

UNICAMP

UNIVERSIDADE ESTADUAL DE
CAMPINAS

Instituto de Matemática, Estatística e
Computação Científica

EMIDIO SANTOS PORTILHO JUNIOR

**Métodos de pontos interiores para resolução de
problemas de regularização de Tikhonov de
grande porte**

Campinas

2020

Emidio Santos Portilho Junior

Métodos de pontos interiores para resolução de problemas de regularização de Tikhonov de grande porte

Tese apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutor em Matemática Aplicada.

Orientador: Aurelio Ribeiro Leite de Oliveira

Este exemplar corresponde à versão final da Tese defendida pelo aluno Emidio Santos Portilho Junior e orientada pelo Prof. Dr. Aurelio Ribeiro Leite de Oliveira.

Campinas

2020

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

P836m Portilho Junior, Emidio Santos, 1984-
Métodos de pontos interiores para resolução de problemas de regularização de Tikhonov de grande porte / Emidio Santos Portilho Junior. – Campinas, SP : [s.n.], 2020.

Orientador: Aurelio Ribeiro Leite de Oliveira.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Regularização de Tikhonov. 2. Programação quadrática. 3. Métodos de pontos interiores. I. Oliveira, Aurelio Ribeiro Leite de, 1962-. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Interior point methods for solving large-scale Tikhonov regularization problems

Palavras-chave em inglês:

Tikhonov regularization

Quadratic programming

Interior-point methods

Área de concentração: Matemática Aplicada

Titulação: Doutor em Matemática Aplicada

Banca examinadora:

Aurelio Ribeiro Leite de Oliveira [Orientador]

Estevão Esmi Laureano

João Batista Florindo

Luiz Rafael dos Santos

Gláucia Maria Bressan

Data de defesa: 18-02-2020

Programa de Pós-Graduação: Matemática Aplicada

Identificação e informações acadêmicas do(a) aluno(a)

- ORCID do autor: <https://orcid.org/0000-0002-4663-8849>

- Currículo Lattes do autor: <http://lattes.cnpq.br/1186170234447332>

**Tese de Doutorado defendida em 18 de fevereiro de 2020 e aprovada
pela banca examinadora composta pelos Profs. Drs.**

Prof(a). Dr(a). AURELIO RIBEIRO LEITE DE OLIVEIRA

Prof(a). Dr(a). ESTEVÃO ESMI LAUREANO

Prof(a). Dr(a). JOÃO BATISTA FLORINDO

Prof(a). Dr(a). LUIZ RAFAEL DOS SANTOS

Prof(a). Dr(a). GLAUCIA MARIA BRESSAN

A Ata da Defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria de Pós-Graduação do Instituto de Matemática, Estatística e Computação Científica.

*Dedico este trabalho a minha esposa e aos meus pais, com todo meu amor e gratidão,
à minha esposa por todo o apoio e dedicação dos últimos anos e aos meus pais por
tudo que fizeram por mim ao longo da vida.*

Agradecimentos

Agradeço a Jeová Deus por me guiar e dar forças para enfrentar cada etapa da minha vida.

À minha esposa Luiza pela amizade, carinho e incentivo constante. Aos meus pais, Emídio e Ozana, que sempre incentivaram e apoiaram os meus estudos. Aos meus irmãos por sempre torcerem para que eu alcança-se meus objetivos.

Agradeço especialmente ao Prof. Dr. Aurelio R. L. Oliveira por todo o auxílio prestado ao longo do meu percurso na Unicamp, pela orientação, por todas as conversas, pelas dicas e sugestões fundamentais para a realização desta pesquisa, além do incentivo, confiança e amizade.

Aos membros da banca, Dra. Glaucia Bressan (UTFPR), Dr. Luiz Rafael Santos (UFSC), Dr. Estevão Esmi Laureano (UNICAMP), Dr. João Batista Florindo (UNICAMP) por aceitarem o convite para avaliação deste trabalho.

Aos meus amigos Rodrigo Bloot, Luciano Panek e Nayene Michele Pitta Paião, pela amizade e incentivo para que eu retomasse meus estudos.

Ao IMECC e à Unicamp por todas as oportunidades concedidas.

Ao CEPETRO pela bolsa de estudos provisória que me deu suporte financeiro entre os anos de 2008 e 2009 e me permitiu dar continuidade aos meus estudos.

À Unioeste - Universidade Estadual do Oeste do Paraná - por ter me concedido afastamento de meu setor de trabalho para realização deste aperfeiçoamento profissional.

