável $\mathcal{V}$ e numa vizinhança de $F(X_k)$ obtido na iteração anterior, o qual é descrito através das condições de restauração.

Dado $X_k \in \Omega$, queremos encontrar Y_k tal que

$$\begin{aligned} h(Y_k) &\leq rh(X_k) \\ F(Y_k) - F(X_k) &\leq \beta h(X_k). \end{aligned}$$

Porquanto, $h(X)$ é uma cota superior para $\|H(X)\|$ representa a medida de inviabilidade de um ponto X.

Dado um ponto viável $Y_k \in \mathcal{V}$ e $D_k \in \mathcal{S}(Y_k)$, temos que

$$D_k = A_k Y_k,$$

em que

$$A_k = (P_k D_k)Y_k^T - Y_k(P_k D_k)^T \quad \text{e} \quad P_k = (I - \frac{1}{2}Y_k Y_k^T).$$

O ponto de restauração Y_{k+1} é determinado pelo esquema de Crank-Nicolson

$$Y_{k+1}(t) = Y_k + \frac{t}{2}A_k(Y_k + Y_{k+1}(t)), \tag{3.19}$$

que por sua vez pode ser visto da forma fechada:

$$Y_{k+1}(t) = \mathcal{C}(\tfrac{t}{2}A_k)Y_k,$$

em que $\mathcal{C}(A) := (I - A)^{-1}(I + A)$ é chamada de transformação de Cayley. A transformação $\mathcal{C}$ mapeia matrizes antissimétricas em matrizes ortogonais. A Figura 3.1 ilustra o novo iterando Y_{k+1} mediante a transformação de Cayley. A seguir, apresentamos alguns resultados relacionados a transformação de Cayley.

Teorema 3.5. *Seja $Y_k \in \mathcal{V}$. Se $A_k \in \mathbb{R}^{n\times n}$ é uma matriz antisimétrica, então*

1. *$(I - \frac{t}{2}A_k)$ é não singular para $t > 0$.*
2. *$Y_{k+1}(t) \in St(n, p)$ isto é, $Y_{k+1}(t)^T Y_{k+1}(t) = I$.*
3. *$Y_{k+1}(0) = Y_k$ e $\frac{d}{d\tau}Y_{k+1}(t)\Big|_{\tau=0} = A_k Y_k = D_k$.*

Demonstração. 1. Seja $v \in \text{Ker}(I - \frac{t}{2}A_k)$, então $(I - \frac{t}{2}A_k)v = 0$. Por outro lado, calculamos

$$\begin{aligned}\langle v, A_k v\rangle = \text{traço}(v^T A_k v) &= -\text{traço}(v^T A_k^T v)\\ &= -\langle A_k v, v\rangle = -\langle v, A_k v\rangle.\end{aligned}$$

Assim, obtemos que $\text{traço}(v^T A_k v) = 0$. Segue que a norma $\|v\|^2 = \text{traço}(v^T v - t v^T A_k v) = \text{traço}(v^T(I - \frac{t}{2}A_k)v) = 0$. Portanto, $I - \frac{t}{2}A_k$ é não-singular.

2. Verificaremos a ortogonalidade de $\mathcal{C}(A_k)$:

$$\begin{aligned}
&\mathcal{C}(\tfrac{t}{2}A_k)^T\mathcal{C}(\tfrac{t}{2}A_k)\\
&\quad= \big((I-\frac{t}{2}A_k)^{-1}(I+\frac{t}{2}A_k)\big)^T(I-\frac{t}{2}A_k)^{-1}(I+\frac{t}{2}A_k)\\
&\quad= (I+\frac{t}{2}A_k)^T((I-\frac{t}{2}A_k)^{-1})^T(I-\frac{t}{2}A_k)^{-1}(I+\frac{t}{2}A{-_k})\\
&\quad= (I-\frac{t}{2}A_k)(I+\frac{t}{2}A_k)^{-1}(I-\frac{t}{2}A_k)^{-1}(I+\frac{t}{2}A_k). \quad (3.20)
\end{aligned}$$

Provaremos que

$$\mathcal{C}(\frac{t}{2}A_k) = (I-\frac{t}{2}A_k)^{-1}(I+\frac{t}{2}A_k) = (I+\frac{t}{2}A_k)(I-\frac{t}{2}A_k)^{-1}. \quad (3.21)$$

De fato, como o produto de matrizes comuta $(I-\frac{t}{2}A_k)$ e $(I+\frac{t}{2}A_k)$, isto é,

$$(I-\frac{t}{2}A_k)(I+\frac{t}{2}A_k) = (I+\frac{t}{2}A_k)(I-\frac{t}{2}A_k) \quad (3.22)$$

e pela não singularidade da matriz $(I-\frac{t}{2}A_k)$, multiplicamos ambos lados da igualdade (3.22) à esquerda e à direita pela inversa $(I-\frac{t}{2}A_k)^{-1}$, respectivamente, e obtemos o resultado

$$(I-\frac{t}{2}A_k)^{-1}(I+\frac{t}{2}A_k) = (I+\frac{t}{2}A_k)(I-\frac{t}{2}A_k)^{-1}.$$

Logo, substituindo (3.22) em (3.20), temos

$$\begin{aligned}
&\mathcal{C}(\tfrac{t}{2}A_k)^T\mathcal{C}(\tfrac{t}{2}A_k)\\
&\quad= (I-\frac{t}{2}A_k)(I+\frac{t}{2}A_k)^{-1}(I+\frac{t}{2}A_k)(I-\frac{t}{2}A_k)^{-1}\\
&\quad= I.
\end{aligned}$$

3. Derivando ambos lados da igualdade (3.19), obtemos $Y'_{k+1}(t) = A_k\frac{(Y_k+Y_{k+1}(t))}{2} + \frac{t}{2}A_kY'_{k+1}(t)$. Logo, para $t=0$, temos $Y'_{k+1}(0) = A_kY_k$.

■

Com os resultados teóricos obtidos, vamos adaptar o Algoritmo 2 para o problema matricial (3.1), sendo este descrito no Algoritmo 3.

Algoritmo 3: Restauração Inexata não monótona aplicada ao Problema de minimização matricial.

Dados: γ, $\mu > 0$, $0 \leq \eta_{\min} \leq \eta_{\max} < 1$, $M \in \mathbb{N}$ e $r \in [0, 1)$.

Passo 0. Inicialização

Escolhemos uma aproximação inicial $X_0 \in \mathbb{R}^{n \times p}$, $Y_0 = X_0$ e $\theta_0 \in (0, 1)$. Defina $C_0 = \Phi(x_0, \theta_0)$, $Q_0 = 1$ e $k = 0$.

Passo 1. Escolha do Parâmetro de Penalidade.

Calcule θ_{k+1} como o primeiro termo da sequência $\{\theta_k/2^j\}_{j\in\mathbb{N}}$ tal que

$$\Phi(Y_k, \theta_{k+1}) \leq \Phi(X_k, \theta_{k+1}) + \frac{1}{2}(h(Y_k) - h(X_k)).$$

Passo 2. Direção de descida

Calcule $D_k \in \mathcal{S}(Y_k)$ tal que $Y_k + D_k \in \Omega$

$$\|D_k\|_F \geq \bar{\mu}\|\mathrm{P}_{\mathcal{S}(Y_k)}(\nabla F(Y_k))\|_F \tag{3.23}$$

$$\langle \nabla F(Y_k), D_k \rangle \leq -\mu\|D_k\|_F^2. \tag{3.24}$$

Passo 3. Calcule t_k como o primeiro termo da sequência $\{1/2^j\}_{j\in\mathbb{N}}$ tal que

$$\Phi(Y_k + t_k D_k, \theta_{k+1}) - T_k \leq \frac{(1-r)}{2}(h(y_k) - h(x_k))$$

em que $T_k := \max\{C_k, \Phi(X_k, \theta_{k+1})\}$.

Passo 4. Restauração

Defina $A_k = (I - \frac{1}{2}Y_kY_k^T)D_kY_k^T - Y_kD_k^T(I - \frac{1}{2}Y_kY_k^T)$

$$X_{k+1} = (I + t_kA_k)\,Y_k \tag{3.25}$$

$$\overline{Y}_{k+1} = \left(I - \frac{t_k}{2}A_k\right)^{-1}\left(I + \frac{t_k}{2}A_k\right)Y_k, \tag{3.26}$$

Faça M iterações locais e encontre $Y_{k+1} \in \mathcal{V}$ tal que

$$F(Y_{k+1}) \leq F(\overline{Y}_{k+1}) \tag{3.27}$$

Passo 5.

Escolha $\eta_k \in [\eta_{\min}, \eta_{\max}]$ e defina

$$Q_{k+1} = \eta_kQ_k + 1, \tag{3.28}$$

$$C_{k+1} = (\eta_kQ_kC_k + \Phi(X_{k+1}, \theta_{k+1}))/Q_{k+1}, \tag{3.29}$$

$k := k + 1$ e volte ao Passo 1.

3.3 Cálculo da direção de descida

Escolheremos a direção de descida D_k no subespaço $\mathcal{S}(Y_k)$ tal que $Y_k + D_k \in \Omega$ e satisfaz as condições (3.23) e (3.24) do Algoritmo 3. Consequentemente estas duas condições garantem que D_k satisfaz (2.14) e (2.15) do Algoritmo 1. Em particular se $D_k = 0$, então Y_k é um ponto AGP.

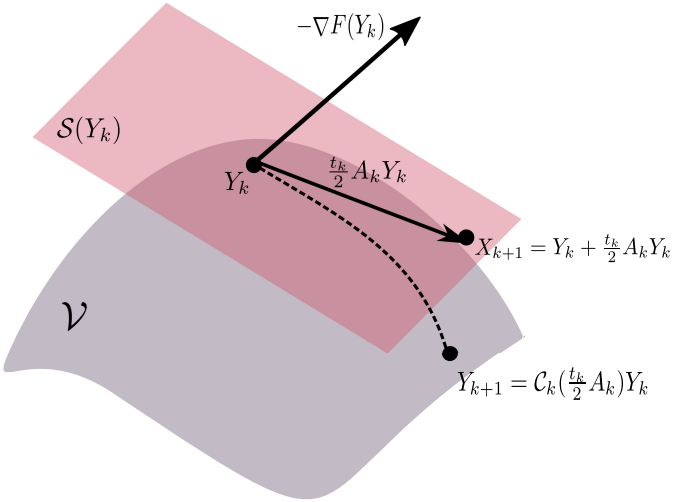

Figura 3.1: Ilustração do k-ésimo passo do Algoritmo 3.

Na prática, calculamos D_k como sendo

$$D_k := \underset{\substack{D \in \mathcal{S}(Y_k) \\ Y_k + D \in \Omega}}{\operatorname{argmin}} \mathcal{Q}_k(Z), \tag{3.30}$$

em que $\mathcal{Q}_k(Z)$ é uma aproximação quadrática para $\mathcal{L}(Y_k + Z, \Lambda_k)$ sendo Λ_k multiplicador de Lagrange associado ao Problema 3.1. Se esta direção satisfaz as duas condições (3.23) e (3.24), então escolhemos D_k como direção de descida do Algoritmo 3. Caso contrário, utilizamos

$$D_k = \frac{1}{\lambda_k^{sp}} \mathrm{P}_{\mathcal{S}(Y_k)}(-\nabla F(Y_k)),$$

em que $\mathrm{P}_{\mathcal{S}(Y_k)} Z$ denota a projeção ortogonal de Z em $\mathcal{S}(Y_k)$ e λ_k^{sp} é o parâmetro espectral introduzido por Barzilai e Borwein em [48] definido por

$$\lambda_k^{sp} = \min\{\max\{\lambda_{\min}, \lambda_k^{BB}\}, \lambda_{\max}\},$$

e

$$\lambda_k^{BB} = \begin{cases} \frac{|\text{traço}(\nabla L(Y_k,\Lambda_k))-\nabla L(Y_{k-1},\Lambda_k))^T(Y_k-Y_{k-1})|}{\|Y_k-Y_{k-1}\|_F^2}, & \text{se k é par} \\ \\ \frac{\|\nabla L(Y_k,\Lambda_k)-\nabla L(Y_{k-1},\Lambda_k)\|_F^2}{|\text{traço}(\nabla L(Y_k,\Lambda_k)-\nabla L(Y_{k-1},\Lambda_k))^T(Y_k-Y_{k-1})|}, & \text{se k é impar.} \end{cases}$$

Note que,esta ultima escolha da direção D_k sempre verifica as duas condições (3.23) e (3.24).

Para encontrar o valor do passo de minimização t_k tal que o valor da função de mérito diminua no subespaço tangente às restrições, utilizamos a caracterização do subespaço $\mathcal{S}(Y_k)$. Como vimos na seção anterior, a busca linear de Armijo pode não oferecer um bom desempenho. Assim utilizamos o método de busca não monótono, baseado em Zhang e Hager [2] aplicado à função de mérito, conforme Algoritmo 2. As condições de minimização do Algoritmo 3 podem ser descritas da seguinte forma: Encontre $t_k > 0$ que satisfaz a desigualdade

$$\Phi(Y_k + tD_k, \theta_{k+1}) \leq T_k + \frac{(1-r)}{2}(h(Y_k) - h(X_k))$$

em que $T_k := \max\{C_k, \Phi(X_k, \theta_{k+1})\}$. O Algoritmo 3 descreve o método de Restauração Inexata não monótono aplicado ao Problema (3.1).

3.4 Análise da convergência do algoritmo proposto

Nesta seção, além de verificarmos que o Algoritmo 3 está bem definido, estudaremos a convergência global do algoritmo proposto, isto é, todo ponto de acumulação é estacionário. Para o problema (3.1), levaremos em consideração as mesmas hipóteses usadas no Capítulo 2.

Agora, mostraremos que, no Passo de Restauração, a sequência $\{Y_k\}$ satisfaz as condições (2.11) e (2.12). Como $Y_k \in \mathcal{V}$, $\forall k$, a condição (2.11) vale para qualquer $r \in [0, 1)$. Daqui em diante, vamos definir

$$h(X) := \|H(X)\|_F. \tag{3.31}$$

Da Álgebra Linear, se $A \in \mathbb{R}^{n\times n}$ é simétrica e semidefinida positiva, então existe uma única matriz simétrica e semidefinida positiva $B \in \mathbb{R}^{n\times n}$ tal que $B^2 = A$. A matriz B é denotada por $A^{1/2}$ [6].

Lema 3.6. *Seja $Y \in \mathcal{V}$ e $X \in \mathcal{T}(Y)$, então $H(X) = X^TX - I$ é simétrica e semidefinida positiva. Portanto, a matriz $H(X)^{1/2}$ está bem definida, $\forall X \in \mathcal{S}(Y)$.*

Demonstração. Veremos que $H(X)$ é semidefinida positiva em $\mathcal{T}(Y)$ para $Y \in \mathcal{V}$.
De fato, seja $X \in \mathcal{T}(Y)$, então X é da forma $X = Y + D,\ D \in \mathcal{S}(Y)$.

$$\begin{aligned} H(X) &= (Y+D)^T(Y+D) - I \\ &= Y^TY + (Y^TD + D^TY + D^TD) - I \end{aligned}$$

Temos que $Y^TY = I$ e $Y \in \mathcal{S}(Y)$, isto é, $Y^TD + D^TY = 0$.
Assim,

$$H(X) = D^TD.$$

Logo, $H(X)$ é semidefinida positiva. Portanto, temos a boa definição de $H(X)^{1/2}$ para cada $X \in \mathcal{S}(Y)$. ■

Teorema 3.7. *Seja $\bar{\beta} > 0$, $Y_{k-1} \in \mathcal{V}$ e $X_k = Y_{k-1} + AY_{k-1} \in \mathcal{T}(Y_{k-1})$, para alguma matriz antissimétrica $A \in \mathbb{R}^{n\times n}$ tal que $\|AY_{k-1}\|_2 \geq 1/\bar{\beta}$. Então $\overline{Y}_k = (I - \frac{1}{2}A)^{-1}(I + \frac{1}{2}A)Y_{k-1} \in \mathcal{V}$ e*

$$\|X_k - \overline{Y}_k\|_2 \leq \bar{\beta}\|H(X_k)\|_2.$$

Demonstração. Como A é antissimétrica, então $(I - \frac{1}{2}A)^{-1}(I + \frac{1}{2}A)$ e $(I + \frac{1}{2}A)(I - \frac{1}{2}A)^{-1}$ são matrizes ortogonais. Além disso, $Y_{k-1} \in \mathcal{V}$, então $\overline{Y}_k = (I - \frac{1}{2}A)^{-1}(I + \frac{1}{2}A)Y_{k-1} \in \mathcal{V}$.
Calculando a norma e por (3.21) segue que

$$\begin{aligned} \|X_k - \overline{Y}_k\|_2 &= \|(I+A)Y_{k-1} - (I - \frac{1}{2}A)^{-1}(I + \frac{1}{2}A)Y_{k-1}\|_2 \\ &= \|(I + \frac{1}{2}A)Y_{k-1} + \frac{1}{2}AY_{k-1} - (I - \frac{1}{2}A)^{-1}(I + \frac{1}{2}A)Y_{k-1}\|_2 \\ &= \|(I + \frac{1}{2}A)Y_{k-1} - (I + \frac{1}{2}A)(I - \frac{1}{2}A)^{-1}Y_{k-1} + \frac{1}{2}AY_{k-1}\|_2 \\ &= \|(I + \frac{1}{2}A)(I - \frac{1}{2}A)^{-1}\left((I - \frac{1}{2}A)Y_{k-1} - Y_{k-1}\right) + \frac{1}{2}AY_{k-1}\|_2 \\ &= \|(I + \frac{1}{2}A)(I - \frac{1}{2}A)^{-1}\left(-\frac{1}{2}AY_{k-1}\right) + \frac{1}{2}AY_{k-1}\|_2 \\ &\leq \|(I + \frac{1}{2}A)(I - \frac{1}{2}A)^{-1}\left(-\frac{1}{2}AY_{k-1}\right)\|_2 + \|\frac{1}{2}AY_{k-1}\|_2. \end{aligned}$$

Desde que a matriz $(I+\frac{1}{2}A)(I-\frac{1}{2}A)^{-1}$ é ortogonal e utilizando o fato de que a norma 2 é invariante por matriz ortogonal, obtém-se

$$\begin{aligned}\|X_k - \overline{Y}_k\|_2 &\leq \|-\frac{1}{2}AY_{k-1}\|_2 + \|\frac{1}{2}AY_{k-1}\|_2 \\ &= \|AY_{k-1}\|_2.\end{aligned}$$

Logo,

$$\|X_k - \overline{Y}_k\|_2 \leq \|AY_{k-1}\|_2 \tag{3.32}$$

Por outro lado,

$$\begin{aligned}\|X_k^T X_k - I\|_2 &= \|Y_{k-1}^T (I+A)^T (I+A) Y_{k-1} - I\|_2 \\ &= \|Y_{k-1}^T (I-A)(I+A) Y_{k-1} - I\|_2 \\ &= \|Y_{k-1}^T (I-A^2) Y_{k-1} - I\|_2 \\ &= \|(AY_{k-1})^T (AY_{k-1})\|_2 \\ &= \|AY_{k-1}\|_2^2.\end{aligned}$$

Usando (3.32) e a hipótese $\|AY_{k-1}\| \geq 1/\bar{\beta}$, obtemos

$$\|X_k - \overline{Y}_k\|_2 \leq \|AY_{k-1}\|_2 \leq \bar{\beta}\|AY_{k-1}\|_2^2 = \bar{\beta}\|X_k^T X_k - I\|_2.$$

Logo,

$$\|X_k - \overline{Y}_k\|_2 \leq \bar{\beta}\|H(X_k)\|_2.$$

■

O seguinte Corolário estabelece que a condição (2.12) é satisfeita pelas sequências geradas pelo Algoritmo 3.

Corolário 3.8. *Com as mesmas hipóteses do Teorema 3.7 tem-se que*

$$F(Y_k) - F(X_k) \leq \beta\|H(X_k)\|_F.$$

Demonstração. Como ∇F é continua em Ω compacto, então existe $L_1 > 0$ tal que

$$\|\nabla F(X)\| \leq L_1, \quad \forall X \in \Omega. \tag{3.33}$$

Então, calculamos a diferença usando o Teorema Fundamental do Cálculo Integral, a desigualdade (3.33) e Teorema 3.7 obtemos

$$\begin{aligned}F(\overline{Y}_k) - F(X_k) &= \int_0^1 \langle \nabla F(X_k + s(\overline{Y}_k - X_k)), \overline{Y}_k - X_k \rangle ds \\ &\leq \int_0^1 \|\nabla F(X_k + s(\overline{Y}_k - X_k))\|_F \|\overline{Y}_k - X_k\|_F ds \\ &\leq L_1 \|\overline{Y}_k - X_k\|_F \\ &\leq L_1 \sqrt{p}\bar{\beta}\|H(X_k)\|_2,\end{aligned}$$

onde temos usado a equivalência entre normas.
Também, pela condição (3.27) temos que $F(Y_k) \leq F(\overline{Y}_k), \forall k$, então

$$F(Y_k) - F(X_k) \leq F(\overline{Y}_k) - F(X_k).$$

Logo,

$$F(Y_k) - F(X_k) \leq \beta \|H(X_k)\|_2 \leq \beta \|H(X_k)\|_F \quad \text{sendo} \quad \beta = L_1 \sqrt{p}\bar{\beta}.$$

■

Os resultados obtidos anteriormente garantem a existência de β e que a transformação de Cayley preserva ortogonalidade.

Teorema 3.9. *Seja $X_k = U_k \Sigma_k V_k^T$ a decomposição em valores singulares reduzida de $X_k \in \mathbb{R}^{n \times p}$. Se $\overline{Y}_k$ o elemento de $\mathcal{V}$ mais próximo de X_k ao conjunto viável, isto é,*

$$\overline{Y}_k = \underset{Y^T Y = I}{\operatorname{argmin}} \|Y - X_k\|_F,$$

então

$$\overline{Y}_k = U_k V_k^T,$$

$$\|X_k - \overline{Y}_k\|_F \leq \|H(X_k)\|_F,$$

e

$$F(Y_k) - F(X_k) \leq \beta \|H(X_k)\|_F, \quad \textit{com} \quad \beta = L.$$

Demonstração. Segundo o resultado de [6](Seção 12.4.1) a solução $\overline{Y}_k$ é da forma $\overline{Y}_k = U_k V_k^T$. Utilizando o fato que a norma Frobenius é invariante por matrizes ortogonais e denotando σ_k^j o valor singular de X_k, temos que

$$\begin{aligned} \|X_k - \overline{Y}_k\|_F^2 &= \|\Sigma_k - I\|_F^2 \\ &= \sum_{j=1}^{p} (\sigma_k^j - 1)^2. \end{aligned}$$

Por outro lado, calculando a norma

$$\|X_k^T X_k - I\|_F^2 = \|\Sigma_k^2 - I\|_F^2 = \sum_{j=1}^{p} ((\sigma_k^j)^2 - 1)^2.$$

Como cada valor singular σ_k^j é não negativo, temos que $|\sigma_k^j - 1| \leq |(\sigma_k^j)^2 - 1|$, para todo $j \in \{1, \ldots, p\}$. Assim,

$$\|X_k - \overline{Y}_k\|_F^2 = \sum_{j=1}^{p} (\sigma_k^j - 1)^2 \leq \sum_{j=1}^{p} ((\sigma_k^j)^2 - 1)^2 = \|X_k^T X_k - I\|_F^2.$$

Seguindo o procedimento análogo a Corolario 3.8 e utilizando a desigualdade acima, obtém-se que

$$F(Y_k) - F(X_k) \leq \beta \|H(X_k)\|_F, \quad \text{com} \quad \beta = L.$$

■

Observação:
Se $\|A_k Y_k\|_2 \geq \frac{1}{\beta}$, então, pelo Corolario 3.8, podemos restaurar $\overline{Y}_{k+1}$ via a transformação de Cayley, caso contrário usamos o Teorema 3.9 e, neste caso $\overline{Y}_{k+1} = U_k V_k^T$. Portanto, as condições, (2.11) e (2.12), são sempre válidas no Algoritmo 3.

Definição 3.10. *Dizemos que $X \in \mathcal{V}$ satisfaz CRCQ se para todo $I \subset \{1, \ldots, p\}$ temos que $\{\nabla H_{i,j}(X)\}_{i,j \in I}$ tem posto constante, então, para qualquer Y em uma vizinhança de X, temos que $\{\nabla H_{i,j}(Y)\}_{i,j \in I}$ tem posto constante.*

A seguir, mostraremos que todo $X \in \Omega$ no conjunto viável satisfaz a condição CRCQ (condição de qualificação de posto constante).

Lema 3.11. *Seja $H_{i,j}(X) = X_i^T X_j - \delta_{ij}$, $i, j = 1, \cdots, p$, em que $X_i \in \mathbb{R}^n$ é a i-ésima coluna de X e $\delta_{ij} = \begin{cases} 1, & se \quad i \neq j \\ 0, & se \quad i = j \end{cases}$ Então, o conjunto $\Gamma = \{\nabla H_{i,j}(X)\}_{i \geq j}$ é linearmente independente.*
Além disso,

$$dim\left(span\{\nabla H_{ij}(X)\}_{i \geq j} \right) = \frac{p(p+1)}{2}. \tag{3.34}$$

Demonstração. Seja $X \in \mathcal{V}$. Organizaremos a matriz $X = [X_1 \cdots X_p] \in \mathbb{R}^{n \times p}$ na forma vetorial:

$$\text{vec}(X) = \begin{pmatrix} X_1 \\ \vdots \\ X_p \end{pmatrix} \in \mathbb{R}^{pn}.$$

Calcule o $\nabla H_{j,j}(vec(X))$:

$$\nabla H_{j,j}(vec(X)) = \begin{pmatrix} 0 \\ \vdots \\ 2X_j \\ \vdots \\ 0 \end{pmatrix}.$$

Note que $\nabla H_{i,j}(vec(X)) = \nabla H_{j,i}(vec(X))$, portanto, calculamos para o caso $i > j$,

$$\nabla H_{ij}(vec(X)) = \begin{pmatrix} 0 \\ \vdots \\ X_i \\ \vdots \\ X_j \\ \vdots \\ 0 \end{pmatrix}.$$

Então, para j fixo, $\{\nabla H_{ij}(X)\}_{i \geq j}$ é linearmente independente.
De fato, considere a combinação linear

$$c_j \nabla H_{jj}(vec(X)) + c_{j+1} \nabla H_{(j+1)j}(vec(X)) + \cdots + c_p \nabla H_{pj}(vec(X)) = 0.$$

Então

$$\begin{pmatrix} 0 \\ \vdots \\ 2c_j X_j + \ldots + c_p X_p \\ c_{j+1} X_j \\ \vdots \\ c_p X_j \end{pmatrix} = 0.$$

Note que, $X_j^T X_j = 1$ e $X_i^T X_j = 0$, se $i \neq j$. Então, $c_j = c_{j+1} = \ldots = c_p = 0$, para j fixo, isto é, $\{\nabla H_{i,j}(X)\}_{i \geq j}$ é linearmente independente. O mesmo é válido se deixarmos j livre: consideramos a combinação linear

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

isto é,

$$\begin{pmatrix} 2c_{11}X_1 \\ c_{21}X_1 + 2c_{22}X_2 \\ \vdots \\ c_{p1}X_1 + c_{p2}X_2 + \ldots + 2c_{pp}X_p \end{pmatrix} = 0.$$

Do fato de que para j fixo temos que a independência linear é satisfeita, temos que $\Gamma = \{\nabla H_{i,j}(X)\}_{i \geq j}$ é linearmente independente.

■

Teorema 3.12. *Se $X \in \mathcal{V}$, então X satisfaz CRCQ [49]. Isto é, se para todo $I \subset \{1, \ldots, p\}$ vale $dim(span\{\nabla H_{ij}(X)\}_{i,j \in I}) = d$, então para qualquer Y e $I \subset \{1, \ldots, p\}$ em uma vizinhança de X temos que*

$$dim(span\{\nabla H_{ij}(Y)\}_{i,j \in I}) = d.$$

Demonstração. Seja $X \in \mathcal{V}$. Pelo Lema anterior temos que $\{\nabla H_{ij}(X)\}_{i \geq j}$ é um conjunto linearmente independente. Demonstraremos que $\{\nabla H_{ij}(Y)\}_{i \geq j}$ é linearmente independente para Y em uma vizinhança de X com raio δ, para algum $\delta > 0$. Suponha, por contradição, que existe c_{ij} não nulo tal que

$$\sum_{j=1}^{p} \sum_{i \geq j}^{p} c_{ij} \nabla H_{ij}(Y) = 0,$$

em que $\|Y - X\| < \delta$, para todo $\delta > 0$. Pela continuidade de $\{\nabla H_{ij}\}$ temos que para todo $\epsilon > 0$, existe $\delta > 0$ tal que, se $\|X - Y\|_F < \delta$, então

$$\|\nabla H_{ij}(X) - \nabla H_{ij}(Y)\|_F < \epsilon / M,$$

em que $M = \sum_{j=1}^{p} \sum_{i \geq j}^{p} c_{ij} > 0$.

Logo,

$$\begin{aligned} \|\sum_{j=1}^{p} \sum_{i \geq j}^{p} c_{ij} \nabla H_{ij}(X)\|_F &= \sum_{j=1}^{p} \sum_{i \geq j}^{p} c_{ij} \|\nabla H_{ij}(X) - \nabla H_{ij}(Y)\|_F \\ &< \sum_{j=1}^{p} \sum_{i \geq j}^{p} c_{ij} \epsilon / M = \epsilon. \end{aligned}$$

Portanto, existe c_{ij} não nulo tal que $\sum_{j=1}^{p} \sum_{i \geq j}^{p} c_{ij} \nabla H_{ij}(X) = 0$, o qual é uma contradição com $\{\nabla H_{ij}(X)\}_{i \geq j}$ ser linearmente independente.

Se $I \subset \{i \geq j : i, j = 1, \ldots, p\}$ então pelo resultado anterior a condição de posto constante é satisfeita para todo $X \in \mathcal{V}$. Note que para $I_0 = \{i < j : i, j = 1, \ldots, p\}$ temos que $\{\nabla H_{ij}(X)\}_{i,j \in I_0} \subset \{\nabla H_{ij}(X)\}_{i \geq j}$ pois verifica-se $\nabla H_{kl}(X) = \nabla H_{lk}(X)$, em que $k < l$. ■

Teorema 3.13. *Suponha que a hipótese (**H1**) é satisfeita. Então o Algoritmo 3 está bem definido e sejam as sequências $\{X_k\}$ e $\{Y_k\}$, geradas pelo Algoritmo 3 e $\lim_{k\to\infty} h(X_k) = \lim_{k\to\infty} h(Y_k) = 0$. Ainda, todo ponto de acumulação X^* de $\{X_k\}$ satisfaz $h(X^*) = 0$.*

Demonstração. A boa definição do Algoritmo 3 segue do Teorema 2.8. A prova de todo ponto de acumulação ser viável, segue do Teorema 2.11 ■

A condição (3.23) exige que o tamanho D_k deve ser pelo menos maior a $\bar{\mu}\|\mathrm{P}_{\mathcal{S}(Y_k)}(\nabla F(Y_k))\|_F$, com $\bar{\mu} > 0$. Esta condição evita direções de busca D_k com tamanho pequeno. Por último, a condição (3.24) considera direções de descida, em que o ângulo entre o D_k e $\nabla F(Y_k)$ seja menor que $\pi/2$.

O seguinte teorema mostra a convergência do Algoritmo de restauração inexata não monótona.

Teorema 3.14. *Seja $\{X_k\}$ a sequência gerada pelo Algoritmo 3 e suponha a validade da hipótese (**H1**). Então, todo ponto limite X^* de $\{X_k\}$ satisfaz a condição de otimalidade AGP. Ainda, as condições KKT são satisfeitas em X^*.*

Demonstração. Para nosso problema, como $D_k \in \mathcal{S}(Y_k)$, temos que

$$\begin{aligned} H(X_{k+1}) &= H(Y_k + t_k D_k) - I \\ &= (Y_k + t_k D_k)^T (Y_k + t_k D_k) - I \\ &= Y_k^T Y_k - I + t_k (Y_k^T D_k + D_k^T Y_k) + t_k^2 D_k^T D_k. \end{aligned}$$

Também, como $D_k \in \mathcal{S}(Y_k)$ e $Y_k \in \mathcal{V}$, segue que, $Y_k^T D_k + D_k^T Y_k = 0$ e $Y_K^T Y_k - I = 0$, respectivamente. Logo,

$$H(X_{k+1}) = t_k^2 D_k^T D_k. \tag{3.35}$$

Observe que $h(X) = \|H(X)\|_F$ e pelo item 2 do Teorema 3.13, temos que

$$\lim_{k\to\infty} H(X_k) = 0.$$

Usando o argumento anterior e (2.11), temos

$$\lim_{k\to\infty} H(Y_k) = 0.$$

Tomando $K_1 \underset{\infty}{\subset} \mathbb{N}$ tal que $\lim_{k\in K_1} X_k = X^*$, e pelo Teorema 2.8, a sequência $\{t_k\}$ admite uma subsequência convergente para um valor acima de zero, isto é, existe $K_2 \underset{\infty}{\subset} K_1$ tal que $\lim_{k\in K_2} t_k \geq \bar{t} > 0$. Tomando o limite em (3.35) e utilizando $\lim_{k\in K_2} H(X_k) = \lim_{k\in K_2} H(Y_k) = 0$, obtemos

$$\lim_{k\in K_2} D_k^T D_k = 0.$$

Calculando a norma de Frobenius, $\lim_{k\in K_2} \|D_k\|_F^2 = \lim_{k\in K_2} \text{traço}((D_k)^T D_k) = 0$. Então $\lim_{k\in K_2} D_k = 0$.
Usando este último resultado e a condição (3.23), obtemos que

$$\lim_{k\in K_2} \|\mathrm{P}_{\mathcal{S}(Y_k)}(\nabla F(Y_k))\|_F = 0. \tag{3.36}$$

Assim, X^* é um ponto que satisfaz a condição AGP. Além disso, como o ponto X^* satisfaz CRCQ. Consequentemente, X^* verifica a condição CPLD. Portanto, segue de Gomes [33](Teorema 3.11) que X^* é um ponto KKT. ■

O seguinte resultado mostra que $Y_{k+1}(t) := (I - \frac{t}{2}A_k)^{-1}(I + \frac{t}{2}A_k)Y_k$ é um caminho viável de descida para F.

Teorema 3.15. *Seja $\{A_k\}$ gerada no Passo 4 do Algoritmo 3. Então, $Y_{k+1}(t) = (I - \frac{t}{2}A_k)^{-1}(I + \frac{t}{2}A_k)Y_k$ é um caminho de descida a partir de $Y_{k+1}(0) = Y_k$ isto é,*

$$\langle \nabla F(Y_{k+1}(0)), Y'_{k+1}(0)\rangle = \langle \nabla F(Y_k), D_k\rangle < 0 \tag{3.37}$$

Demonstração. Pela definição de $Y_{k+1}(t)$, temos $(I - \frac{t}{2}A_k)Y_{k+1}(t) = (I + \frac{t}{2}A_k)Y_k$ e, derivando com respeito a t em ambos lados,

$$\frac{\mathrm{d}}{\mathrm{dt}}((I - \frac{t}{2}A_k)Y_{k+1}(t)) = \frac{1}{2}A_kY_k.$$

Assim,

$$Y'_{k+1}(t) = \frac{1}{2}(I - \frac{t}{2}A_k)^{-1}A_k(Y_k + Y_{k+1}(t)).$$

Note que $Y_{k+1}(0) = Y_k$, então, para $t = 0$, obtemos $Y'_{k+1}(0) = A_k Y_k$. Além disso, podemos verificar que $A_k Y_k = D_k$ com

$$\begin{aligned} A_k Y_k &= (P_k D_k Y_k^T + Y_k D_k^T P_k) Y_k \\ &= P_k D_k + Y_k D_k^T P_k Y_k \\ &= D_k - \frac{1}{2} Y_k Y_k^T D_k + \frac{1}{2} Y_k D_k^T Y_k \\ &= D_k - \frac{1}{2} Y_k (Y_k^T D_k + D_k^T Y_k). \end{aligned}$$

Como $D_k \in \mathcal{S}(Y_k)$, então $Y'_{k+1}(0) = A_k Y_k = D_k$. Portanto, pela condição (3.23)

$$\langle \nabla F(Y_{k+1}(0)), Y'_{k+1}(0) \rangle = \langle \nabla F(Y_k), D_k \rangle < 0.$$

■

Teorema 3.16. *Assuma que os passos 1 e 3 do Algoritmo são satisfeitos para todo $k \in \mathbb{N}$. Assuma que existe $c > 0, \zeta \in [0, 1)$ tal que, para todo $k \in \mathbb{N}$, existe $\Lambda^k \in \mathbb{R}^{p \times p}, \zeta_k \in [0, \zeta]$ tal que*

$$\begin{aligned} \|\nabla \mathcal{L}(Y_k + D_k, \Lambda_{k+1})\| &\leq \zeta_k \|\nabla \mathcal{L}(Y_k, \Lambda_k)\| \\ \|D_k\| + \|\Lambda_{k+1} - \Lambda_k\| &\leq c \|\nabla \mathcal{L}(Y_k, \Lambda_k)\|. \end{aligned}$$

Também, assuma que $(\bar{X}, \bar{\Lambda})$ é um ponto estacionário e que $t_k = 1$ para k suficientemente grande. Então,

- *Se $\zeta = 0$, a convergência é R-quadrática.*

Este resultado é consequência do Teorema 2.6.
Com base no teorema 3.16, na fase de restauração fazemos M iterações locais com $t_k = 1$ para reduzir o número de avaliações de função do Algoritmo de RI e conseguir um $Y_k \in \mathcal{V}$ tal que $F(Y_k) \leq F(\overline{Y}_k)$.