*“Libertem-se da escravidão mental.
Ninguém além de nós mesmos pode
libertar nossas mentes”.
(Bob Marley, Redemption Song)*

Resumo

A discretização de problemas inversos lineares geralmente resultam em sistemas cujos valores singulares da matriz dos coeficientes se concentram próximos à origem e decaem gradativamente a zero, o que torna a matriz severamente mal condicionada. Tais sistemas são frequentemente chamados de problemas lineares discretos mal-postos. Vários métodos baseados em decomposição SVD são propostos para resolução de problemas mal-postos, oriundos da discretização de problemas inversos lineares. A solução direta de sistemas de equações lineares mal-postos com dados contaminados por erros geralmente não fornece resultados significativos, porque o erro propagado destrói a solução calculada. Os problemas precisam ser modificados para reduzir sua sensibilidade ao erro nos dados. Em 1963 Andrei Nikolaevich Tikhonov propôs uma formulação geral para problemas mal-postos chamada regularização ou problema de regularização de Tikhonov. Nosso foco está na resolução do problema de regularização de Tikhonov. A dificuldade encontrada pelos métodos baseados em SVD para resolução do problema de regularização de Tikhonov está no custo computacional e numérico envolvido no cálculo da decomposição SVD de uma matriz, em especial no caso de matrizes de grande porte. Nesta tese reescrevemos o problema em questão como um problema de programação quadrática através de uma formulação Primal-Dual com barreira logarítmica. Na busca por maior eficiência, incluímos a direção de centragem e a direção de correção no mesmo método, obtendo um método do tipo Preditor-Corretor. Implementamos um Método de pontos interiores Primal-Dual e Preditor-Corretor combinado a dois preconditionadores, a Fatoração Controlada de Cholesky e o Precondicionador Separador. A eficiência dos métodos propostos é comprovada pelos resultados de experimentos numéricos com problemas de regularização de Tikhonov.

Palavras-chave: Problema de regularização de Tikhonov, programação quadrática, métodos de pontos interiores.

Abstract

The discretization of inverse linear problems usually results in systems which the singular values of the matrix accumulate at the origin and decay gradually to zero, which makes the matrix severely ill-conditioned. Such systems are often referred to as linear discrete ill-posed problems. Several methods based on SVD decomposition are proposed for solving problems arising from the discretization of inverse linear problems. The direct solution of linear ill-posed problems equations with error-contaminated data generally does not yield significant results. The propagated error destroys the computed solution. The problems need to be modified to reduce their sensitivity to data error. In 1963 Andrei Nikolaevich Tikhonov proposed a general formulation for ill-posed problems called regularization or Tikhonov regularization problem. Our focus is to solve the Tikhonov regularization problem. The difficulty encountered by SVD based methods for solving the Tikhonov regularization problems is on the computational and numerical cost involved in calculating the SVD decomposition of a matrix, especially in the case of large-scale matrices. In this thesis we rewrite the problem in question as a quadratic programming problem through a Primal-Dual formulation with a logarithmic barrier function. In the search for more efficiency, we include the centering direction and the correction direction in the same method, obtaining a Predictor-Corrector method. We implemented a Primal-Dual and Predictor-Corrector interior point method combined with two preconditioners: Controlled Cholesky Factorization and Splitting Preconditioner. The efficiency of the proposed methods is proved by the results of numerical experiments with Tikhonov regularization problems.

Keywords: Tikhonov regularization problem, Quadratic Programming, Interior Point Methods.

Lista de ilustrações

Figura 1 – Gráfico de $\ x\ _2$ em função de $\ \mathbf{r}\ _2$	24
Figura 2 – Problema de teste Blur: (a) Imagem de teste original, (b) Imagem de teste borrada, (c) Imagem recuperada pelo MPILU.	69
Figura 3 – Comparação entre a solução exata e a solução obtida pelo MPCFCC para o problema Shaw com $NL = 10^{-4}$	71
Figura 4 – Problema de teste PRtomo: (a) Imagem de teste original, (b) Imagem recuperada.	77