3.5 Critério de parada

O próximo teorema estabelece uma equivalência entre as condições KKT do Problema (3.1) e projeção no subespaço tangente. Para isto, definimos alguns conceitos importantes.
Dizemos que $X \in \Omega$ satisfaz a condição KKT se existe $\Lambda \in \mathbb{R}^{p \times p}$ tal que

$$\begin{cases} \nabla F(X) + X\Lambda = 0 \\ H(X) = X^T X - I = 0. \end{cases}$$

O seguinte teorema estabelece condições equivalentes para que uma matriz $X^* \in \Omega$ seja um ponto estacionário.

Teorema 3.17. *Dado $X^* \in \mathcal{V}$, as seguintes afirmações são equivalentes.*

1. *X^* satisfaz as condições KKT de* (3.1).
2. $-2\nabla F(X^*) + X^*((X^*)^T\nabla F(X^*) + \nabla F(X^*)^T X^*) = 0.$
3. $\nabla F(X^*) = X^* S$, *para alguma matriz simétrica $S \in \mathbb{R}^{p\times p}$.*

Demonstração. Mostraremos que (1) implica em (2). Se X^* satisfaz a condição KKT, então existe Λ tal que

$$\nabla F(X^*) + \langle \nabla H(X^*)^T, \Lambda\rangle = 0, \tag{3.38}$$

isto é, $-\nabla F(X^*) \in \mathrm{Im}(\nabla H(X^*)^T) = \mathrm{Ker}(\nabla H(X^*))^\perp$, então

$$\mathrm{P}_{\mathcal{S}(X^*)}(-\nabla F(X^*)) = 0$$

e usando a forma fechada da projeção (3.17), temos que

$$-2\nabla F(X^*) + X^*((X^*)^T\nabla F(X^*) + \nabla F(X^*)^T X^*) = 0.$$

Demonstraremos que (2) implica em (3). Pela hipótese, temos que

$$\nabla F(X^*) = X^*\left(\frac{(X^*)^T\nabla F(X^*) + \nabla F(X^*)^T X^*}{2}\right).$$

Assim, o gradiente pode ser visto como $\nabla F(X^*) = X^* S$, em que

$$S := \frac{(X^*)^T\nabla F(X^*) + \nabla F(X^*)^T X^*}{2}.$$

Por último, temos que (3) implica em (1). De fato, para todo $Z \in \mathrm{Ker}(\nabla H(X^*))$,

$$\begin{aligned}\langle \nabla F(X^*), Z\rangle &= \langle X^* S, Z\rangle \\ &= \langle S, (X^*)^T Z\rangle \\ &= 0,\end{aligned}$$

pois $(X^*)^T Z + Z^T(X^*) = 0$. Assim, $\nabla F(X^*) \in \mathrm{Im}(\nabla H(X^*)^T)$. ∎

3.6 Estimativa do multiplicador de Lagrange

A direção de descida D_k na iteração k pode ser calculada a partir do subproblema no subespaço tangente às restrições

$$D_k := \underset{Z \in \mathcal{S}(Y_k)}{\operatorname{argmin}} \mathcal{Q}(\mathcal{Z}),$$

em que $\mathcal{Q}(Z)$ é uma aproximação quadrática para $(Y_k + Z, \Lambda_k)$. Portanto, é importante estimar os multiplicadores de Lagrange Λ_k a cada iteração, pois deles depende o cálculo de D_k. A ideia para estimar o multiplicador é aproximar o valor da função objetivo no conjunto viável usando a função de Lagrange definida no subespaço tangente. Dado $X \in \mathcal{S}(Y)$, o valor da função de Lagrange em X deve aproximar o valor da função objetivo num ponto viável mais próximo de X. A iteração de Newton aplicada à $H(X) = 0$ é uma ferramenta que nos permite encontrar um ponto viável próximo de X. Neste caso, uma iteração de Newton para encontrar este ponto seria

$$\bar{X} = X - (X^\dagger)^T \mathcal{H}(X),$$

em que $X^\dagger := (X^T X)^{-1} X^T$ e $\mathcal{H}(X) := \frac{1}{2}(X^T X - I)$.
O valor de F em $\bar{X}$ é aproximadamente dado por

$$F(\bar{X}) \approx F(X) + \langle \nabla F(X), \bar{X} - X \rangle. \tag{3.39}$$

Agora,

$$\begin{aligned} \langle \nabla F(X), \bar{X} - X \rangle =& \text{traço}(\nabla F(X)^T (\bar{X} - X)) \\ =& - \text{traço}(\nabla F(X)^T (X^\dagger)^T \mathcal{H}(X)). \end{aligned} \tag{3.40}$$

Por outro lado,

$$\begin{aligned} \text{traço}(\nabla F(X)^T (X^\dagger)^T \mathcal{H}(X)) =& \text{traço}(\mathcal{H}(X)^T X^\dagger \nabla F(X)) \\ =& \text{traço}(X^\dagger \nabla F(X) \mathcal{H}(X)). \end{aligned} \tag{3.41}$$

Substituindo (3.41) em (3.40), temos que

$$\begin{aligned}\langle \nabla F(X), \bar{X} - X\rangle =& -\frac{1}{2}[\text{traço}(\nabla F(X)^T (X^\dagger)^T \mathcal{H}(X)) \\ &+ \text{traço}(X^\dagger \nabla F(X)\mathcal{H}(X))] \\ =& -\frac{1}{2}[\langle X^\dagger \nabla F(X), \mathcal{H}(X)\rangle \\ &+ \langle \nabla F(X)^T (X^\dagger)^T, \mathcal{H}(X)\rangle] \\ =& -\left\langle \frac{X^\dagger \nabla F(X) + \nabla F(X)^T (X^\dagger)^T}{2}, \mathcal{H}(X)\right\rangle. \end{aligned} \tag{3.42}$$

Substituindo (3.42) em (3.39), obtemos

$$F(\bar{X}) \approx F(X) + \left\langle \frac{-(X^\dagger \nabla F(X) + \nabla F(X)^T (X^\dagger)^T)}{2}, \mathcal{H}(X)\right\rangle.$$

Como queremos calcular o valor dos multiplicadores de Lagrange na fase de otimização, temos o seguinte:

$$\Lambda_k = -\frac{Y_k^\dagger \nabla F(Y_k) + \nabla F(Y_k)^T (Y_k^\dagger)^T}{2}.$$

Além disso, como o ponto Y_k satisfaz $Y_k^T Y_k = I$, então $Y_k^+ = Y_k^T$ e

$$\Lambda_k = -\frac{Y_k^T \nabla F(Y_k) + \nabla F(Y_k)^T Y_k}{2}. \tag{3.43}$$

3.7 Método dos gradientes conjugados com restrições lineares

Nesta seção estudaremos o método dos gradientes conjugados que calcula uma solução aproximada do subproblema

$$\begin{aligned} \text{minimizar} \quad & \mathcal{Q}_k(Z) \\ \text{s.a.} \quad & Z \in \mathcal{S}(Y_k) \end{aligned}$$

em que $\mathcal{Q}_k(Z)$ é uma aproximação quadrática do Lagrangiano no ponto $Y_k + Z$.

Considere o problema

$$\begin{aligned} \text{minimizar} \quad & \frac{1}{2}x^T Gx + x^T w \\ \text{s.a.} \quad & C^T x = b, \end{aligned} \tag{3.44}$$

onde G é uma matriz simétrica e definida positiva, $C \in \mathbb{R}^{n\times m}$, com $m < n$ e de posto m, isto é, as colunas de C são linearmente independentes.

Shariff [3] desenvolveu um método dos gradientes conjugados para resolver (3.44), o qual é considerado uma extensão do Algoritmo de Fletcher e Reeves. Este método de direções conjugadas consiste em, dada uma aproximação inicial x_0 para a solução de (3.44) e seu resíduo $g_0 = Gx_0 - w$, escolhermos a primeira direção de busca

$$d_0 = -Hg_0,$$

em que H é a matriz de projeção no subespaço nulo de C^T, ao longo da qual será feita uma busca linear exata, a fim de obter o novo iterado x_1. Depois, calculamos uma nova direção d_1 conjugada a d_0, ao longo da qual obtemos o novo ponto x_2. De forma recursiva, a iteração continua até atingir uma solução do problema (3.44).

Segundo o artigo de Shariff [3], garante-se que o método converge em máximo $n - m$ iterações, com taxa de convergência que depende do espalhamento dos autovalores de G. Este método é descrito no Algoritmo 4.

3.7.1 Descrição do algoritmo de Shariff

Suporemos que $Z \in \mathbb{R}^{n\times(m-n)}$ é uma matriz cujas colunas geram o espaço nulo de C^T, ou seja, $C^T Z = 0$ e $W \in \mathbb{R}^{n\times m}$ é uma matriz tal que as colunas geram uma base da imagem de C. Assim, as colunas de $[Z, W]$ geram o espaço $\mathbb{R}^n$. Todo $x \in \mathbb{R}^n$ pode ser decomposto como:

$$x = Zy + Wq, \tag{3.45}$$

para qualquer $y \in \mathbb{R}^{n-m}$ e $q \in \mathbb{R}^m$. Se x é algum ponto viável, então substituindo (3.45) em $C^T x = b$, temos que

$$C^T x = C^T(Zy + Wq) = b$$

e como $C^T Zy = 0$, então $C^T Wq = b$.

Pela definição de W, a matriz $C^T W$ é não-singular, então existe um único q^* tal que $C^T W q^* = b$. Portanto, x pode ser escrito como

$$x = Zy + Wq^*,$$

ou seja, o Problema (3.44) se reduz a um problema de dimensão $n - m$. Toda direção viável pode ser escrita como uma combinação linear das colunas de Z, ou seja, pode-se escrever

$$z = Zg, \tag{3.46}$$

para algum $g \in \mathbb{R}^{(n-m)}$. Considere o gradiente na direção Y

$$g = -\nabla_Y F(X) = -Z^T \nabla F(X), \tag{3.47}$$

e substituindo (3.47) em (3.46), obtemos $z = -ZZ^T \nabla F(x)$. Repare que a matriz de projeção $P = ZZ^T$ não é única e não satisfaz $PP = P$. Em particular, considere a projeção ortogonal

$$P_m = Z(Z^T Z)^{-1} Z^T. \tag{3.48}$$

Se as colunas de Z são ortonormais, isto é, $Z^T Z = I$, então

$$z = -P_m g.$$

Dado um ponto inicial $x_0 \in \{x \in \mathbb{R}^n : C^T x = b\}$ e um conjunto de direções G conjugadas $\{d_0, d_1, \ldots, d_{n-m-1}\}$, utilizaremos a relação de recorrência

$$x_{i+1} = x_i + \alpha_i d_i, \tag{3.49}$$

em que

$$\alpha_i := \underset{\alpha \in \mathbb{R}}{\operatorname{argmin}} F(x_i + \alpha d_i).$$

Agora, geraremos as direções G conjugadas. Dado $g_0 = Gx_0 + w$, $z_0 = Pg_0$ e para $i = 0, \ldots, n - m - 2$

$$d_{i+1} = z_{i+1} + \beta_i d_i,$$

em que x_{i+1} é dado por (3.49), z_{i+1} é o gradiente projetado e β_i é calculado de tal forma que a direção d_{i+1} seja G-conjugada com d_i.

Algoritmo 4: Gradiente Conjugado com restrições lineares de igualdade.

Entrada: $\epsilon > 0$;
Escolha x_0 que satisfaz $C^T x = b$;
Faça $g_0 = Gx_0 - w$, $z_0 = Hg_0$, $d_0 = -z_0$, $i = 0$;

Enquanto $\|d_i\| > \epsilon \times \|d_0\|$

$\alpha_i = \frac{-g_i^T d_i}{d_i^T G d_i}$

$x_{i+1} = x_i + \alpha_i d_i$;

$g_{i+1} = g_i + \alpha_i G d_i$;

$z_{i+1} = H g_{i+1}$;

$\beta_i = \dfrac{g_{i+1}^T z_{i+1}}{g_i^T z_i}$;

$d_{i+1} = -z_{i+1} + \beta_i d_i$;

$i = i + 1$;

Saída: Uma aproximação x^* para a solução de (3.44).
Fim

No artigo [3], temos a convergência e a taxa de convergência do método. A prova dos resultados apresentados abaixo pode ser encontrada nos Teoremas [3].

Teorema 3.18. *Para qualquer $x_0 \in \{x \in \mathbb{R}^n : C^T x = b\}$, a sequência gerada pelo Algoritmo 4 converge para a solução x^* do problema* (3.44) *em no máximo $n - m$ iterações.*

Demonstração. A prova deste teorema encontra-se em Shariff [3](Teorema 2). ■

Teorema 3.19. *Se $Z^T G Z$ tem autovalores $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$, temos*

$$\|x_k - x^*\|_G \leq T_k\Big(\frac{\lambda_{n-m} + \lambda_1}{\lambda_{n-m} - \lambda_1}\Big)^{-1} \|x_0 - x^*\|_G,$$

em que $\|x\|_G^2 = x^T G x$, T_k é o polinômio de Chebyshev de grau k. Além disso, para qualquer $\epsilon > 0$, se $\rho(\epsilon)$ é definido como sendo o menor inteiro k tal que

$$\|x_k - x^*\|_G \leq \epsilon \|x_o - x^*\|_G,$$

então $\rho(\epsilon) \leq \frac{1}{2}\sqrt{c}\, ln(2/\epsilon) + 1$, *em que* c *é o número de condição de* $Z^T G Z$.

Demonstração. A prova deste teorema encontra-se em Shariff [3](Teorema 3).

■

A seguir, apresentamos a notação que será utilizada ao longo do texto.
Considere $\Psi : U \subset \mathbb{R}^{n\times p} \to \mathbb{R}$ uma função diferenciável no ponto $X \in U$, em que denotamos o gradiente de Ψ por $\mathcal{D}\Psi = \left(\frac{\partial \Psi}{\partial X_{i,j}}\right)_{n\times p}$.

Seja a função $\varphi : U \subset \mathbb{R}^{n\times p} \to \mathbb{R}^{n\times p}$ e denote a derivada de φ no ponto $X \in U$ na direção Z, como

$$\mathcal{D}\varphi(X)(Z) := \lim_{t\to 0} \frac{\varphi(X + tZ) - \varphi(X)}{t} \in \mathbb{R}^{n\times p}. \tag{3.50}$$

Vale observar que, no Passo 2 do Algoritmo 3 de Restauração Inexata não monótono, a direção $D_k \in \mathcal{S}(Y_k)$ pode ser calculada como solução do subproblema

$$\begin{array}{cc} \text{minimizar} & \mathcal{Q}_k(Z) \\ \text{s.a.} & Z \in \mathcal{S}(Y_k) \end{array} \tag{3.51}$$

em que $\mathcal{Q}_k(Z) := \frac{1}{2}\langle Z, \mathcal{D}(\mathcal{D}\mathcal{L}(Y_k + Z, \Lambda_k))(Z)\rangle + \langle \mathcal{D}\mathcal{L}(Y_k + Z, \Lambda_k), Z\rangle$. A função $\mathcal{Q}_k(Z)$ é uma aproximação quadrática para $\mathcal{L}(Y_k + Z, \Lambda_k)$, para o caso de problemas quadráticos.

Note que o problema (3.51) pode ser resolvido utilizando métodos iterativos, tais como o Lagrangiano aumentado [50] e o Gradiente Conjugado para restrições lineares [3]. Neste trabalho, empregaremos uma adaptação do método dos gradientes conjugados com restrições lineares de igualdade [3] para o problema matricial (3.51). Isto é descrito no Algoritmo 5, em que utilizaremos o Teorema (3.3) para calcular a projeção de um ponto no subespaço $\mathcal{S}(Y_k)$.

Consideraremos a matriz $P = P_m$, com P_m definida em (3.48), como a matriz projeção tem a propriedade $P_m P_m = P_m$, então

$$\beta_i = \frac{g_{i+1}^T z_{i+1}}{g_i^T z_i} = \frac{z_{i+1}^T z_{i+1}}{z_i^T z_i}. \tag{3.52}$$

Algoritmo 5: Gradiente Conjugado aplicado ao problema de minimização no subespaço tangente.

Entrada: $\epsilon > 0$;
Dado $Z_0 \in \mathcal{S}(Y_k)$;
Faça $R_0 = \nabla_x \mathcal{Q}_k(Z_0)$, $G_0 = \mathrm{P}_{\mathcal{S}(Y_k)} R_0$, $V_0 = -G_0$, $i = 0$;

Enquanto $\|V_i\| < \epsilon$

$$\alpha_i = -\frac{\text{traço}(R_i^T G_i)}{\text{traço}(V_i^T \mathcal{D}(\mathcal{D}\mathcal{Q}_k(Z_i))(V_i))};$$

$$Z_{i+1} = Z_i + \alpha_i V_i;$$

$$R_{i+1} = R_i + \alpha_i \mathcal{D}(\mathcal{D}\mathcal{Q}_k(Z_i)(V_i);$$

$$G_{i+1} = R_{i+1} - \tfrac{1}{2} Y (R_{i+1}^T Y_k + Y_k^T R_{i+1});$$

$$\beta_i = \frac{\text{traço}(R_{i+1}^T G_{i+1})}{\text{traço}(R_i^T G_i)};$$

$$V_{i+1} = -G_{i+1} + \beta_i V_i;$$

$$i = i + 1;$$

Saída: Uma aproximação Z^* da solução de (3.51).
Fim

Para encontrar o valor de α_i minimizamos $\mathcal{Q}_k$ ao longo da reta que passa por Z_i na direção V_i. Para tal, defina $\phi : \mathbb{R} \to \mathbb{R}$ por $\phi(\alpha) = \mathcal{Q}_k(Z_i + \alpha V_i)$. Utilizando a definição de α_i, obtemos

$$\phi'(\alpha_i) = \langle \mathcal{D}\mathcal{Q}_k(Z_i + \alpha_i V_i), V_i \rangle = 0. \tag{3.53}$$

Utilizando, a expansão local do gradiente de $\mathcal{Q}_k$ no ponto $Z_i + \alpha_i V_i$, temos que

$$\mathcal{D}\mathcal{Q}_k(Z_i + \alpha_i V_i) = \mathcal{D}\mathcal{Q}_k(Z_i) + \alpha_i \mathcal{D}(\mathcal{D}\mathcal{Q}_k(Z_i))(V_i) + o(\|V_i^2\|). \tag{3.54}$$

Substituindo (3.54) em (3.53)

$$\text{traço}\left(V_i^T \mathcal{D}\mathcal{Q}_k(Z_i) + \alpha_i V_i^T \mathcal{D}(\mathcal{D}\mathcal{Q}_k(Z_i))(V_i) \right) = 0$$

$$\text{traço}\left(V_i^T \mathcal{D}\mathcal{Q}_k(Z_i) \right) + \alpha_i \text{traço}\left(V_i^T \mathcal{D}(\mathcal{D}\mathcal{Q}_k(Z_i))(V_i) \right) = 0.$$

Logo,

$$\alpha_i = -\frac{\text{traço}(V_i^T \mathcal{D}\mathcal{Q}_k(Z_i))}{\text{traço}(V_i^T \mathcal{D}(\mathcal{D}\mathcal{Q}_k(Z_i))(V_i))}.$$

3.8 Redução do sistema linear

No passo de restauração para encontrar Y_k a partir de X_k, precisamos calcular a matriz $(I - \frac{t_k}{2}A_k)^{-1}$. Na maioria das aplicações, temos que $p < n/2$. Assim, no teorema a seguir, mostraremos que a dimensão do sistema pode ser reduzida a $2p \times 2p$. Para tanto, aplicamos o Teorema de Sherman-Morrison-Woodbury para calcular, de maneira eficiente, a matriz inversa.