Lista de tabelas

Tabela 1 – MPIFCC aplicado à matriz de Hilbert.	63
Tabela 2 – MPILU aplicado à matriz de Hilbert.	64
Tabela 3 – MPCFCC aplicado à matriz de Hilbert.	64
Tabela 4 – MPCLU aplicado à matriz de Hilbert.	64
Tabela 5 – MPIFCC aplicado à matriz de Baart.	65
Tabela 6 – MPILU aplicado à matriz de Baart.	66
Tabela 7 – MPCFCC aplicado à matriz de Baart.	66
Tabela 8 – MPCLU aplicado à matriz de Baart.	66
Tabela 9 – MPCLU aplicado à matriz de Baart de ordem $n=18000$	67
Tabela 10 – MPIFCC aplicado à matriz de Blur.	67
Tabela 11 – MPILU aplicado à matriz de Blur.	68
Tabela 12 – MPCFCC aplicado à matriz de Blur.	68
Tabela 13 – MPCLU aplicado à matriz de Blur.	68
Tabela 14 – MPIFCC aplicado à matriz de Shaw.	70
Tabela 15 – MPILU aplicado à matriz de Shaw.	70
Tabela 16 – MPCFCC aplicado à matriz de Shaw.	70
Tabela 17 – MPCLU aplicado à matriz de Shaw.	70
Tabela 18 – MPIFCC aplicado à matriz de Phillips.	72
Tabela 19 – MPILU aplicado à matriz de Phillips.	72
Tabela 20 – MPCFCC aplicado à matriz de Phillips.	72
Tabela 21 – MPCLU aplicado à matriz de Phillips.	72
Tabela 22 – Dimensões dos Problemas de Testes	73
Tabela 23 – Resultados obtidos pelos MPI's para $NL=10^{-4}$	73
Tabela 24 – Resultados obtidos pelos MPC's para $NL=10^{-4}$	74
Tabela 25 – Resultados obtidos pelos MPI's para $NL=10^{-3}$	74
Tabela 26 – Resultados obtidos pelos MPC's para $NL=10^{-3}$	74
Tabela 27 – Resultados obtidos pelos MPI's para $NL=10^{-2}$	74
Tabela 28 – Resultados obtidos pelos MPC's para $NL=10^{-2}$	75
Tabela 29 – Dimensões dos Problemas de Testes	76
Tabela 30 – MPCLU vs SVD	77

Lista de abreviaturas e siglas

<i>D</i>	Formulação Dual
FCC	Fatoração Controlada de Cholesky
KKT	Karush-Kunh-Tucker
LU	Fatoração LU
MGC	Método de Gradientes Conjugados
MGCP	Método de Gradientes Conjugados preconditionado
MPI	Método de pontos interiores
MPC	Método Preditor-Corretor de Mehrotra
<i>P</i>	Formulação Primal
PL	Problema de Programação Linear
SVD	Decomposição SVD de uma matriz

Lista de símbolos

x_j	j-ésima componente do vetor x .
A_j	j-ésima coluna da matriz A .
$A(j, :)$	j-ésima linha da matriz A .
e	Vetor com todas as componentes iguais a um.
$x > 0$	Indica que as componentes do vetor x são positivas.
$x \geq 0$	Indica que as componentes do vetor x são não negativas.
A^T	Transposta da matriz A .
A^{-1}	Inversa da matriz A .
$\mathbb{R}^{m \times n}$	Conjunto das matrizes de ordem $m \times n$ com entradas reais.
\mathbb{R}_+^n	Conjunto dos vetores x pertencentes a \mathbb{R}^n que satisfazem $x \geq 0$.
\mathbb{R}_{++}^n	Conjunto dos vetores x pertencentes a \mathbb{R}^n que satisfazem $x > 0$.
$\ \cdot\ _p$	Norma-p de uma matriz ou vetor.
I_n	Matriz identidade de ordem $n \times n$.
λ_{max}	Maior autovalor de uma matriz A .
λ_{min}	Menor autovalor de uma matriz A .
$\kappa(A)$	Número de condição da matriz A segundo a norma euclidiana.
$X = \text{diag}(x)$	Matriz diagonal com as componentes do vetor $x \in \mathbb{R}^n$ na diagonal.
A^\dagger	Pseudo-inversa de Moore-Penrose

Sumário

1	INTRODUÇÃO	16
1.1	Problema Motivador e Objetivo	16
2	PROBLEMAS DE REGULARIZAÇÃO	18
2.1	Uma breve nota histórica	18
2.2	Regularização	20
2.2.1	Decomposição em valores singulares	20
2.2.2	Regularização por filtragem	21
2.2.3	Regularização de Tikhonov	23
2.3	Considerações finais	25
3	MÉTODOS DE PONTOS INTERIORES	26
3.1	Programação convexa	26
3.2	Uma estrutura primal-dual para regularização de Tikhonov	29
3.3	Um método de pontos interiores primal-dual para regularização de Tikhonov	36
3.3.1	Trajetória central	40
3.3.2	Etapas do método de pontos interiores primal-dual	42
3.3.3	Método preditor-corretor de Mehrotra	43
3.3.4	Etapas do MPI primal-dual na sua variante PCM	47
3.4	Cálculo da direção de busca	48
3.4.1	Direção de busca para o MPI primal-dual	48
3.4.2	Cálculo da direção de busca do PCM	50
4	SISTEMA DE EQUAÇÕES NORMAIS	53
4.1	Análise espectral do Sistema de Equações Normais	53
4.2	Abordagens para resolver o Sistema de Equações Normais	56
4.2.1	Método dos Gradientes Conjugados	56
4.2.2	Precondicionamento	57
4.2.3	Fatoração Controlada de Cholesky	59
4.2.4	Precondicionador Separador	62
5	RESULTADOS NUMÉRICOS	63
5.1	Hilbert	63
5.2	Baart	65
5.3	Blur	67