Segundo o Teorema 3.3, temos que A_k pode ser reescrita da forma

$$A_k = U_k V_k^T, \tag{3.55}$$

em que $U_k = \begin{bmatrix} P_k D_k & Y_k \end{bmatrix} \in \mathbb{R}^{n \times 2p}$ e $V = \begin{bmatrix} Y_k & -P_k D_k \end{bmatrix} \in \mathbb{R}^{n \times 2p}$.

Teorema 3.20. *Seja $A_k \in \mathbb{R}^{n \times n}$ da forma* (3.55)*. Então $\overline{Y}_{k+1} = (I - \frac{t_k}{2}A_k)^{-1}(I + \frac{t_k}{2}A_k)Y_k$ é equivalente a*

$$\overline{Y}_{k+1} = Y_k + t_k U_k (I - \frac{t_k}{2} V_k^T U_k)^{-1} V_k^T Y_k. \tag{3.56}$$

Demonstração. Como $I - \frac{t_k}{2}A_k = I - \frac{t_k}{2}U_k V_k^T$, aplicando a fórmula de Sherman-Morrison-Woodbury [51]:

$$(A + CD^T)^{-1} = A^{-1} - A^{-1}C(I + D^T A^{-1} C)^{-1} D^T A^{-1}, \tag{3.57}$$

obtemos

$$(I - \frac{t_k}{2} U_k V_k^T)^{-1} = I + \frac{t_2}{2}(I - \frac{t_k}{2} V_k^T U_k)^{-1} V_k^T. \tag{3.58}$$

Logo,

$$\begin{aligned}
\overline{Y}_{k+1} &= (I + \frac{t_k}{2} U_k (I - \frac{t_k}{2} V_k^T U_k)^{-1} V_k^T)(I + \frac{t_k}{2} U_k V_k^T) Y_k \\
&= Y_k + \frac{t_k}{2} U_k \left(I + (I - \frac{t_k}{2} V_k^T U_k)^{-1} (I + \frac{t_k}{2} V_k^T U_k) \right) V_k^T Y_k \\
&= Y_k + t_k U_k (I - \frac{t_k}{2} V_k^T U_k)^{-1} V_k^T Y_k.
\end{aligned}$$

■

Portanto, o uso da fórmula de Sherman-Morrison-Woodbury para inverter a matriz $I - \frac{t_k}{2} A_k$ reduz o custo computacional de se obter $\overline{Y}_{k+1}$, o qual é $4np^2 + \mathcal{O}(p^3)$.

Cabe observar que, se $p \geq \frac{n}{2}$, então vale a pena resolver o sistema linear $(I - \frac{t_k}{2} A_k)\overline{Y}_{k+1} = (I + \frac{t_k}{2} A_k)Y_k$ para encontrar o ponto restaurado Y_{k+1}.

Capítulo 4

Resultados numéricos

Neste capítulo, estudaremos o desempenho do Algoritmo de Restauração Inexata não monótono, Algoritmo 3, aplicado a diferentes problemas de otimização matriciais envolvendo restrições de ortogonalidade. Primeiro, estudaremos todos os detalhes da implementação do algoritmo proposto a fim de realizar os testes computacionais. Na segunda parte, fazemos comparações do Algoritmo proposto com outros dois métodos para resolver problemas com restrições de ortogonalidade, o método de Busca Curvilinear [4](Wen e Yin), que será denotado por OptStiefel e o método de gradiente conjugado, que será denotado por Conj-Grad do pacote ManOpt [5]. A implementação do Algoritmo 3 foi realizada em MATLAB versão R2017b de 64-bits.

Todos os experimentos numéricos foram realizados em um notebook com processador Intel(R) Core(TM) i5-2520M de 2.50GHz e 4Gb de memória RAM.

4.1 Detalhes da implementação

Assumiremos como ponto inicial $X_0 = UV^T$, em que U e V são obtidas pela decomposição em valores singulares da matriz randn(n, p) e $Y_0 = X_0$. Em todos os testes numéricos utlizaremos: $\theta_0 = 0.999$.

Consideramos nos experimentos numéricos, problemas em que o número de restrições p menor que n. A fim de analisar o desempenho e aplicação do Algoritmo de Restauração Inexata não monótona foram realizados diferentes testes numéricos dos problemas: autovalor linear [4], procrustes ortogonal [9], minimização da energia total [11] e mini-

mização de formas quadráticas heterogêneas [52].

Além disso, descrevemos os critérios de parada utilizados nos testes numéricos. A execução do Algoritmo 3 é interrompida nos seguintes casos:

- O número de iterações do Algoritmo excede $k_{\max} = 1000$.
- O tamanho da derivada do Lagrangiano no ponto (Y_k, Λ_k) está próximo de zero, isto é,

$$\|\mathrm{P}_{\mathcal{S}(Y_k)}(\nabla F(Y_k))\|_F < \epsilon, \tag{4.1}$$

  O ponto Y_k que satisfaz a condição (4.1) é chamado de ϵ-AGP. Para os testes utilizamos $\epsilon = 10^{-5}$.
- O valor do comprimento do passo t_k é pequeno, ou seja, $t_k < 10^{-10}$. Note que é interessante aceitar passos $t_k = 1$, para garantir a convergência superlinear ou quadrática, conforme Teorema 2.6.
- A falta de progresso da iteração ou dos valores da função, isto é, $\|X_k - X_{k-1}\|_F < xtol$ ou $|F(X_k) - F(X_{k-1})| < ftol$. Nos experimentos utilizamos $xtol = ftol = 10^{-10}$.

A medida de eficiência a ser utilizada nos problemas testes será o tempo computacional de execução. Neste capítulo, compararemos o desempenho do algoritmo proposto com OptStiefel [4] e Conj-Grad [5]. Além disso, estudaremos os perfis de desempenho na resolução dos problemas testes para determinar alguns parâmetros do Algoritmo η e M.

Uma característica importante do método de Restauração Inexata é a liberdade de escolha das estratégias para resolver as fases de Restauração e Otimalidade. A seguir, apresentamos as abordagens para a resolução de cada fase do algoritmo.

4.1.1 Fase de restauração

Nesta fase, encontraremos a sequência $\{Y_k\} \in \mathcal{V}$ que satisfaz as condições dadas em (2.24) e (2.25).

Pelo Algoritmo 3, a sequência $\overline{Y}_k$ é definida pela transformação de Cayley

$$\overline{Y}_{k+1} = (I - \frac{t_k}{2}A_k)^{-1}(I + \frac{t_k}{2}A_k)Y_k$$

e Y_{k+1} pode ser qualquer ponto viável tal que $F(Y_{k+1}) \leq F(\overline{Y}_{k+1})$.

Como o Teorema 2.6 garante boas propriedades de convergência local com tamanho de passo $t_k = 1$, geraremos uma quantidade finita de iterações da seguinte maneira: Dado o ponto inicial $Y_{1,0} = \overline{Y}_{k+1}$ e o multiplicador de Lagrange $\Lambda_k \in \mathbb{R}^{p\times p}$, definiremos para $j = 1, \ldots, M$, com $M \in \mathbb{N}$,

$$\begin{aligned} D_{k,j} &= -\frac{1}{\lambda_{k,j}^{sp}} \mathrm{P}_{\mathcal{S}(Y_{k,j})}(\nabla F(Y_{k,j})), \\ A_{k,j} &= (I - \frac{1}{2}Y_{k,j}Y_{k,j}^T)D_{k,j}Y_{k,j}^T - Y_{k,j}D_{k,j}^T(I - \frac{1}{2}Y_{k,j}Y_{k,j}^T), \\ Y_{k,j+1} &= (I - \frac{1}{2}A_{k,j})^{-1}(I + \frac{1}{2}A_{k,j})Y_{k,j}, \end{aligned}$$

em que $\lambda_{k,j}^{sp}$ é o parâmetro espectral definido por

$$\lambda_{k,j}^{sp} = \min\{\max\{\lambda_{\min}, \lambda_{k,j}^{BB}\}, \lambda_{\max}\},$$

e

$$\lambda_{k,j}^{BB} = \begin{cases} \frac{|\text{traço}(\nabla L(Y_{k,j},\Lambda_k)) - \nabla L(Y_{k-1,j},\Lambda_{k+1}))^T(Y_{k,j}-Y_{k-1,j})|}{\|Y_{k,j}-Y_{k-1,j}\|_F^2}, & \text{se j é par} \\ \\ \frac{\|\nabla L(Y_{k,j},\Lambda_k) - \nabla L(Y_{k-1,j},\Lambda_k)\|_F^2}{|\text{traço}(\nabla L(Y_{k,j},\Lambda_k) - \nabla L(Y_{k-1,j},\Lambda_k))^T(Y_{k,j}-Y_{k-1,j})|}, & \text{se j é impar.} \end{cases}$$

A execução deste algoritmo é interrompida se

$$\|\mathrm{P}_{\mathcal{S}(Y_{k,j})}(\nabla F(Y_{k,j})\| < \epsilon.$$

Portanto, o iterando $Y_{k,j}$ é solução aproximada do problema.
Após a realização das M iterações, atualizamos $Y_{k+1} = Y_{k,j}$, em que $j \in \{1, \ldots, M\}$ é o maior índice tal que $F(Y_{k,j}) \leq F(Y_{k,1})$. A seguir apresentamos o Algoritmo 6, que corresponde à iteração interna do Passo 4 do Algoritmo 3.

Pelo Teorema 3.20, $\overline{Y}_{k+1}$ pode ser obtido resolvendo-se um sistema linear de tamanho $2p \times 2p$

$$\overline{Y}_{k+1} = Y_k + t_k U_k (I - \frac{t_k}{2} V_k^T U_k)^{-1} V_k^T Y_k.$$

Vale observar que outros algoritmos podem ser utilizados na realização desta fase. Em particular, se $\|\frac{t_k}{2}A_k\|_F < 1$, podemos utilizar o Lema de Banach [53] para aproximar a inversa $(I - \frac{t_k}{2}A_k)^{-1}$. Então, o ponto Y_k poderá ser estimado pela expressão

$$\overline{Y}_{k+1} = Y_k + 2\left(\sum_{j=1}^{m}(\frac{t_k}{2}A_k)^j\right)Y_k.$$

Algoritmo 6: Iteração local.

Entrada: $\overline{Y}_{k+1} \in \mathbb{R}^{n\times p}$ viável,
$Y_k \in \mathbb{R}^{n\times p}, \Lambda_k \in \mathbb{R}^{p\times p}, M \in \mathbb{N}, \epsilon > 0.$

Saída: $Y_{k+1} \in \mathbb{R}^{n\times p}$ viável

Faça $Y_{k,0} = Y_k, Y_{k,1} = \overline{Y}_{k+1}, j = 1;$

enquanto $j \leq M$ **faça**

  se j *par* **então**

    $\lambda_{k,j}^{BB} = \frac{|\text{traço}(\nabla L(Y_{k,j},\Lambda_k)) - \nabla L(Y_{k,j-1},\Lambda_k))^T (Y_{k,j-1} - Y_{k,j})|}{\|Y_{k,j} - Y_{k,j-1}\|_F^2}$

    senão

      $\lambda_{k,j}^{BB} = \frac{\|\nabla L(Y_{k,j},\Lambda_k) - \nabla L(Y_{k,j-1},\Lambda_k)\|_F^2}{|\text{traço}(\nabla L(Y_{k,j},\Lambda_k)) - \nabla L(Y_{k,j-1},\Lambda_k))^T (Y_{k,j-1} - Y_{k,j})|}$

    fim

  fim

  $\lambda_{k,j}^{sp} = \min\{\max\{\lambda_{\min}, \lambda_{k,j}^{BB}\}, \lambda_{\max}\};$

  $D_{k,j} = -\frac{1}{\lambda_{k,j}^{sp}} \mathrm{P}_{\mathcal{S}(Y_{k,j})}(\nabla F(Y_{k,j}));$

  $A_{k,j} = (I - \frac{1}{2} Y_{k,j} Y_{k,j}^T) D_{k,j} Y_{k,j}^T - Y_{k,j} D_{k,j}^T (I - \frac{1}{2} Y_{k,j} Y_{k,j}^T);$

  $Y_{k,j+1} = (I - \frac{1}{2} A_{k,j})^{-1} (I + \frac{1}{2} A_{k,j}) Y_{k,j};$

  se $F(Y_{k,j+1}) \leq F(Y_{k,1})$ **então**

    $Y_{\min} = Y_{k,j+1};$

  fim

  se $\|P_{\mathcal{S}(Y_{k,j})}(\nabla F(Y_{k,j}))\| < \epsilon$ **então**

    $Y_{k,j}$ é solução aproximada do problema; Pare;

  fim

  Faça $j = j + 1.$

fim

Faça $Y_{k+1} = Y_{\min}.$

4.1.2 Fase de otimização (escolha de D_k)

Neste trabalho, a escolha da direção tangente D_k do Passo 2 do Algoritmo 3 será feita de duas formas. Primeiro, se $\|\mathrm{P}_{\mathcal{S}(Y_k)}(\nabla F(Y_k))\| \geq 10^{-2}$, fazemos

$$D_k = \frac{1}{\lambda_k^{sp}} \mathrm{P}_{\mathcal{S}(Y_k)}(-\nabla F(Y_k)),$$

isto é, pelo Teorema 3.3,

$$D_k = \frac{1}{\lambda_k^{sp}}(-\nabla F(Y_k) + \frac{1}{2}Y_k(\nabla F(Y_k)^T Y_k + Y_k^T \nabla F(Y_k))), \qquad (4.2)$$

em que λ_k^{sp} é o parâmetro espectral. Neste caso,

$$\lambda_k^{sp} = \min\{\max\{\lambda_{\min}, \lambda_k^{BB}\}, \lambda_{\max}\},$$

e

$$\lambda_k^{BB} = \begin{cases} \frac{|\text{traço}(\nabla L(Y_k,\Lambda_k)) - \nabla L(Y_{k-1},\Lambda_k))^T(Y_k - Y_{k-1})|}{\|Y_k - Y_{k-1}\|_F^2}, & \text{se k é par} \\ \\ \frac{\|\nabla L(Y_k,\Lambda_k) - \nabla L(Y_{k-1},\Lambda_k)\|_F^2}{|\text{traço}(\nabla L(Y_k,\Lambda_k) - \nabla L(Y_{k-1},\Lambda_k))^T(Y_k - Y_{k-1})|}, & \text{se k é impar.} \end{cases}$$

Agora se, $\|\mathrm{P}_{\mathcal{S}(Y_k)}(\nabla F(Y_k))\| < 10^{-2}$, o candidato à D_k é uma solução do subproblema

$$\begin{aligned} \text{minimizar} \quad & \mathcal{Q}_k(Z) \\ \text{s.a.} \quad & Z \in \mathcal{S}(Y_k) \end{aligned}$$

em que $\mathcal{Q}_k(Z) = \frac{1}{2}\langle Z, \mathcal{D}(\mathcal{D}\mathcal{L}(Y_k + Z, \Lambda_k))(Z)\rangle + \langle \mathcal{D}\mathcal{L}(Y_k + Z, \Lambda_k), Z\rangle$. O Algoritmo 5 gera uma sequência $\{Z_i^k\}$ que aproxima a solução do subproblema acima. Consideraremos, o ponto inicial
$Z_0^k = \mathrm{P}_{\mathcal{S}(Y_k)}(-\nabla F(Y_k)) \in \mathcal{S}(Y_k)$ e estabelecemos os seguintes critérios de parada utilizados no Algoritmo 5:

1. O tamanho $\|Z_i^k\|_F$ é menor que uma tolerância 10^{-4}.

2. Uma direção de curvatura negativa ou nula é encontrada, isto é,

$$(Z_i^k)^T \mathcal{D}(\mathcal{D}\mathcal{Q}_k(Z_i^k))Z_i^k < \epsilon_c,$$

   em que $\epsilon_c = 10^{-10}$.

3. A quantidade de iterações do método dos gradientes conjugados excede 50 iterações.

4. O tamanho de Z_i^k excede a região de confiança, $\|Z_i^k\|_F > \Delta$, em que $\Delta := max\{\|\nabla \mathcal{Q}_k Z_0^k\|_F, \|Z_0^k\|_F, 10\sqrt{p}\}$ como ilustrado na Figura 4.1.

Após o cálculo da direção D_k, verificaremos se D_k satisfaz as duas condições:

$$\|D_k\|_F \geq \bar{\mu}\|\mathrm{P}_{\mathcal{S}(Y_k)}(\nabla F(Y_k))\|_F \quad e \quad \langle \nabla F(Y_k), D_k\rangle \leq -\mu\|D_k\|_F^2.$$

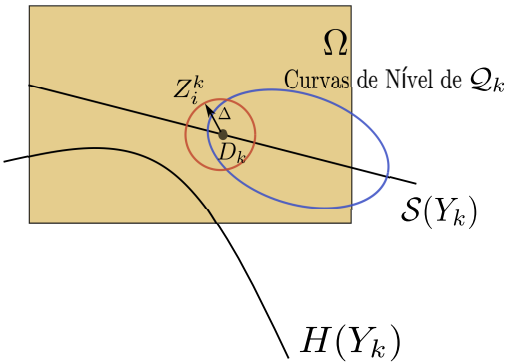

Figura 4.1: Região de confiança de raio Δ e centro D_k.

Se não satisfaz, escolhemos

$$D_k = \frac{1}{\lambda_k^{sp}} \mathrm{P}_{\mathcal{S}(Y_k)}(-\nabla F(Y_k)).$$

Vamos mostrar, na sequência, que com estas escolhas, nossa implementação é um caso particular do Algoritmo 3.