5.4	Shaw	69
5.5	Phillips	71
5.6	Problemas de teste: Foxgood, Gravity, Odftomo, Sphericaltomo e PRtomo	73
5.7	MPCLU vs SVD	76
6	CONCLUSÕES E PERSPECTIVAS FUTURAS	78
6.1	Conclusões	78
6.2	Perspectivas Futuras	79
	REFERÊNCIAS	80

1 Introdução

Quando os valores singulares da matriz dos coeficientes de um sistema $Ax = b$ se acumulam próximos à origem e decaem gradualmente a zero, isso torna a matriz severamente mal condicionada. Tais sistemas são frequentemente chamados de problemas lineares discretos e mal-postos (REZGHI; HOSSEINI, 2009).

Quando sistemas de equações são mal-postos, uma solução obtida com aritmética de precisão finita pode ser muito diferente da solução real, o que frequentemente se caracteriza por oscilações de grande amplitude nas componentes da solução x . Para contornar este problema foram desenvolvidas técnicas que permitem uma regularização da solução. A ideia é substituir o problema original, mal condicionado, por um problema “próximo”, melhor condicionado, tendo por objetivo ganhar com a redução nos erros provindos da aritmética de precisão finita, o que foi perdido na substituição do problema original por outro semelhante (PINA, 2010).

Praticamente todos os métodos de regularização para determinar soluções estáveis envolvem uma relação entre a “magnitude” da solução regularizada e a qualidade do ajuste que esta provê para os dados fornecidos. O que distingue os vários métodos de regularização é como estes medem e relacionam estas duas propriedades (HANSEN, 2000).

1.1 Problema Motivador e Objetivo

A discretização de um problema inverso comumente fornece um sistema linear de equações

$$Ax = b, \quad A \in \mathbb{R}^{m \times n}, \quad x \in \mathbb{R}^n, \quad b \in \mathbb{R}^m \quad (1.1.1)$$

com uma matriz mal-posta A . O cálculo de uma solução aproximada significativa do sistema linear (1.1.1) em geral requer que o sistema seja trocado por um sistema próximo que é menos sensível a perturbações. O método de regularização de Tikhonov é um dos mais antigos e mais populares métodos de regularização. Este método aproxima o sistema linear (1.1.1) pelo sistema regularizado

$$(A^T A + \alpha^2 I)x = A^T b, \quad (1.1.2)$$

onde $\alpha \geq 0$ é o parâmetro de regularização que determina a quantidade de regularização e I é o operador identidade.

Para qualquer $\alpha > 0$ fixo, o sistema (1.1.2) tem solução única

$$x_\alpha = (A^T A + \alpha^2 I)^{-1} A^T b. \quad (1.1.3)$$

Existem vários métodos iterativos para determinar valores aceitáveis do parâmetro de regularização α . Alguns destes podem ser encontrados em (BJORCK, 1988).

Note que a solução (1.1.3) de (1.1.2) satisfaz $x_\alpha \rightarrow x_0 = A^\dagger b$ com $\alpha \rightarrow 0^+$, onde A^\dagger denota a pseudo-inversa de Moore-Penrose de A , veja (MEYER, 2010). Em muitos problemas, a matriz A tem muitos valores singulares pequenos, o que acaba acarretando um mal condicionamento da matriz A e conseqüentemente de $A^T A$, visto que os autovalores de $A^T A$ são os valores singulares de A elevados ao quadrado. Além disso, em geral o vetor b está contaminado por erros de medidas (ruídos). Portanto, a solução x_0 em geral tem componentes grandes. Na Subseção 2.2.3 podemos observar que a solução (1.1.3) satisfaz o problema de minimização

$$\min_x \|x\|_2, \text{ sujeito a } \|b - Ax\|_2 \leq \mathbf{E}_0 \quad (1.1.4)$$

onde \mathbf{E}_0 é o valor da norma do resíduo que estamos dispostos a aceitar e $\|\cdot\|_2$ denota a norma Euclidiana. O objetivo deste trabalho é desenvolver métodos de pontos interiores que sejam capazes de obter uma solução para o problema de minimização (1.1.4), em especial quando o sistema $Ax = b$ é de grande porte.

Esta tese está organizada do seguinte modo: no Capítulo 2, apresentamos uma visão geral dos problemas de regularização. Os métodos de pontos interiores desenvolvidos para o problema de regularização de Tikhonov, encontram-se no Capítulo 3. No Capítulo 4 apresentamos técnicas para resolução dos sistemas de equações normais gerados pelos métodos de pontos interiores desenvolvidos no Capítulo 3. Os resultados obtidos pelos experimentos numéricos estão apresentados no Capítulo 5, e por fim, no Capítulo 6 são apresentados as conclusões e propostas para trabalhos futuros.