Teorema 4.1. *Assuma que a hipótese (**H1**) é satisfeita. Então, existem $\gamma, \bar{\gamma}, \tau > 0$ tais que*

$$F(Y_k + tD_k) \leq F(Y_k) - \gamma t \|D_k\|_F^2,$$
$$h(Y_k + tD_k) \leq h(Y_k) + \bar{\gamma} t^2 \|D_k\|_F^2,$$

para todo $Y_k \in \Omega$ e $t \in [0, \tau]$.

Demonstração. Pelo Teorema fundamental do Cálculo Integral,

$$\begin{aligned} F(Y_k + tD_k) =& F(Y_k) + t\langle \nabla F(Y_k), D_k \rangle \\ &+ t \int_0^1 \langle \nabla F(Y_k + tsD_k) - \nabla F(Y_k), D_k \rangle \, s\mathrm{ds}, \end{aligned} \tag{4.3}$$

$\forall t \in [0, 1]$.
Utilizando o fato que ∇F é Lipschitz contínua em Ω, Y_k, $Y_k + D_k \in \Omega$ e pela convexidade de Ω, temos que

$$F(Y_k + tD_k) \leq F(Y_k) + t\langle \nabla F(Y_k), D_k \rangle + \frac{1}{2} t^2 L \|D_k\|_F^2, \quad \forall t \in [0, 1].$$

Logo, pela condição (3.23)

$$\begin{aligned} F(Y_k + tD_k) &\leq F(Y_k) - \mu t\|D_k\|_F^2 + \frac{1}{2}t^2 L\|D_k\|_F^2 \\ &= F(Y_k) - t(\mu - \frac{1}{2}tL)\|D_k\|_F^2. \end{aligned}$$

Portanto, se $t \leq \tau := \min\{1, \frac{\mu}{2L}\}$, então

$$F(Y_k + tD_k) \leq F(Y_k) - \gamma t\|D_k\|_F^2,$$

com $\gamma = \mu/2$.
Aplicando o Teorema Fundamental do Cálculo a H e se $D_k \in \mathcal{S}(Y_k)$, isto é, a derivada de H no ponto Y_k, e na direção D_k, é zero, $\nabla H(Y_k)D_k = 0$, temos que

$$H(Y_k+tD_k) = H(Y_k)+t\int_0^1 (\nabla H(Y_k + tsD_k) - \nabla H(Y_k))D_k \,\mathrm{ds}, \quad (4.4)$$

para todo $t \in [0, 1]$.
Tomando a norma em ambos os lados da igualdade (4.4) e pela hipótese ∇H ser Lipschitz contínua, segue que

$$\|H(Y_k + tD_k)\|_F \leq \|H(Y_k)\|_F + \frac{1}{2}Lt^2\|D_k\|_F^2$$

para todo $t \in [0, 1]$. Portanto, a desigualdade seguinte

$$\|H(Y_k + tD_k)\|_F \leq \|H(Y_k)\|_F + \bar{\gamma}t^2\|D_k\|_F^2.$$

vale para todo $t \in [0, \tau] \subseteq [0, 1]$ com $\bar{\gamma} := \frac{1}{2L}$. ■

4.2 Escolha dos parâmetros de não monotonía η e iterações locais M

A fim de escolher os parâmetros η_k e M, analisamos o perfil de desempenho do Algoritmo 3. Para tanto, foram resolvidos 39 e 36 problemas respectivamente. As instâncias para analisar η_k estão nas Tabelas 12 a 15, e para analisar M, nas Tabelas 16 a 19.
Os resultados encontram-se resumidos nas Figuras 4.2 e 4.3 respectivamente, utilizando o perfil de desempenho introduzido por Dolan e Moré [54]. Dado um conjunto de problemas P e um conjunto de métodos de otimização S, comparamos o desempenho do problema $p \in P$ por um

algoritmo em particular $s \in S$, com o melhor desempenho realizado por outro método para este problema. Denotamos $t_{p,s}$ a quantidade de avaliações funcionais requeridas para resolver o problema $p \in P$ usando o método $s \in S$, define-se o coeficiente de desempenho da seguinte forma

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : s \in S\}},$$

e assuma que $r_{p,s}$ pertence ao intervalo $[1, r_M]$. O valor $r_{p,s}$ será igual r_M quando o problema p não foi resolvido usando o método s. A seguir definimos o perfil de desempenho do método s, pela fração

$$\rho_s(\tau) = \frac{1}{n_p}\text{size}\{p \in P : r_{p,s} \leq \tau\}, \tag{4.5}$$

em que n_p denota a quantidade total de problemas resolvidos. O perfil de desempenho é uma função que associa a um valor $\tau \in \mathbb{R}^+$, a fração de elementos resolvidos pelo método s com um desempenho dentro de um fator τ do melhor desempenho obtido. Devido a que r_M pode ser muito maior do que 1, será utilizada escala logarítmica para a apresentação dos perfis de desempenho, com o seguinte mapeamento:

$$\tau \to \frac{1}{n_p}\text{size}\{p \in P : \log_2(r_{p,s}) \leq \tau\} \tag{4.6}$$

Na figura do perfil de desempenho, a curva superior representa o método mais eficiente.

Primeiro, estudamos o desempenho para diferentes valores η_k com parâmetro $M = 0$ fixo para cada problema. A Figura 4.2 ilustra o perfil de desempenho para diferentes valores $\eta_k \in \{0,\ 0.25,\ 0.5,\ 0.85,\ 0.99\}$ no intervalo [0.1361, 1]. A Figura 4.2 mostra que $\eta_k = 0.85$ resolve, com o menor tempo, 74% problemas testados. As tabelas associadas aos perfis de desempenho estão organizados no Apêndice. Ao final, no apêndice encontram-se as tabelas com a informação obtida para os problemas testes: autovalor linear, Procrustes ortogonal, minimização total de energia e minimização de formas quadráticas.

O perfil de desempenho correspondente à execução do Algoritmo 3 para diferentes valores das iterações locais $M \in \{0,\ 10,\ 15,\ 20,\ 50,\ 100\}$ e $\eta_k = 0.85$ são apresentados na Figura 4.3. Observe que, o valor $M = 15$ obtém um bom desempenho, pois 80% dos problemas testados foram resolvidos em tempo significativamente menor. Os detalhes dos resultados obtidos encontram-se no apêndice.

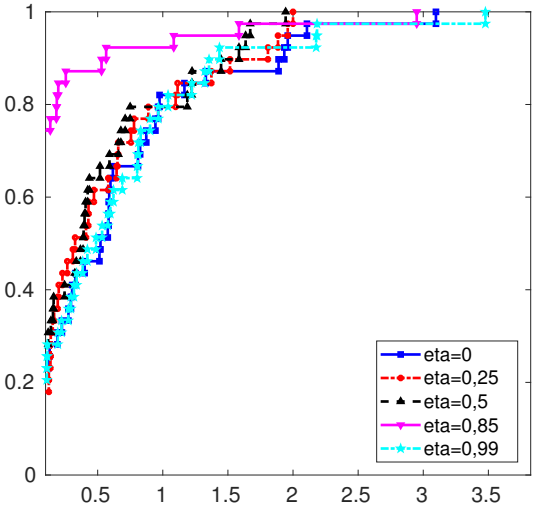

Figura 4.2: Perfil de desempenho para diferentes valores η_k.

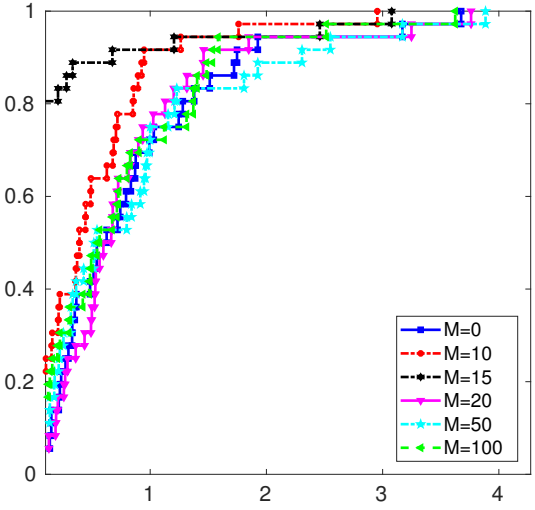

Figura 4.3: Perfil de desempenho para diferentes valores de M

4.3 Problemas testes

4.3.1 Problema de autovalor linear

Dada uma matriz simétrica $A \in \mathbb{R}^{n\times n}$ e $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ seus autovalores, o problema de encontrar uma base ortonormal para o subespaço invariante associado aos seus p maiores autovalores pode ser escrito como

$$\begin{array}{ll} \text{maximizar} & \frac{1}{2}\text{traço}(X^T AX) \\ \text{s.a.} & X^T X = I,\ X \in \mathbb{R}^{n\times p},\ n \geq p. \end{array} \tag{4.7}$$

Neste caso, o valor da função objetivo no ótimo global é $\sum_{i=1}^{p} \lambda_i$. O cálculo do subespaço invariante principal de uma matriz simétrica tem muitas aplicações em problemas, tais como: análise de estabilidade de sistemas dinâmicos, partição de grafos, entre outros.
Note que o problema (4.7) é equivalente a minimizar $-\text{traço}(X^T AX)$. Aqui, definimos a função objetivo

$$F(X) := -\text{traço}(X^T AX)$$

e $\mathcal{V} := \{X \in \mathbb{R}^{n\times p} : X^T X = I\}$. Considere o problema

$$\begin{array}{ll} \text{minimizar} & F(X) \\ \text{s.a.} & X \in \mathcal{V}. \end{array} \tag{4.8}$$

O gradiente de F de é da forma $\nabla F(X) = -AX$. Assim, as condições KKT do Problema (4.8) são

$$\begin{aligned} AX - X\Lambda &= 0 \\ X^T X &= I, \end{aligned}$$

em que $\Lambda \in \mathbb{R}^{p\times p}$ é a matriz formada pelos multiplicadores de Lagrange do Problema (4.8).
Considere a seguinte notação: "Feval" o valor da função objetivo na solução encontrada, "Feasi" a medida da viabilidade, "normG" a medida da otimalidade $\|\mathrm{P}_{\mathcal{S}(Y_k)}(\nabla F(Y_k))\|_F$, "itr" o número de iterações, "nfeval" o número de avaliações da função objetivo, "CPUtime" o tempo em segundos. O "Erro" denota o erro relativo entre os valores da função objetivo, dados pelo comando `eigs` do MATLAB, e os valores obtidos por cada algoritmo, isto é,

$$Erro = \frac{\frac{1}{2}\sum_{i=1}^{p} \lambda_i - \frac{1}{2}\text{traço}(X_k^T AX_k)}{\frac{1}{2}\sum_{i=1}^{p} \lambda_i},$$

em que X_k é a solução aproximada calculada pelos Algoritmos.
Além disso, nas tabelas, o valor entre parênteses indica o total de iterações do Algoritmo 5, dado pela soma do número de iteraçõs do Gradiente Conjugado para encontrar a direção de descida.

Exemplo 1: Os elementos da diagonal são gerados através de uma distribução normal no intervalo $[10, 12]$.

A Tabela 4.1 apresenta o desempenho dos métodos para o Exemplo 1 com $n \in \{250, 500, 1000, 2000\}$ e $p \in \{5, 20, 20, 40\}$. Podemos notar que tanto o Algoritmo 3 quanto OptStiefel foram mais eficientes em termos do tempo CPU, do que Conj-Grad. Além disso, da Tabela 4.1 percebemos que o Algoritmo 3 converge em poucas iterações para todos os casos. Também a medida que aumentamos a dimensão de n e p o tempo de execução dos algoritmos é maior.
A Figura 4.4 à esquerda mostra o comportamento do erro relativo da função objetivo para o Exemplo 1 com $n = 250$, $p = 5$. A figura à direita ilustra o comportamento da condição de otimalidade.

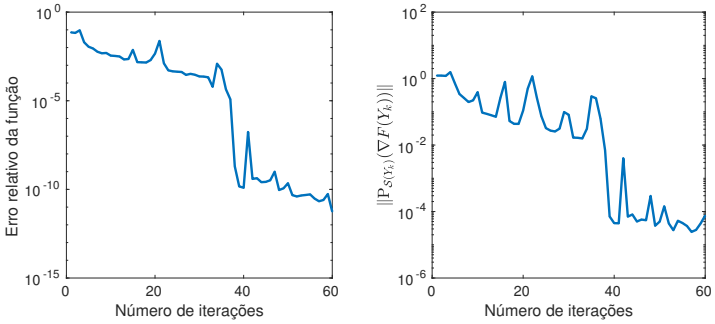

Figura 4.4: Comportamento do Algoritmo 3 para o Exemplo 1 com $n = 250$, $p = 5$.

Exemplo 2: A matriz A é da forma $A = \bar{A}^T \bar{A}$, em que $\bar{A} \in \mathbb{R}^{n \times n}$ é uma matriz randômica com distribução normal.

A Tabela 4.2 expõe os resultados alcançados para o Exemplo 2 com $n \in \{500, 1000, 2000, 4000\}$ e $p = 50$. Notemos que, o Algoritmo 3 e OptStiefel tiveram melhor desempenho computacional do que Conj-Grad. Além disso, para $n = 500$ e $n = 2000$, o Algoritmo 3 atingiu

$n = 250$, $p = 5$							
Algoritmo	Feval	Feasi	normG	itr	nfeval	CPUtime	Erro
Algoritmo 3	5.993e+01	2.817e-12	4.646e-06	60	61(86)	0.048	5.498e-12
OptStiefel	5.993e+01	2.008e-16	4.856e-06	80	84	0.015	1.176e-11
Conj-Grad	5.993e+01	3.130e-15	9.788e-06	71	120	0.127	1.400e-11
$n = 500$, $p = 10$							
Algoritmo 3	1.198e+02	7.982e-12	9.754e-06	78	79(127)	0.108	3.853e-11
OptStiefel	1.198e+02	5.251e-16	9.433e-06	138	148	0.057	7.137e-11
Conj-Grad	1.198e+02	7.585e-15	7.771e-06	133	219	0.241	6.107e-12
$n = 1000$, $p = 20$							
Algoritmo 3	2.397e+02	4.903e-13	9.775e-06	127	128(379)	0.930	3.095e-11
OptStiefel	2.397e+02	9.657e-16	9.953e-06	320	341	0.493	7.938e-11
Conj-Grad	2.397e+02	1.723e-14	9.331e-06	283	485	1.338	4.112e-11
$n = 2000$, $p = 40$							
Algoritmo 3	4.793e+02	4.657e-11	5.443e-05	361	363(1167)	8.551	2.429e-10
OptStiefel	4.793e+02	1.744e-15	1.198e-04	1000	1060	6.159	3.550e-16
Conj-Grad	4.793e+02	1.455e-14	9.198e-06	429	727	5.799	3.070e-10

Tabela 4.1: Resultados numéricos para o Exemplo 1 do Problema de Autovalor Linear.

melhor tempo de execução. Para os dois últimos valores de n, o Algoritmo 3 não realizou iterações do gradiente conjugado.

Os gráficos das figuras 4.5 mostram o comportamento dos três métodos para o Problema de Autovalor Linear com $n = 500$ e $p = 50$. O gráfico inferior à direita é um resumo dos gráficos anteriores. Note que, o Algoritmo 3 apresentou um menor número de iterações para calcular a solução aproximada.

A Tabela 4.3 apresenta os resultados para o Exemplo 2 com $n = 5000$ e $p \in \{10, 40, 70, 100\}$. Nestes problemas, o Algoritmo 3 e OptStiefel apresentaram um desempenho similar. Repare que, com $p = 40$ o Algoritmo 3 utilizou menor tempo de CPU. Além disso, o Algoritmo 3 apresentou menos avaliações da função objetivo.

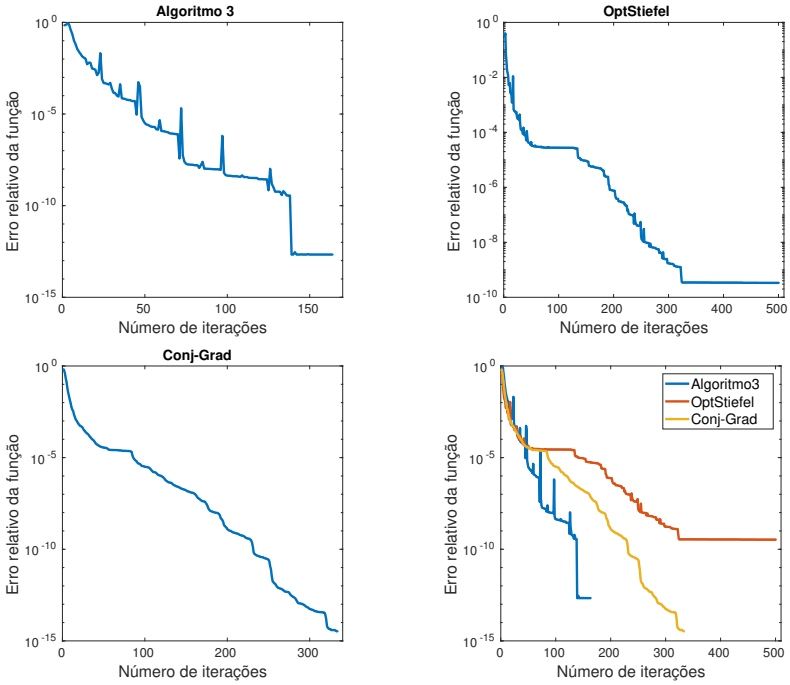

Figura 4.5: Comportamento dos Algoritmos para o Exemplo 2 do Problema de Autovalor Linear com $n = 500$, $p = 50$.

$n = 500$							
Algoritmo	Feval	Feasi	normG	itr	nfeval	CPUtime	Erro
Algoritmo 3	7.816e+04	2.929e-11	8.156e-06	164	165(104)	1.368	2.154e-13
OptStiefel	7.816e+04	1.701e-15	3.038e-05	501	545	1.636	1.852e-16
Conj-Grad	7.816e+04	7.353e-15	2.780e-05	333	579	3.748	3.333e-15
$n = 1000$							
Algoritmo 3	1.716e+05	1.497e-11	7.271e-06	222	223(83)	3.903	3.813e-13
OptStiefel	1.716e+05	1.440e-15	1.920e-05	1326	344	2.440	8.481e-16
Conj-Grad	1.716e+05	8.342e-15	1.246e-04	290	505	7.777	3.562e-15
$n = 2000$							
Algoritmo 3	3.656e+05	2.122e-12	1.087e-02	179	181(0)	7.127	1.946e-11
OptStiefel	3.656e+05	1.484e-15	1.727e-05	309	332	7.273	4.776e-16
Conj-Grad	3.656e+05	8.358e-15	5.963e-04	223	386	20.156	1.720e-14
$n = 4000$							
Algoritmo 3	7.558e+05	1.105e-11	6.480e-02	225	227(0)	22.666	1.042e-10
OptStiefel	7.558e+05	1.349e-15	1.975e-04	248	256	19.723	1.386e-15
Conj-Grad	7.558e+05	8.870e-15	6.649e-04	248	434	77.377	4.313e-15

Tabela 4.2: Resultados numéricos para o Exemplo 2 do Problema de Autovalor Linear com $p = 50$.