2 Problemas de Regularização

Segundo (VOGEL, 2002), não é incomum que a discretização de um problema inverso linear leve a um sistema linear mal-posto. Ainda segundo (VOGEL, 2002), em geral, o sistema linear obtido pode ser regularizado para tornar possível o cálculo de uma solução aproximada significativa. Um parâmetro de regularização especifica a quantidade de regularização, no entanto, um valor apropriado deste parâmetro não é conhecido *a priori*. Na próxima seção baseado em (HANSEN, 2000) traçaremos uma breve linha histórica de como o problema foi e é abordado, sendo uma dessas abordagens o método de regularização de Tikhonov clássico, na seção subsequente apresentaremos algumas técnicas seguidas de exemplos.

2.1 Uma breve nota histórica

Em (RILEY, 1955), temos a descrição do que pode ser o primeiro esquema equivalente a regularização de Tikhonov. Neste trabalho, Riley abordou sistemas simétricos definidos positivos mal condicionados, $Ax = b$ e propôs resolver o sistema $(A + \alpha I)x = b$, no lugar do sistema original, onde α é uma constante positiva pequena.

O primeiro artigo a abordar problemas mais gerais foi (PHILLIPS, 1962). Nesse, A é uma matriz quadrada obtida de uma Equação Integral de Fredholm por meio de uma regra de quadratura. Nesse artigo, sem usar notação matricial, Phillips propôs o cálculo de uma solução regularizada.

S. Twomey, em 1963 reformulou as expressões de Phillips para x_α por meio de equações normais regularizadas e obteve a expressão bem conhecida

$$x_\alpha = (A^T A + \alpha^2 L^T L)^{-1} A^T b,$$

com $L = \text{tridiag}(1, -2, 1)$. Detalhes desta formulação podem ser encontrados em (TWO-MEY, 1963). Ele também propôs a inclusão de uma estimativa a priori x_0 , mas somente em conexão com a escolha $L = I$ (a matriz identidade), levando à expressão

$$x_\alpha = (A^T A + \alpha^2 I)^{-1} (A^T b + \alpha^2 x_0).$$

Ainda em 1963, A. N. Tikhonov propôs uma formulação muito mais geral, veja (TIKHONOV, 1963). Ele considerou o problema $Kf = g$, onde f e g são funções e K é um operador integral. Tikhonov propôs a formulação $f_\alpha = \arg \min \{ \|Kf - g\|_2^2 + \alpha^2 \Omega(f) \}$ com o funcional

$$\Omega(f) = \int_a^b (v(s)f(s)^2 + w(s)f'(s)^2) ds,$$

onde v e w são funções de peso positivas. Recorrendo à modelagem numérica e computacional, Tikhonov usou a técnica de quadratura de ponto médio para chegar ao problema

$$\min\{\|Ax - b\|_2^2 + \alpha^2(\|D_v^{1/2}x\|_2^2 + \|LD_w^{1/2}x\|_2^2)\},$$

em que D_v e D_w são matrizes diagonais de peso correspondentes a v e w , e $L = \text{bidiag}(-1, 1)$. Via equações normais regularizadas ele obteve a expressão

$$x_\alpha = (A^T A + \alpha^2(D_v + L^T D_w L))^{-1} A^T b.$$

Em (GOLUB, 1965) é proposto resolver

$$x_\alpha = \arg \min \{ \|Ax - b\|_2^2 + \alpha^2 \|L(x - x_0)\|_2^2 \},$$

via a formulação de quadrados mínimos, isto é,

$$x_\alpha = \arg \min \left\| \begin{pmatrix} A \\ \alpha L \end{pmatrix} x - \begin{pmatrix} b \\ \alpha L x_0 \end{pmatrix} \right\|_2 \quad (2.1.1)$$

e a fatoração QR da matriz de coeficientes associada. Golub propôs esta abordagem em conexão com o esquema de (RILEY, 1955).

Ao usar a formulação de regularização de Tikhonov em um conjunto estocástico, Joel Franklin chegou a um sistema de “equações normais regularizado”, detalhes podem ser encontrados em (FRANKLIN, 1970). Já no artigo (FRANKLIN, 1978) é proposto a variante $x_\alpha = (A + \alpha B)^{-1} b$, com A e B sendo simétricas e positiva semi-definidas, e α um escalar positivo, o que se conecta novamente ao artigo de Riley de 1955.

Na literatura estatística, a regularização de Tikhonov é conhecida como uma técnica de regressão e parece datar dos artigos (HOERL; KENNARD, 1970b) e (HOERL; KENNARD, 1970a). (MARQUARDT, 1970) usou esta abordagem como uma base para análise de seus métodos iterativos de 1963 para resolver problemas de quadrados mínimos não lineares, onde é incorporado a forma padrão da regularização de Tikhonov em cada passo.

Uma forma de calcular soluções de Tikhonov x_α para um conjunto de parâmetros de regularização α (que é quase sempre o caso na prática) é por meio da bidiagonalização de Lars Eldén, descrita em (ELDÉN, 1977), combinada com uma transformação a forma padrão, se $L \neq I$. Mais recentemente, vários outros algoritmos foram desenvolvidos, a exemplo disto podemos citar (FROMMER; MAASS, 1999), (GOLUB; MATT, 1991) e (HANKE; VOGEL, 1999).

2.2 Regularização

A fim de familiarizar o leitor com o conceito de regularização, nesta seção apresentamos algumas técnicas de regularização simples seguidas de exemplos. Para tanto, começamos por apresentar de forma sucinta uma das principais ferramentas de resolução e análise de problemas de regularização, a decomposição SVD, mais detalhes sobre esta decomposição podem ser encontrados em (GOLUB; LOAN, 1996).

Todos os resultados numéricos apresentados nesta seção foram desenvolvidos em MATLAB R2013b com sistema operacional 64-bit Windows 10, processador Intel Core I7-8550U, 1.99 Ghz, 16 GB de memória RAM.

2.2.1 Decomposição em valores singulares

A decomposição em valores singulares de uma matriz $A \in \mathbb{R}^{m \times n}$ é uma decomposição da forma

$$A = U_{m \times m} \Sigma_{m \times n} V_{n \times n}^T = \sum_{i=1}^n s_i u_i v_i^T, \quad (2.2.1)$$

em que os vetores singulares $u_i \in \mathbb{R}^m$ são ortonormais entre si e compõem as colunas da matriz U , por sua vez os vetores singulares $v_i \in \mathbb{R}^n$ são ortonormais entre si e compõem as colunas da matriz V . Os valores singulares s_i são valores não negativos que aparecem em ordem decrescente,

$$s_1 \geq s_2 \geq s_3 \geq \dots, \geq s_n \geq 0,$$

e correspondem aos valores da diagonal principal da matriz Σ . Para matrizes A provenientes da discretização de problemas inversos, os valores singulares decaem gradualmente à zero, o que dificulta sua aplicação direta na resolução desses problemas.

Neste trabalho, assumiremos que $m \geq n$ e que os erros no problema dado

$$\min \|Ax - b\|_2 \quad (2.2.2)$$

estão restritos ao lado direito, isto é, dado b podemos escrever

$$b = \bar{b} + \mathbf{e} \quad \bar{b} = A\bar{x},$$

onde \bar{b} representa os dados exatos não perturbados, $\bar{x} = A^\dagger \bar{b}$ representa a solução exata, e o vetor \mathbf{e} representa os erros nos dados.

Obter a solução do problema (2.2.2) consiste em determinar o vetor $x \in \mathbb{R}^n$ que minimize o valor da norma residual dada por:

$$\|\mathbf{r}\|_2^2 = \|b - Ax\|_2^2 = x^T A^T A x - 2x^T A^T b + b^T b. \quad (2.2.3)$$

Derivando (2.2.3) em relação a x e igualando a zero chegamos à relação

$$A^T A x = A^T b \quad (2.2.4)$$

que é conhecida como sistema de equações normais. Se inserirmos a decomposição em valores singulares $A = U\Sigma V^T$ no sistema de equações normais (2.2.4), então a sequência de operações

$$\begin{aligned} c &= U^T b \\ \Sigma \Sigma^T y &= \Sigma c \\ x &= V y. \end{aligned}$$

nos fornecerá o vetor x_{LS} que minimiza o resíduo (2.2.3). De forma mais explícita podemos escrever

$$x_{LS} = \sum_{i=1}^n \frac{u_i^T b}{s_i} v_i. \quad (2.2.5)$$

Nesta etapa, é necessário assumirmos a chamada Condição Discreta de Picard: *Os coeficientes exatos da SVD, $|u_i^T \bar{b}|$, decaem mais rápido do que $|s_i|$.* Esta condição assegura que a solução de quadrados mínimos $\bar{x} = A^\dagger \bar{b}$ do problema não perturbado não tem norma grande, porque os coeficientes da solução exata $|v_i^T \bar{x}| = |u_i^T \bar{b}/s_i|$ também decaem. Esta Condição Discreta de Picard nos assegura que existe uma solução fisicamente mensurável para os problemas inversos em consideração e também assegura que a solução pode ser aproximada por uma solução regularizada. Detalhes sobre esta condição podem ser encontrados em (HANSEN, 1990) e (HANSEN, 1998).