$p = 10$							
Algoritmo	Feval	Feasi	normG	itr	nfeval	CPUtime	Erro
Algoritmo 3	1.965e+05	6.495e-13	1.179e-02	191	193(0)	15.536	2.298e-11
OptStiefel	1.965e+05	7.142e-16	4.764e-04	294	316	12.471	3.791e-14
Conj-Grad	1.965e+05	3.114e-15	1.894e-04	247	423	63.688	8.886e-16
$p = 40$							
Algoritmo 3	7.662e+05	3.036e-11	1.138e-01	171	173(0)	28.309	2.592e-10
OptStiefel	7.662e+05	1.157e-15	2.336e-04	315	346	28.955	3.191e-15
Conj-Grad	7.662e+05	8.083e-15	1.406e-03	266	459	111.016	1.611e-14
$p = 70$							
Algoritmo 3	1.315e+06	2.735e-12	2.011e-02	260	262(0)	80.336	6.956e-12
OptStiefel	1.314e+06	2.137e-15	4.760e-04	408	439	68.588	1.240e-14
Conj-Grad	1.314e+06	1.192e-14	1.902e-03	312	545	206.573	8.058e-14
$p = 100$							
Algoritmo 3	1.847e+06	8.140e-14	4.538e-05	370	372(76)	185.940	6.303e-16
OptStiefel	1.847e+06	2.612e-15	6.182e-04	498	527	122.940	8.825e-16
Conj-Grad	1.847e+06	1.589e-14	1.945e-03	451	770	393.865	8.988e-14

Tabela 4.3: Resultados numéricos para o Exemplo 2 do Problema de Autovalor Linear com $n = 5000$.

4.3.2 Problema de procrustes ortogonal

Seja $X \in \mathbb{R}^{n\times p}$, $n \geq p$. Dadas as matrizes $A \in \mathbb{R}^{m\times n}$ e $B \in \mathbb{R}^{m\times p}$, o problema de Procrustes Ortogonal (OPP) consiste em resolver

$$\begin{aligned} \text{minimizar} \quad & \frac{1}{2}\|AX - B\|_F^2 \\ \text{s.a.} \quad & X^T X = I,\ X \in \mathbb{R}^{n\times p},\ n \geq p, \end{aligned} \tag{4.9}$$

O problema OPP tem varias aplicações em diversas áreas tais como: análise [44], psicometria [55], sistema de posicionamento global(GPS) [56] entre outros.

O lema seguinte garante a forma fechada para o problema (4.9) para o caso $m = n$.

Lema 4.2. *Seja $A = U\Sigma V^T$ a decomposição em valores singulares reduzida de $A \in \mathbb{R}^{n\times n}$. Então $X^* = UV^T$ é solução de*

$$\begin{aligned} \textit{minimizar} \quad & \|X - A\|_F^2 \\ & X^T X = I,\ X \in \mathbb{R}^{n\times p}. \end{aligned}$$

Demonstração. A prova do lema encontra-se em [6](Seção 12.4.1). ■

Portanto, pelo Lema 4.2 a solução do problema (4.9) é da forma: $X = U_1 V^T$, em que $A^T B = U\Sigma V^T$ é a decomposição em valores singulares de $A^T B$ e $U = \begin{bmatrix} U_1 & U_2 \end{bmatrix}$.

Para os experimentos numéricos consideramos $p = q$, $m = n$, a matriz A é da forma $A = PSR^T$, em que P e R são matrizes randômicas ortogonais com distribução normal e S uma matriz diagonal definida para cada tipo de problema. A fim de estudar o comportamento da solução aproximada X_k com respeito a solução exata, escolheremos a solução X^* e tomaremos a matriz $B = AX^*$, em que X^* é uma matriz com colunas ortogonais geradas randomicamente. Os problemas testes foram retirados de [57] e são descritos a seguir.

Exemplo 1: Os elementos da diagonal principal de S são randômicos e uniformemente distribuídos no intervalo $[10, 12]$, sendo um problema bem condicionado.

Denotamos o resíduo $\frac{1}{2}\|AX_k - B\|_F^2$ por Res e a norma do erro na solução $\|X_k - X^*\|_F$ por Erro. Observe que, para este problema, a matriz A é bem condicionada pois tem valores singulares muito próximos. Da Tabela 4.4 podemos chegar às seguintes conclusões: o Algoritmo 3 exibe um bom desempenho para todos os casos. Além disso, nosso Algoritmo realizou pouca iterações e avaliações da função. Também podemos observar que o resíduo e o erro decaem à medida que aumenta o número de iterações, como é mostrado na Figura 4.6 para o problema com $n = 500$ e $p = 50$.

Exemplo 2: Cada entrada da diagonal de S é dada por $S_{ii} = 1 + \frac{99(i-1)}{n-1} + 2r_i$, com r_i distribuídos uniformemente no intervalo $[0, 1]$.

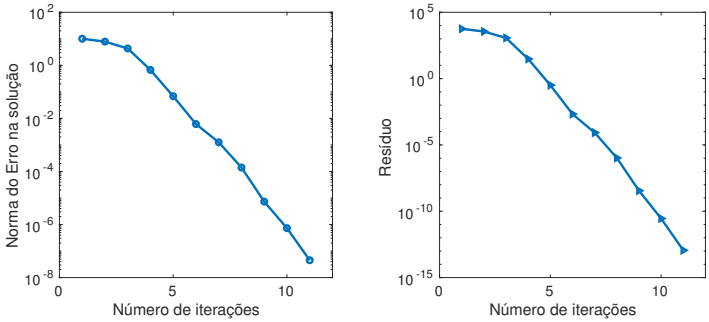

Figura 4.6: Comportamento do Algoritmo 3 para o Exemplo 1 com $n = 500$, $p = 50$.

A Tabela 4.5 mostra que o desempenho numérico do Algoritmo 3 foi melhor do que os métodos OptStiefel e Conj-Grad com $n = 100$ e $p \in \{10, 30, 50, 70\}$. Cabe ressaltar que, para os valores $p = 30$ e $p = 70$ nosso Algoritmo alcançou o valor aproximado com 333 e 372 iterações respectivamente, porém o método OptStiefel atingiu o número máximo de iterações.

Exemplo 3: A matriz S é definida por
$S = \texttt{diag}([10{*}\texttt{ones}(1, m_1){+}\texttt{randn}(1, m_1), 5{*}\texttt{ones}(1, m_2){+}\texttt{rand}(1, m_2), 2 * \texttt{ones}(1, m_3) + \texttt{rand}(1, m_3), \texttt{rand}(1, m_4)/1000])$, em que $m_1 + m_2 + m_3 + m_4 = n$.
Vamos considerar os seguintes valores para m_1, m_2, m_3, m_4:

$n = 500$							
Algoritmo	Res	Feasi	normG	itr	nfeval	CPUtime	Erro
Algoritmo 3	1.143e-13	2.184e-14	5.101e-06	11	12(4)	0.123	4.505e-08
OptStiefel	1.143e-13	2.587e-14	2.605e-06	16	21	0.147	1.822e-08
Conj-Grad	8.978e-14	2.619e-15	4.290e-06	24	35	0.210	4.186e-08
$n = 1000$							
Algoritmo 3	1.220e-13	2.216e-14	5.296e-06	11	12(4)	0.251	4.631e-08
OptStiefel	4.981e-15	2.453e-14	1.107e-06	16	20	0.266	9.696e-09
Conj-Grad	9.938e-14	3.075e-15	4.499e-06	24	35	0.531	4.419e-08
$n = 2000$							
Algoritmo 3	1.099e-13	3.043e-14	5.029e-06	11	12(4)	0.798	4.394e-08
OptStiefel	2.115e-13	2.465e-14	6.912e-06	16	19	0.841	6.181e-08
Conj-Grad	1.015e-13	3.326e-15	4.551e-06	24	35	1.668	4.460e-08
$n = 3000$							
Algoritmo 3	1.488e-13	2.063e-14	5.799e-06	11	12(4)	1.770	5.155e-08
OptStiefel	1.087e-13	2.107e-14	4.736e-06	16	19	1.892	4.594e-08
Conj-Grad	1.008e-13	3.142e-15	4.531e-06	24	35	3.635	4.451e-08
$n = 4000$							
Algoritmo 3	1.400e-13	1.925e-14	5.639e-06	11	12(4)	2.970	4.990e-08
OptStiefel	2.291e-13	2.166e-14	8.343e-06	15	18	3.048	6.664e-08
Conj-Grad	1.046e-13	3.277e-15	4.615e-06	24	35	6.328	4.533e-08

Tabela 4.4: Resultados numéricos para o Exemplo 1 do Problema de Procrustes Ortogonal com $p = 50$.

- $m_1 = m_2 = 15$, $m_3 = 12$, $m_4 = 8$
- $m_1 = m_2 = m_3 = 30$, $m_4 = 5$
- $m_1 = m_2 = m_3 = 160$, $m_4 = 20$.

Observe que, para este problema, a matriz A possui valores singulares muito pequenos quando comparados com o maior valor singular. Nos problemas testados, o Algoritmo 3 foi o mais eficiente, como podemos observar na Tabela 4.6. Para a metade dos problemas, o método OptStiefel não encontrou a solução aproximada no número máximo de iterações estabelecidas.
Podemos observar na Figura 4.7, que o resíduo obtido pelo Algoritmo 3 decresce mais rápido do que para os outros métodos.

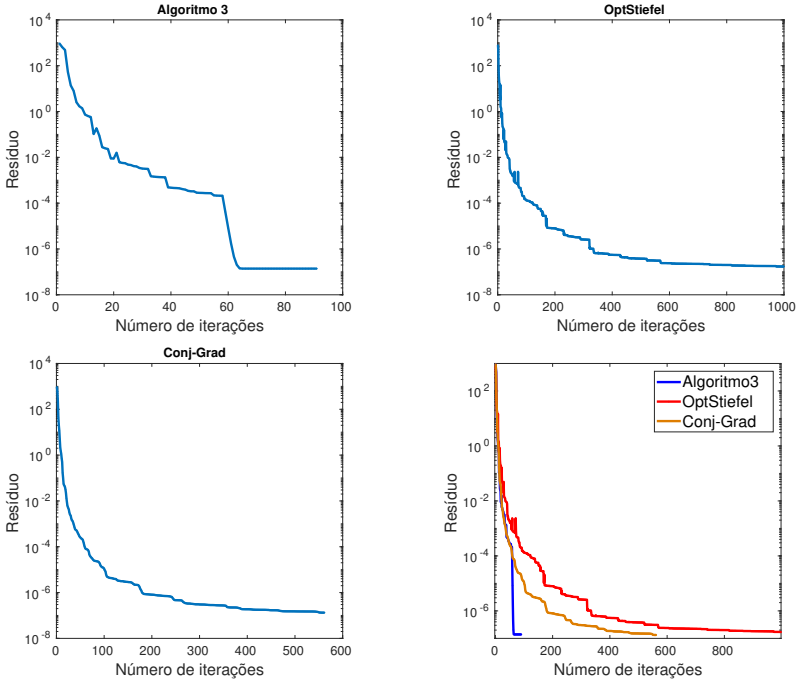

Figura 4.7: Comportamento dos Algoritmos para o problema Procrustes Ortogonal, Exemplo 3 com $n = 50$, $p = 15$.

Exemplo 4: A matriz S tem valores singulares agrupados e é definida por

			$p = 10$				
Algoritmo 3	2.575e-13	5.545e-14	6.720e-06	339	340(217)	0.183	1.488e-07
OptStiefel	1.319e+01	7.376e-14	7.709e-06	791	846	0.173	1.958e+00
Conj-Grad	1.319e+01	6.028e-16	9.906e-06	922	1576	1.180	1.958e+00
			$p = 30$				
Algoritmo	Res	Feasi	normG	itr	nfeval	CPUtime	Erro
Algoritmo 3	3.545e-13	2.048e-14	5.442e-06	333	334(223)	0.349	2.333e-07
OptStiefel	1.062e+01	9.448e-16	3.464e-02	1000	1059	0.514	1.970e+00
Conj-Grad	2.051e-12	1.988e-15	8.498e-06	549	921	1.000	8.417e-07
			$p = 50$				
Algoritmo 3	3.725e-13	3.624e-14	8.822e-06	420	421(415)	0.783	2.755e-07
OptStiefel	3.063e+00	1.278e-15	2.031e-05	845	888	0.819	1.999e+00
Conj-Grad	4.883e-12	2.978e-15	9.954e-06	567	971	1.263	1.622e-06
			$p = 70$				
Algoritmo 3	7.197e-13	6.067e-14	9.444e-06	372	373(192)	0.889	3.810e-07
OptStiefel	1.132e+01	1.683e-15	2.529e-02	1000	1030	1.047	1.961e+00
Conj-Grad	2.560e-12	3.706e-15	9.181e-06	536	901	1.5840	8.873e-07

Tabela 4.5: Resultados numéricos para o Exemplo 2 do Problema de Procrustes Ortogonal com $n = 100$.

$n = 50, p = 5\,(m_1 = m_2 = 15, m_3 = 12,\ m_4 = 8)$							
Algoritmo	Res	Feasi	normG	itr	nfeval	CPUtime	Erro
Algoritmo 3	8.751e-09	4.979e-15	9.766e-06	110	111(195)	0.109	3.551e-01
OptStiefel	5.494e-08	3.881e-15	1.076e-05	1000	1021	0.159	4.297e-01
Conj-Grad	3.718e-08	8.388e-16	5.963e-06	502	871	0.654	4.108e-01
$n = 50, p = 15\,(m_1 = m_2 = 15, m_3 = 12,\ m_4 = 8)$							
Algoritmo 3	1.389e-07	1.086e-14	6.958e-06	91	92(95)	0.111	7.688e-01
OptStiefel	1.705e-07	1.300e-14	1.757e-05	1000	1019	0.257	9.030e-01
Conj-Grad	1.331e-07	1.664e-15	9.712e-06	630	948	0.743	9.046e-01
$n = 95, p = 5\,(m_1 = m_2 = m_3 = 30,\ m_4 = 5)$							
Algoritmo 3	3.525e-09	4.465e-15	9.974e-06	182	183(512)	0.212	3.446e-01
OptStiefel	2.164e-08	4.493e-15	9.631e-06	862	893	0.233	3.118e-01
Conj-Grad	6.545e-09	8.748e-16	9.880e-06	695	695	0.877	3.192e-01
$n = 95, p = 10\,(m_1 = m_2 = m_3 = 30,\ m_4 = 5)$							
Algoritmo 3	5.675e-09	7.377e-15	9.342e-06	149	150(290)	0.177	1.752e-01
OptStiefel	2.651e-08	7.650e-15	9.170e-06	999	1019	0.364	1.803e-01
Conj-Grad	2.355e-08	7.840e-16	9.937e-06	552	944	0.765	1.931e-01
$n = 500, p = 10\,(m_1 = m_2 = m_3 = 160, m_4 = 20)$							
Algoritmo 3	4.208e-09	5.655e-15	8.965e-06	104	105(519)	1.099	1.389e-01
OptStiefel	6.381e-08	7.732e-15	1.797e-05	1000	1032	1.399	1.380e-01
Conj-Grad	1.859e-08	1.323e-15	9.608e-06	742	1294	2.588	1.373e-01

Tabela 4.6: Resultados numéricos para o Exemplo 3 do Problema de Procrustes Ortogonal.

$S = \texttt{diag}([10 + \texttt{rand}(1, n_1), 5 + \texttt{rand}(1, n_2), 2 + \texttt{rand}(1, n_3), \texttt{rand}(1, n_4)/10000])$, em que $n_1 + n_2 + n_3 + n_4 = n$. Vamos considerar n_1, n_2, n_3, n_4 da seguinte forma:

- $n_1 = \texttt{floor}((3/10) * n)$
- $n_2 = \texttt{floor}((n - n1)/3)$
- $n_3 = \texttt{floor}((n - n1 - n2)/2)$
- $n_4 = n - (n_1 + n_2 + n_3)$,

em que $\texttt{floor}(x)$ arredonda o valor de x para o inteiro mais próximo inferiormente.

Note que, para este problema a matriz A tem valores singulares agrupados conforme mostrado na Figura 4.8. Na mesma figura expomos o comportamento dos resíduos gerados pelos três métodos. Novamente o Algoritmo 3 mostrou-se eficaz como pode ser observado nos dados da Tabela 4.7. Na mesma tabela, notamos que o método OptStiefel alcançou o número máximo de iterações para todos os problemas testados.

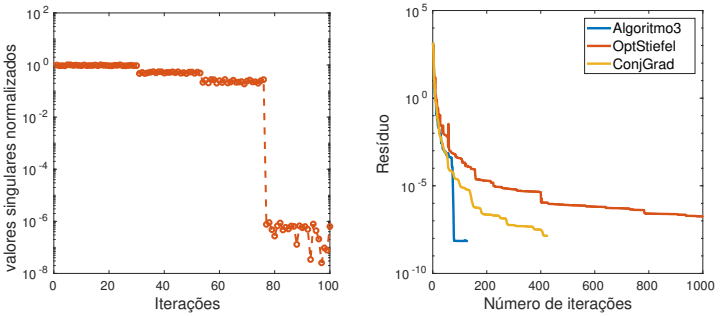

Figura 4.8: Valores singulares de A e comportamento dos Algoritmos para o problema Procrustes, Exemplo 4 com $n = 100$, $p = 20$.

$p = 10$							
Algoritmo	Res	Feasi	normG	itr	nfeval	CPUtime	Erro
Algoritmo 3	1.645e-09	8.039e-15	9.065e-06	114	115(261)	0.253	6.316e-01
OptStiefel	1.505e-07	8.823e-15	3.543e-05	1000	1037	0.345	5.822e-01
Conj-Grad	1.769e-08	1.192e-15	9.105e-06	449	768	0.838	5.905e-01
$p = 20$							
Algoritmo 3	7.074e-09	1.277e-14	8.446e-06	128	129(192)	0.165	7.825e-01
OptStiefel	1.796e-07	1.683e-14	3.257e-05	1000	1030	0.395	7.713e-01
Conj-Grad	1.402e-08	1.900e-15	8.914e-06	424	719	0.697	7.609e-01
$p = 40$							
Algoritmo 3	9.579e-10	2.642e-14	9.709e-06	118	119(233)	0.324	1.323e+00
OptStiefel	5.682e-07	3.885e-14	6.511e-05	1000	1009	0.712	1.251e+00
Conj-Grad	3.226e-08	2.926e-15	9.368e-06	786	1333	1.549	1.255e+00
$p = 80$							
Algoritmo 3	5.326e-10	7.872e-14	5.542e-06	158	159(229)	0.815	3.469e+00
OptStiefel	3.342e-06	1.757e-15	1.538e-04	1000	1020	1.020	3.372e+00
Manopt	7.633e-08	4.811e-15	9.718e-06	983	1639	2.952	3.367e+00

Tabela 4.7: Resultados numéricos para o Exemplo 4 do Problema de Procrustes Ortogonal com $n = 100$.

4.3.3 Problema de minimização da energia total (Autovalor não linear)

Nesta subseção, estamos interessados em resolver um tipo de problema de Autovalor não linear definido por:

$$\begin{aligned} \text{minimizar} \quad & E(X) := \frac{1}{2}\text{traço}(X^T L X) + \frac{\alpha}{4}\rho(X)^T L^\dagger \rho(X) \\ \text{s.a.} \quad & X^T X = I, \end{aligned} \tag{4.10}$$

em que L denota o operador Laplaciano discreto, α uma constante positiva, $\rho(X) := \text{diag}(XX^T)$ a densidade de carga e $L^\dagger$ a matriz inversa generalizada de Moore-Penrose. O problema (4.10) é uma forma reduzida da minimização total da energia da equação de Hartree-Fock e de Kohn-Sham, que tem aplicacões no cálculo de estruturas eletrônicas [4, 58].
Neste caso, o gradiente da função $E(X)$ é

$$\nabla E(X) = LX + \alpha \text{diag}(L^\dagger \rho(X))X.$$

As condições necessárias de primeira ordem do problema (4.10) podem ser escritas como:

$$\begin{aligned} H(X)X - X\Lambda &= 0 \\ X^T X &= I, \end{aligned}$$

em que $H(X) = L + \alpha \text{diag}(L^\dagger \rho(X))$ e Λ é uma matriz diagonal cujos elementos são os menores autovalores da matriz simétrica $H(X)$.
A seguir, testaremos o problema (4.10) para valores diferentes de n, α e k como sugerido em [47].
Consideramos a matriz L uma matriz tridiagonal simétrica com valores: a diagonal principal igual a 2 e as subs diagonais inferior e superior iguais com valor igual -1. O ponto inicial $X_0 = \texttt{eigs}(H(\hat{X}), k, 'sm')$, em que $\hat{X}$ é uma matriz randômica com distribução normal tal que $\hat{X} \in \mathcal{V}$.

Exemplo 1:

(a) $n = 2$, $k = 1$, $\alpha = 3$

(b) $n = 10$, $k = 2$, $\alpha = 0.6$

(c) $n = 100$, $k = 10$, $\alpha = 0.005$

Com base na Tabela 4.8, embora o Algoritmo 3 atingiu o número máximo de iterações com $(n, k, \alpha) = (100, 10, 0.005)$, teve um desempenho superior a OptStiefel e Conj-Grad. Além disso, para os dois primeiros problemas, o Algoritmo 3 realizou poucas iterações de gradientes conjugados.

Exemplo 2:

(a) $n = 2$, $k = 1$, $\alpha = 9$

(b) $n = 10$, $k = 2$, $\alpha = 3$

(c) $n = 100$, $k = 10$, $\alpha = 1$

(d) $n = 100$, $k = 4$, $\alpha = 2$

Vemos na Tabela 4.9, que o Algoritmo 3 e OptStiefel tiveram um desempenho similar com $(n, k, \alpha) = (2, 1, 9)$, $(10, 2, 3)$. Por outro lado, neste mesma tabela observamos que para valores maiores de n não obtivemos bons resultados. De fato, para o último problema, o Algoritmo 3 atingiu o número máximo de iterações.

4.3.4 Problema de minimização de formas quadráticas heterogêneas

Nesta subseção, consideramos o seguinte problema, que surge em aplicações relacionadas à estatística multivariada [21].
Seja $X \in \mathcal{V}$ e dadas p matrizes simétricas, A_1, A_2 $\ldots, A_p \in \mathbb{R}^{n\times n}$, o problema de minimização de formas quadráticas heterogêneas consiste em resolver

$$\begin{aligned} \text{minimizar} \quad & \sum_{i=1}^{p} X_i^T A_i X_i \\ \text{s.a.} \quad & X^T X = I,\ X \in \mathbb{R}^{n\times p},\ n \geq p, \end{aligned} \tag{4.11}$$

em que X_i denota a i-ésima 'coluna de X. Aqui, a função objetivo é $F(X) = \sum_{i=1}^{p} X_i^T A_i X_i$ e o gradiente de F é

$$G = \nabla F(X) = [G_i] = [2A_i X_i]\,,$$

em que G_i denota a i-ésima coluna de G.
A fim de testar o desempenho numérico dos métodos, consideramos os problemas seguintes, que foram retirados de [28].

$n = 2,\ k = 1,\ \alpha = 3$						
Algoritmo	Res	Feasi	normG	itr	nfeval	CPUtime
Algoritmo 3	8.750e-01	4.441e-16	6.716e-06	5	6(1)	0.002
OptStiefel	8.750e-01	4.441e-16	1.061e-07	4	5	0.003
Conj-Grad	8.750e-01	8.882e-16	4.524e-06	9	26	0.017
$n = 10,\ k = 2,\ \alpha = 0.6$						
Algoritmo 3	8.495e-01	1.573e-15	9.095e-06	24	25(56)	0.025
OptStiefel	8.495e-01	5.443e-15	2.697e-06	18	19	0.032
Conj-Grad	8.495e-01	6.293e-16	7.681e-06	18	34	0.045
$n = 100,\ k = 10,\ \alpha = 0.005$						
Algoritmo 3	1.055e+00	2.420e-14	5.920e-02	999	1001(11132)	2.391
OptStiefel	1.055e+00	6.009e-16	9.185e-06	80	86	0.027
Conj-Grad	1.055e+00	3.968e-15	4.801e-06	82	140	0.125

Tabela 4.8: Resultados numéricos para o Exemplo 1 do Problema de minimização da Energia Total.

$n = 2$, $k = 1$, $\alpha = 9$						
Algoritmo	Res	Feasi	normG	itr	nfeval	CPUtime
Algoritmo 3	1.625e+00	6.661e-16	2.978e-08	5	6(0)	0.0018
OptStiefel	1.625e+00	4.441e-16	1.263e-09	5	6	0.0017
Conj-Grad	1.625e+00	2.220e-16	3.318e-02	2	4	0.01062
$n = 10$, $k = 2$, $\alpha = 3$						
Algoritmo 3	2.505e+00	1.977e-15	4.564e-06	18	19(12)	0.006
OptStiefel	2.505e+00	2.351e-14	1.631e-06	18	19	0.007
Conj-Grad	2.505e+00	6.284e-16	6.615e-06	18	30	0.033
$n = 100$, $k = 10$, $\alpha = 1$						
Algoritmo 3	3.571e+01	1.944e-13	7.379e-06	71	72(170)	0.152
OptStiefel	3.571e+01	5.332e-16	8.851e-06	61	64	0.027
Conj-Grad	3.571e+01	3.015e-15	9.208e-06	52	90	0.104
$n = 100$, $k = 4$, $\alpha = 2$						
Algoritmo 3	7.701e+00	5.974e-14	7.201e-02	999	1001(4429)	1.174
OptStiefel	7.700e+00	3.147e-16	7.536e-06	36	37	0.010
Conj-Grad	7.700e+00	2.158e-16	5.350e-06	29	49	0.039

Tabela 4.9: Resultados numéricos para o Exemplo 2 do Problema de minimização da Energia Total

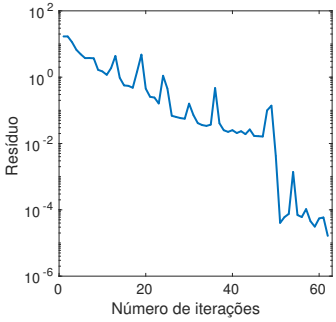

Figura 4.9: Comportamento dos Algoritmos para o Exemplo 1 do Problema de minimização de formas quadráticas heterogêneas

Exemplo 1: As matrizes A_i são da forma

$$A_i = \texttt{diag}\left(\frac{(i-1)n+1}{p} : \frac{1}{p} : \frac{in}{p}\right), \tag{4.12}$$

em que $\texttt{diag}(v)$ é uma matriz diagonal cujos elementos da diagonal são as componentes do vetor v.
No teorema a seguir encontramos a solução exata para o problema 4.11.

Teorema 4.3. *A solução do problema* (4.11) *com matrizes de coeficientes dadas por* (4.12) *é o conjunto de matrizes da forma* $X^* = \begin{pmatrix} Q \\ 0 \end{pmatrix}$, *em que* Q *é uma matriz ortogonal. Além disso, o valor ótimo do problema* (4.11) *é* $\frac{n(p-1)+p+1}{2}$.

Demonstração. A prova do teorema encontra-se em [28] (Proposição 1). ■

A Tabela 4.10 mostra que o Algoritmo 3 teve um bom desempenho para todos os casos. Observe que, ao aumentar as dimensões n e p o nosso Algoritmo se sobressai ainda mais em relação aos outros dois métodos.
A Figura 4.9 mostra o comportamento da condição de otimalidade com $n = 100$ e $p = 10$ ao longo das iterações.

Exemplo 2: As matrizes A_i são da forma

$$A_i = \texttt{diag}\left(\frac{(i-1)n+1}{p} : \frac{1}{p} : \frac{in}{p}\right) + B_i + B_i^T,$$

$n = 100, p = 10$						
Algoritmo 3	4.555e+02	2.362e-13	7.930e-06	62	63(42)	0.057
OptStiefel	4.555e+02	6.642e-16	4.071e-06	88	91	0.058
Conj-Grad	4.555e+02	4.397e-15	5.075e-06	86	142	0.194
$n = 500, p = 50$						
Algoritmo	Res	Feasi	normG	itr	nfeval	CPUtime
Algoritmo 3	1.228e+04	3.083e-11	8.188e-05	104	106(134)	1.182
OptStiefel	1.228e+04	2.193e-15	1.614e-01	1000	1077	4.518
Conj-Grad	1.228e+04	1.832e-14	9.680e-06	205	344	2.075
$n = 1000, p = 100$						
Algoritmo 3	4.955e+04	6.605e-11	1.239e-02	140	142(25)	3.862
OptStiefel	4.955e+04	4.134e-15	6.760e-01	1000	1083	18.103
Conj-Grad	4.955e+04	3.911e-14	8.878e-06	324	545	11.098
$n = 2000, p = 200$						
Algoritmo 3	1.991e+05	1.592e-10	3.146e-02	69	71(0)	11.829
OptStiefel	1.991e+05	8.030e-15	1.621e+01	1000	3983	208.095
Conj-Grad	1.991e+05	1.372e-13	5.411e-05	355	617	60.973

Tabela 4.10: Resultados numéricos para o Exemplo 1 do Problema de minimização de formas quadráticas heterogêneas.

em que B_i são matrizes randômicas geradas por $B_i = \texttt{randn(n)}/10$.

Na Tabela 4.11 observa-se que o desempenho da estratégia de restauração inexata não monótona foi superior aos outros dois métodos. Cabe ressaltar que o número de avaliações da função é maior à medida que a dimensão p aumenta.

$p = 5$						
Algoritmo	Res	Feasi	normG	itr	nfeval	CPUtime
Algoritmo 3	9.989e+02	2.435e-13	6.304e-05	259	261(224)	13.066
OptStiefel	9.989e+02	4.948e-16	3.955e-05	557	594	19.623
Conj-Grad	9.989e+02	2.439e-15	9.713e-06	291	494	16.349
$p = 10$						
Algoritmo 3	2.241e+03	8.737e-12	3.065e-05	340	342(215)	32.3481
OptStiefel	2.241e+03	5.833e-16	3.488e-04	612	676	45.366
Conj-Grad	2.241e+03	4.696e-15	8.58e-06	456	774	51.781
$p = 20$						
Algoritmo 3	4.716e+03	7.765e-12	3.122e-05	220	222(243)	48.445
OptStiefel	4.716e+03	1.126e-15	8.650e-04	586	668	87.465
Conj-Grad	4.716e+03	7.412e-15	9.640e-06	313	543	72.321

Tabela 4.11: Resultados numéricos para o Exemplo 2 do Problema de minimização de formas quadráticas heterogêneas com $n = 500$.

Conclusões

Neste trabalho, estudamos o problema de minimização matricial com restrições de ortogonalidade. Temos interesse em estudar este tipo de problemas devido às inúmeras aplicações e à dificuldade computacional para resolvê-las. Existem vários métodos que podem ser empregados para a resolução destes problemas. Dentre eles, encontra-se o método de Restauração Inexata proposto por Fischer e Friedlander [1]. A escolha desta abordagem foi motivada pelo fato de que o Algoritmo de Restauração Inexata possui duas fases: viabilidade e otimalidade, com flexibilidade na escolha de métodos para realizar estas duas fases. Portanto, nos permite explorar as características e propriedades do problema, principalmente do conjunto viável.

Na primeira etapa deste trabalho, introduzimos o método de Restauração Inexata não monótona para resolver problemas com restrições de igualdade. Na fase de otimização, escolhemos as direções de descida gradiente projetado no subespaço tangente. Além disso, a melhora na viabilidade e na otimalidade é controlada pela função de mérito. Utilizando hipóteses adicionais é possível garantir a convergência global. Nossa proposta consiste em incorporar a idéia de não monotonia na fase de otimalidade, o qual permite melhorar a eficiência do método. Mostramos a boa definição desta nova proposta e além disso, obtemos que todo de acumulação da sequência gerada pelo Algoritmo 2 é viável.

Na segunda etapa do trabalho, estudamos o método de Restauração Inexata não monótono aplicado ao problema de minimização com restrições de ortogonalidade. Cabe ressaltar que aproveitamos a estrututa e características da região viável a fim de encontrar representações para o subespaço tangente às restrições assim com projeções dos pontos neste subespaço. Os resultados teóricos obtidos nesta etapa nos permitem estabelecer o critério de parada e a estimativa para o multiplicador de Lagrange. Na fase de otimalidade, a escolha das direções de descida podem decorrer de duas formas: a projeção do gradiente no subespaço

tangente com tamanho de passo espectral ou a solução do subproblema de minimização de um modelo quadrático para o Lagrangiano restrito ao subespaço tangente. Na fase de viabilidade, o ponto restaurado é obtido utilizando a transformação de Cayley. Além disso, com hipóteses adicionais garantimos a convergência global do algoritmo a pontos viáveis, que satisfazem adequadas condições de qualificação.

Na implementação numérica o método proposto mostrou-se bastante competitivo, obtendo um bom desempenho na maioria dos problemas testes. No Capítulo 4, para a convergência dos testes, foi de muita importância o ajuste dos parâmetros: grau de não monotonicidade η_k e número de iterações locais M. Destacamos que o método proposto baseado na Restauração Inexata apresenta uma alternativa com sólida base teórica para resolver problemas de minimização com restrições de ortogonalidade, o qual resulta num aporte significativo na área de otimização.
Finalmente, propomos algumas sugestões para trabalhos futuros:

1. Aplicar o método proposto para resolver problemas

$$\begin{aligned} \text{minimizar} \quad & F(X) \\ \text{s.a.} \quad & X^T X = I, \\ & X \geq 0 \\ & X \in \Omega, \end{aligned}$$

   em que $X \geq 0$ denota cada entrada de $X_{i,j}$ é maior igual a 0.

2. Na fase de otimalidade, aplicar outros métodos para resolver o subproblema tangencial.

3. Estender a teoria de convergência do algoritmo de restauração inexata não monótono (capítulo 2) para problemas gerais de programação não linear.

Apêndices

(n,p)	nfeval, CPUtime				
	$\eta_k = 0$	$\eta_k = 0.25$	$\eta_k = 0.5$	$\eta_k = 0.85$	$\eta_k = 0.99$
(500,2)	93(0.527)	70 (0.233)	112(0.232)	67(0.157)	80(0.138)
(1000,2)	100(0.544)	81(0.363)	105(0.388)	100(0.524)	95(0.709)
(3000,2)	136(5.568)	167(7.096)	119(2.481)	163(5.260)	120(6.347)
(5000,2)	126(20.163)	202(20.769)	170(18.735)	147(9.193)	138(8.017)
(500,50)	99(0.899)	124(0.898)	135(0.888)	123(0.892)	241(1.326)
(1000,50)	128(2.248)	162(2.584)	134(2.344)	186(3.326)	159(2.756)
bcsstk36	148(1.423)	103(0.940)	142(1.272)	127(1.122)	125(1.137)
bcsstk38	1001(3.156)	1001(3.319)	1001(3.158)	1001(3.178)	1001(3.226)
bcsstm39	54(0.679)	41(0.455)	52(0.587)	46(0.500)	64(0.794)
mhd4800b	18(0.069)	23(0.071)	14(0.045)	22(0.045)	18(0.079)

Tabela 12: Número de avaliações da função e tempo de execução do Algoritmo 3 para a resolução do Problema de Autovalor Linear, utilizando diversos valores do parâmetro η_k.

(n,p)	nfeval, CPUtime				
	$\eta_k = 0$	$\eta_k = 0.25$	$\eta_k = 0.5$	$\eta_k = 0.85$	$\eta_k = 0.99$
			Exemplo 1		
(100,5)	11(0.011)	11 (0.044)	11(0.030)	11(0.085)	11(0.050)
(100,50)	11(0.078)	11(0.061)	11(0.063)	11(0.044)	11(0.049)
(500,70)	11(0.167)	11(0.171)	11(0.182)	11(0.138)	11(0.172)
(1000,100)	11(0.466)	11(0.553)	11(0.461)	11(0.443)	11(0.478)
			Exemplo 2		
(100,5)	316(0.161)	361(0.231)	415(0.156)	469(0.147)	549(0.237)
(500,5)	432(0.760)	395(0.646)	358(0.607)	388(0.576)	617(0.834)
(800,5)	513(2.323)	427(2.188)	514(2.215)	533(1.987)	656(2.513)
(1000,5)	480(3.421)	611(3.189)	513(3.120)	345(2.789)	663(4.880)
			Exemplo 3		
(50,5)	109(0.149)	139(0.113)	149(0.128)	111(0.085)	93(0.111)
(50,20)	103(0.173)	91(0.151)	89(0.139)	103(0.088)	125(0.181)
(95,2)	159(0.196)	107(0.144)	94(0.140)	82(0.107)	155(0.150)
(95,20)	93(0.193)	149(0.227)	121(0.240)	105(0.181)	94(0.0193)

Tabela 13: Número de avaliações da função e tempo de execução do Algoritmo 3 para a resolução do Problema de Procrustes, utilizando diversos valores do parâmetro η_k.