2.2.2 Regularização por filtragem

A fim de exemplificar uma aplicação útil da decomposição em valores singulares na obtenção da regularização por filtragem da solução de um sistema de equações lineares mal condicionado, seguimos (VOGEL, 2002) e consideramos o sistema mal condicionado $Ax = b$ em que A é a matriz de Hilbert, isto é, $a_{ij} = 1/(i + j - 1)$. A verificação do mal condicionamento das matrizes de Hilbert pode ser encontrado em (HOANG; RAMM, 2008). Consideramos também que A tem ordem $n = 100$ e b é construído de modo que a solução seja $\bar{\mathbf{x}} = (1 \ 1 \ \dots \ 1)^T$. Se \mathbf{x} denota a solução do sistema $A^T A x = A^T b$ obtida pela decomposição em valores singulares $A = U\Sigma V^T$, temos que $\|\bar{\mathbf{x}} - \mathbf{x}\|_2 = 49,6895$. É bem

evidente que a solução obtida está muito longe da solução correta. A razão disto é que a divisão por valores singulares extremamente pequenos tem o efeito de ampliar os erros de arredondamento cometidos no processo de cálculo. Devido a isso, cogitamos que seja propício a filtragem de valores singulares extremamente pequenos. Adotamos o seguinte critério de filtragem: Fixamos uma tolerância ζ , com $0 < \zeta < 1$ e executamos os passos do Algoritmo 1.

Algoritmo 1 – Regularização por filtragem

Dados: U, Σ, V, b

Resultado: Solução regularizada por filtragem do sistema $A^T A x = A^T b$

$c = U^T b$;

para $i = 1 : n$ **faça**

se $s_i < \zeta s_1$ **então**

| $y_i = 0$

senão

| $y_i = c_i/s_i$;

fim

fim

$x = Vy$.

Este procedimento trunca os valores singulares que estão abaixo do patamar determinado pelo valor ζ . Repetindo os cálculos para $\zeta = 10^{-5}$, o que equivale a filtrar alguns dos menores valores singulares, obtemos uma solução \mathbf{x}_ζ , com $\|\bar{\mathbf{x}} - \mathbf{x}_\zeta\|_2 = 0,0246$ e $\|\mathbf{r}\|_2 = \|b - A\mathbf{x}_\zeta\|_2 = 1,9406 \times 10^{-7}$, onde \mathbf{r} denota o resíduo obtido pelo processo de filtragem. Tendo em vista o mal condicionamento da matriz de Hilbert e o erro cometido em relação a solução real, podemos considerar a solução obtida como aceitável. O valor de tolerância ζ deve ser fixado com cuidado, visto que um valor baixo demais não filtra o suficiente, ao passo que um valor alto demais filtra excessivamente.

Este procedimento de filtragem desconsidera a solução explícita obtida pela decomposição em valores singulares e toma como solução aquela obtida por um fator de filtragem f_i , ou seja, toma a solução filtrada \mathbf{x}_f dada por

$$\mathbf{x}_f = \sum_{i=1}^n f_i \frac{u_i^T b}{s_i} v_i.$$

A solução não filtrada corresponde a tomar $f_i = 1$ para todas as componentes. Essa filtragem efetuada por truncagem pressupõe que é possível demarcar com nitidez a fronteira entre os valores singulares e determinar aqueles que devem ser excluídos, o que nem sempre é o caso. No que segue, estabelecemos outros fatores de filtragem que em alguns casos podem sobrepujar esta dificuldade.

2.2.3 Regularização de Tikhonov

Uma solução irregular \mathbf{x} está com frequência associada a um valor elevado de $\|\mathbf{x}\|_2$. O método apresentado em (TIKHONOV; ARSEININ, 1977) tem por objetivo obter soluções do sistema de equações lineares $Ax = b$ com norma pequena, do que é proposto resolver o problema de minimização:

$$\min_x \|x\|_2, \text{ sujeito a } \|b - Ax\|_2 \leq \mathbf{T}_0$$

onde \mathbf{T}_0 é o valor da norma do resíduo que estamos dispostos a aceitar. Sendo assim, estamos diante de um problema de minimização com restrições de desigualdade.