(n,p,α)	nfeval, CPUtime				
	$\eta_k = 0$	$\eta_k = 0.25$	$\eta_k = 0.5$	$\eta_k = 0.85$	$\eta_k = 0.99$
(2,1,3)	9(0.003)	5(0.007)	7(0.006)	5(0.002)	8(0.005)
(10,2,3)	27(0.049)	28(0.061)	19(0.043)	18(0.039)	20(0.059)
(100,10,0.005)	1001(2.684)	1001(2.891)	1001(2.606)	1001(2.466)	1001(2.506)
(2,1,9)	6(0.035)	7(0.035)	5(0.021)	8(0.009)	4(0.016)
(10,2,3)	19(0.056)	17(0.048)	19(0.050)	17(0.039)	20(0.013)
(100,10,1)	109(0.178)	102(0.167)	78(0.203)	82(0.124)	95(0.166)
(200,10,1)	74(0.191)	60(0.190)	60(0.159)	60(0.127)	60(0.137)
(400,10,1)	60(0.264)	62(0.277)	61(0.276)	61(0.254)	61(0.266)
(800,10,1)	66(1.265)	66(1.313)	66(1.228)	63(1.204)	66(1.302))
(100,20,0.1)	158(0.405)	146(0.363)	161(0.448)	144(0.355)	185(1.606)
(100,20,1)	1001(1.900)	126(0.222)	132(0.690)	147(0.252)	301(0.518)

Tabela 14: Número de avaliações da função e tempo de execução do Algoritmo 3 para a resolução do Problema de minimização total da energia, utilizando diversos valores do parâmetro η_k.

(n,p)	nfeval, CPUtime				
	$\eta_k = 0$	$\eta_k = 0.25$	$\eta_k = 0.5$	$\eta_k = 0.85$	$\eta_k = 0.99$
			Exemplo 1		
(50,5)	37(0.100)	43(0.050)	47(0.086)	49(0.027)	48(0.073)
(500,5)	102(0.163)	103(0.147)	152(0.177)	129(0.132)	131(0.151)
(1000,5)	208(0.303)	310(0.314)	161(0.374)	217(0.285)	179(0.294)
			Exemplo 2		
(50,5)	99(0.119)	85(0.132)	83(0.139)	53(0.061)	90(0.114)
(100,5)	114(0.250)	78(0.143)	120(0.208)	97(0.130)	85(0.200)
(200,5)	178(0.838)	337(1.257)	165(0.696)	114(0.587)	167(0.685)

Tabela 15: Número de avaliações da função e tempo de execução do Algoritmo 3 para a resolução do Problema de minimização de formas quadráticas heterogêneas, utilizando diversos valores do parâmetro η_k.

(n,p)	nfeval, CPUtime					
	$M=0$	$M=10$	$M=15$	$M=20$	$M=50$	$M=100$
(500,2)	127(0.366)	89 (0.181)	43(0.111)	106(0.195)	74(0.193)	91(0.184)
(1000,2)	98(0.414)	81(0.563)	67(0.352)	91(0.476)	91(0.630)	99(0.495)
(2000,2)	98(2.145)	129(1.158)	100(0.883)	135(2.198)	122(1.100)	146(2.646)
(3000,2)	101(2.144)	83(1.706)	113(3.933)	123(2.705)	155(3.396)	157(5.383)
(5000,2)	157(15.857)	307(32.104)	163(9.476)	267(21.809)	239(22.210)	190(24.520)
(500,100)	177(2.978)	111(1.804)	103(1.629)	211(3.318)	201(3.170)	174(2.695)
(1000,100)	230(9.014)	210(7.105)	152(5.057)	178(6.091)	268(9.749)	398(13.081)

Tabela 16: Número de avaliações da função e tempo de execução do Algoritmo 3 para a resolução do Problema de Autovalor Linear, utilizando diferentes valores do M.

(n,p)	nfeval, CPUtime					
	$M = 0$	$M = 10$	$M = 15$	$M = 20$	$M = 50$	$M = 100$
			Exemplo 1			
(100,5)	14(0.047)	10 (0.037)	11(0.023)	11(0.044)	11(0.046)	11(0.050)
(100,50)	11(0.047)	11(0.027)	11(0.017)	11(0.014)	11(0.082)	11(0.080)
(500,70)	11(0.173)	11(0.165)	11(0.156)	11(0.224)	11(0.186)	11(0.174)
(1000,100)	11(0.457)	11(0.438)	11(0.429)	11(0.469)	11(0.613)	11(0.449)
			Exemplo 2			
(100,5)	779(0.291)	336(0.184)	306(0.102)	368(0.167)	475(0.227)	447(0.181)
(500,5)	406(0.612)	430(0.783)	296(0.507)	426(0.731)	727(0.956)	482(0.705)
(800,5)	511(2.510)	492(1.982)	428(1.968)	561(2.269)	523(2.164)	497(1.984)
(1000,5)	429(2.858)	507(3.300)	543(3.299)	587(4.133)	574(3.675)	735(4.227)
			Exemplo 3			
(50,5)	101(0.079)	91(0.087)	113(0.048)	127(0.132)	133(0.111)	139(0.127)
(50,20)	197(0.243)	137(0.227)	151(0.170)	1187(0.204)	127(0.167)	107(0.192)
(95,5)	140(0.185)	121(0.173)	112(0.093)	171(0.255)	179(0.460)	165(0.174)
(95,20)	118(0.215)	151(0.214)	91(0.167)	124(0.247)	75(0.189)	113(0.195)

Tabela 17: Número de avaliações da função e tempo de execução do Algoritmo 3 para a resolução do Problema de Procrustes, utilizando diferentes valores do M.

(n,p,α)	nfeval, CPUtime					
	$\eta_k = 0$	$\eta_k = 0.25$	$\eta_k = 0.5$	$\eta_k = 0.85$	$\eta_k = 0.99$	
(2,1,3)	5(0.036)	6(0.031)	5(0.022)	8(0.038)	8(0.036)	8(0.004)
(10,2,0.6)	20(0.064)	30(0.027)	23(0.034)	20(0.048)	19(0.053)	19(0.047)
(100,10,0.005)	1001(2.732)	1001(2.646)	1001(2.463)	1001(2.475)	1001(2.522)	1001(2.681)
(2,1,9)	5(0.026)	5(0.019)	5(0.010)	7(0.016)	7(0.035)	6(0.029)
(10,2,3)	19(0.057)	17(0.036)	18(0.015)	20(0.054)	21(0.057)	18(0.042)
(100,10,1)	82(0.173)	91(0.162)	89(0.119)	113(0.180)	102(0.710)	97(0.147)
(200,10,1)	60(0.255)	60(0.161)	60(0.139)	96(0.259)	60(0.187)	83(0.186)
(400,10,1)	65(0.296)	61(0.255)	61(0.241)	89(0.385)	62(0.272)	62(0.251)
(800,10,1)	66(1.327)	66(1.460)	63(1.137)	66(1.359)	66(1.250))	61(1.194)
(100,20,0.1)	166(1.317)	151(1.194)	193(0.923)	209(1.297)	156(1.070)	177(1.514)
(100,20,1)	124(0.344)	145(0.223)	227(0.356)	185(0.315)	146(0.281)	117(0.554)

Tabela 18: Número de avaliações da função e tempo de execução do Algoritmo 3 para a resolução do Problema de minimização total da energia, utilizando diferentes valores do M.

(n,p)	nfeval, CPUtime					
	$M = 0$	$M = 10$	$M = 15$	$M = 20$	$M = 50$	$M = 100$
			Exemplo 1			
(50,5)	51(0.109)	41(0.0587)	50(0.079)	59(0.089)	49(0.091)	53(0.078)
(500,5)	127(0.160)	144(0.181)	143(0.138)	126(0.177)	124(0.172)	182(0.201)
(1000,5)	205(0.288)	191(0.431)	191(0.265)	153(0.298)	1760.262)	155(0.288)
			Exemplo 2			
(50,5)	87(0.115)	141(0.090)	67(0.076)	89(0.122)	86(0.133)	103(0.111)
(100,5)	178(0.239)	105(0.161)	76(0.138)	147(0.228)	123(0.201)	141(0.220)
(200,5)	296(13.499)	286(15.072)	187(0.752)	504(23.500)	166(11.307)	234(13.236)

Tabela 19: Número de avaliações da função e tempo de execução do Algoritmo 3 para a resolução do Problema de minimização de formas quadráticas heterogêneas, utilizando diferentes valores do M.

Referências Bibliográficas

[1] A. Fischer and A. Friedlander, "A new line search inexact restoration approach for nonlinear programming," *Computational Optimization and Applications*, vol. 46, no. 2, pp. 333–346, 2010.

[2] H. Zhang and W. W. Hager, "A nonmonotone line search technique and its application to unconstrained optimization," *SIAM journal on Optimization*, vol. 14, no. 4, pp. 1043–1056, 2004.

[3] M. Shariff, "A constrained conjugate gradient method and the solution of linear equations," *Computers & Mathematics with Applications*, vol. 30, no. 11, pp. 25–37, 1995.

[4] Z. Wen and W. Yin, "A feasible method for optimization with orthogonality constraints," *Mathematical Programming*, vol. 142, no. 1-2, pp. 397–434, 2013.

[5] N. Boumal, B. Mishra, P.-A. Absil, R. Sepulchre, *et al.*, "Manopt, a matlab toolbox for optimization on manifolds.," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1455–1459, 2014.

[6] G. H. Golub and C. F. Van Loan, *Matrix computations*, vol. 3. JHU Press, 2012.

[7] Y. Saad, *Numerical methods for large eigenvalue problems*. Manchester University Press, 1992.

[8] J. Francisco and T. Martini, "Spectral projected gradient method for the Procrustes problem," *TEMA (São Carlos)*, vol. 15, no. 1, pp. 83–96, 2014.

[9] L. Eldén and H. Park, "A Procrustes problem on the Stiefel manifold," *Numerische Mathematik*, vol. 82, no. 4, pp. 599–619, 1999.

[10] P. H. Schönemann, "A generalized solution of the orthogonal Procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966.

[11] Z. Zhao, Z.-J. Bai, and X.-Q. Jin, "A riemannian newton algorithm for nonlinear eigenvalue problems," *SIAM Journal on Matrix Analysis and Applications*, vol. 36, no. 2, pp. 752–774, 2015.

[12] M. Joho and H. Mathis, "Joint diagonalization of correlation matrices by using gradient methods with application to blind signal separation," *Sensor Array and Multichannel Signal Processing Workshop Proceedings*, pp. 273–277, 2002.

[13] F. J. Theis, T. P. Cason, and P.-A. Absil, *Soft dimension reduction for ICA by joint diagonalization on the Stiefel manifold.* Springer, 2009.

[14] A. d'Aspremont, L. El Ghaoui, M. I. Jordan, and G. R. Lanckriet, "A direct formulation for sparse pca using semidefinite programming.," *SIAM Review*, vol. 49, no. 3, pp. 434–448, 2007.

[15] M. Journée, Y. Nesterov, P. Richtárik, and R. Sepulchre, "Generalized power method for sparse principal component analysis," *Journal of Machine Learning Research*, vol. 11, no. Feb, pp. 517–553, 2010.

[16] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *Journal of Computational and Graphical Statistics*, vol. 15, no. 2, pp. 265–286, 2006.

[17] I. Grubišić and R. Pietersz, "Efficient rank reduction of correlation matrices," *Linear Algebra and its Applications*, vol. 422, no. 2-3, pp. 629–653, 2007.

[18] R. Pietersz 4 and P. J. Groenen, "Rank reduction of correlation matrices by majorization," *Quantitative Finance*, vol. 4, no. 6, pp. 649–662, 2004.

[19] P. Jaeckel and R. Rebonato, "The most general methodology for creating a valid correlation matrix for risk management and option pricing purposes," *Journal of Risk*, vol. 2, no. 2, pp. 17–28, 1999.

[20] E. Stiefel, "Richtungsfelder und fernparallelismus in n-dimensionalen mannigfaltigkeiten," *Commentarii Mathematici Helvetici*, vol. 8, no. 1, pp. 305–353, 1935.

[21] M. Bolla, G. Michaletzky, G. Tusnády, and M. Ziermann, "Extrema of sums of heterogeneous quadratic forms," *Linear Algebra and its Applications*, vol. 269, no. 1-3, pp. 331–365, 1998.

[22] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM Journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, 1998.

[23] Z. Wen, C. Yang, X. Liu, and Y. Zhang, "Trace-penalty minimization for large-scale eigenspace computation," *Journal of Scientific Computing*, vol. 66, no. 3, pp. 1175–1203, 2016.

[24] H. Oviedo, H. Lara, and O. Dalmau, "A non-monotone linear search method with mixed direction on Stiefel manifold," *arXiv preprint arXiv:1702.04303*, 2017.

[25] W. Chen, H. Ji, and Y. You, "An augmented Lagrangian method for 1-regularized optimization problems with orthogonality constraints," *SIAM Journal on Scientific Computing*, vol. 38, no. 4, pp. B570–B592, 2016.

[26] S. Wright and J. Nocedal, "Numerical optimization," *Springer Science*, vol. 35, pp. 67–68, 1999.

[27] R. Lai and S. Osher, "A splitting method for orthogonality constrained problems," *Journal of Scientific Computing*, vol. 58, no. 2, pp. 431–449, 2014.

[28] H. Zhu, X. Zhang, D. Chu, and L.-Z. Liao, "Nonconvex and nonsmooth optimization with generalized orthogonality constraints: An approximate augmented Lagrangian method," *Journal of Scientific Computing*, pp. 1–42, 2017.

[29] J. M. Martinez and E. A. Pilotta, "Inexact restoration methods for nonlinear programming: advances and perspectives," *Optimization and Control with applications*, pp. 271–291, 2005.

[30] J. Martínez, "Inexact-restoration method with Lagrangian tangent decrease and new merit function for nonlinear programming," *Journal of Optimization Theory and Applications*, vol. 111, no. 1, pp. 39–58, 2001.

[31] J. M. Martínez and B. F. Svaiter, "A practical optimality condition without constraint qualifications for nonlinear programming," *Journal of Optimization Theory and Applications*, vol. 118, no. 1, pp. 117–133, 2003.

[32] E. Birgin and J. Martínez, "Local convergence of an inexact-restoration method and numerical experiments," *Journal of Optimization Theory and Applications*, vol. 127, no. 2, pp. 229–247, 2005.

[33] M. A. Gomes-Ruggiero, J. M. Martínez, and S. A. Santos, "Spectral projected gradient method with inexact restoration for minimization with nonconvex constraints," *SIAM Journal on Scientific Computing*, vol. 31, no. 3, pp. 1628–1652, 2009.

[34] L. F. Bueno, G. Haeser, and J. M. Martínez, "A flexible inexact-restoration method for constrained optimization," *Journal of Optimization Theory and Applications*, vol. 165, no. 1, pp. 188–208, 2015.

[35] E. W. Karas, A. P. Oening, and A. A. Ribeiro, "Global convergence of slanting filter methods for nonlinear programming," *Applied Mathematics and Computation*, vol. 200, no. 2, pp. 486–500, 2008.

[36] L. Grippo, F. Lampariello, and S. Lucidi, "A truncated Newton method with nonmonotone line search for unconstrained optimization," *Journal of Optimization Theory and Applications*, vol. 60, no. 3, pp. 401–419, 1989.

[37] P. L. Toint, "An assessment of nonmonotone linesearch techniques for unconstrained optimization," *SIAM Journal on Scientific Computing*, vol. 17, no. 3, pp. 725–739, 1996.

[38] R. Borsdorf and N. J. Higham, "A preconditioned Newton algorithm for the nearest correlation matrix," *IMA Journal of Numerical Analysis*, vol. 30, no. 1, pp. 94–107, 2009.

[39] Y.-H. Dai and L.-Z. Liao, "R-linear convergence of the Barzilai and Borwein gradient method," *IMA Journal of Numerical Analysis*, vol. 22, no. 1, pp. 1–10, 2002.

[40] M. Raydan, "The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem," *SIAM Journal on Optimization*, vol. 7, no. 1, pp. 26–33, 1997.

[41] J. Mo, C. Liu, and S. Yan, "A nonmonotone trust region method based on nonincreasing technique of weighted average of the successive function values," *Journal of Computational and Applied Mathematics*, vol. 209, no. 1, pp. 97–108, 2007.

[42] K. Amini, M. Ahookhosh, and H. Nosratipour, "An inexact line search approach using modified nonmonotone strategy for unconstrained optimization," *Numerical Algorithms*, vol. 66, no. 1, pp. 49–78, 2014.

[43] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifold*. Princeton University Press, 2009.

[44] J. C. Gower and G. B. Dijksterhuis, *Procrustes problems*, vol. 30. Oxford University Press on Demand, 2004.

[45] X. Liu, Z. Wen, X. Wang, M. Ulbrich, and Y. Yuan, "On the analysis of the discretized kohn–sham density functional theory," *SIAM Journal on Numerical Analysis*, vol. 53, no. 4, pp. 1758–1785, 2015.

[46] X. Liu, Z. Wen, and Y. Zhang, "Limited memory block krylov subspace optimization for computing dominant singular value decompositions," *SIAM Journal on Scientific Computing*, vol. 35, no. 3, pp. A1641–A1668, 2013.

[47] H. Sato and T. Iwai, "A riemannian optimization approach to the matrix singular value decomposition," *SIAM Journal on Optimization*, vol. 23, no. 1, pp. 188–212, 2013.

[48] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA journal of numerical analysis*, vol. 8, no. 1, pp. 141–148, 1988.

[49] R. Janin, "Directional derivative of the marginal function in nonlinear programming," *Sensitivity, Stability and Parametric Analysis*, vol. 21, pp. 110–126, 1984.

[50] D. P. Bertsekas, *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.

[51] C. D. Meyer, *Matrix analysis and applied linear algebra*, vol. 2. Philadelphia: SIAM, 2000.

[52] J. Balogh, T. Csendes, and T. Rapcsák, "Some global optimization problems on Stiefel manifolds," *Journal of Global Optimization*, vol. 30, no. 1, pp. 91–101, 2004.

[53] E. Kreyszig, *Introductory functional analysis with applications*. Wiley, 1978.

[54] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical programming*, vol. 91, no. 2, pp. 201–213, 2002.

[55] J. R. Hurley and R. B. Cattell, "The Procrustes program: Producing direct rotation to test a hypothesized factor structure," *Systems Research and Behavioral Science*, vol. 7, no. 2, pp. 258–262, 1962.

[56] T. Bell, "Global positioning system-based attitude determination and the orthogonal Procrustes problem," *Journal of Guidance Control and Dynamics*, vol. 26, no. 5, pp. 820–821, 2003.

[57] J. B. Francisco and F. S. V. Bazán, "Nonmonotone algorithm for minimization on closed sets with applications to minimization on Stiefel manifolds," *Journal of Computational and Applied Mathematics*, vol. 236, no. 10, pp. 2717–2727, 2012.

[58] C. Yang, J. C. Meza, B. Lee, and L.-W. Wang, "KSSOLV- a matlab toolbox for solving the Kohn-Sham equations," *ACM Transactions on Mathematical Software (TOMS)*, vol. 36, no. 2, p. 10, 2009.