Via o método dos multiplicadores de Lagrange, podemos reescrever este problema como um problema de minimização sem restrições, dada por

$$L(x, \lambda) = \|x\|_2^2 + \lambda(\|b - Ax\|_2^2 - \mathbf{T}_0^2)$$

onde $\lambda > 0$ é o multiplicador de Lagrange. Derivando e igualando a zero, obtemos as condições necessárias para o mínimo,

$$\begin{aligned} (A^T A + \alpha^2 I)x &= A^T b \\ \|b - Ax\|_2 &= \mathbf{T}_0 \end{aligned} \quad (2.2.6)$$

com $\alpha^2 = 1/\lambda$. Se inserirmos a decomposição SVD na formulação de quadrados mínimos (2.2.6) então podemos provar que a solução de Tikhonov é dada por

$$x_\alpha = \sum_{i=1}^n f_i \frac{u_i^T b}{s_i} v_i, \quad (2.2.7)$$

onde f_1, \dots, f_n são os fatores de filtragem de Tikhonov, que dependem de s_i e α com

$$f_i = \frac{s_i^2}{s_i^2 + \alpha^2} \simeq \begin{cases} 1, & s_i \gg \alpha \\ \frac{s_i^2}{\alpha^2}, & s_i \ll \alpha \end{cases} \quad (2.2.8)$$

Em particular, a solução de mínimos quadrados x_{LS} é dada por (2.2.7) com $\alpha = 0$ e todos os fatores de filtragem iguais a um. Comparando x_α com x_{LS} vemos que os fatores de filtragem, na prática, filtram as entradas de x_α correspondentes a valores singulares pequenos, enquanto deixam as componentes correspondentes a valores singulares

grandes quase inalterados. O vetor residual correspondente a x_α é chamado desajuste e é dado em termos da SVD por

$$b - Ax_\alpha = \sum_{i=1}^n (1 - f_i) u_i^T b u_i + b_0, \quad (2.2.9)$$

onde o vetor $b_0 = b - \sum_{i=1}^n u_i u_i^T b$ é a componente de b que se encontra fora da imagem de A , e $1 - f_i = \alpha^2 / (s_i^2 + \alpha^2)$. Note que $b_0 = 0$ quando $m = n$.

Munidos das expressões (2.2.7) e (2.2.9) nós podemos escrever a solução e as normas residuais em termos da SVD:

$$\|x_\alpha\|_2^2 = \sum_{i=1}^n \left(f_i \frac{u_i^T b}{s_i} \right)^2 \quad (2.2.10)$$

$$\|Ax_\alpha - b\|_2^2 = \sum_{i=1}^n ((1 - f_i) u_i^T b)^2. \quad (2.2.11)$$

Estas expressões formam a base para determinação do parâmetro α , através do método da L-curva, veja (HANSEN, 2000), o qual exemplificamos a seguir.

Voltando ao exemplo da matrizes de Hilbert, se construirmos um gráfico em escalas logarítmicas decimais, de $\|x\|_2$ e $\|r\|_2$ tomando como parâmetro α , obtemos a curva da Figura 1.

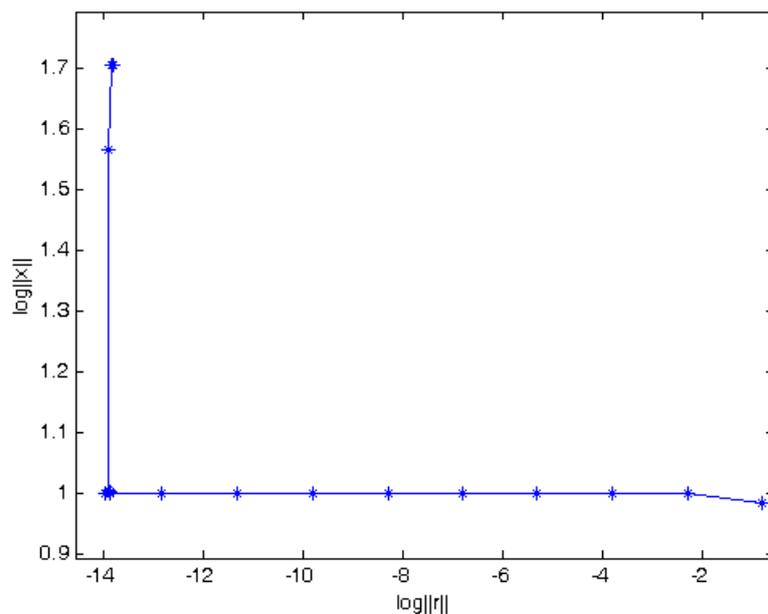


Figura 1 – Gráfico de $\|x\|_2$ em função de $\|r\|_2$

A forma em L desta curva é comum nesta classe de problemas e mostra que um bom resultado é conseguido tomando o ponto do canto do L já que esta corresponde a um resíduo baixo e um valor de $\|\mathbf{x}\|_2$ razoável. À esquerda deste ponto o valor de $\|\mathbf{x}\|_2$ sobe sem que tenhamos uma diminuição significativa de $\|\mathbf{r}\|_2$; à direita é $\|\mathbf{r}\|_2$ que cresce sem que tenhamos uma diminuição significativa de $\|\mathbf{x}\|_2$. O valor de α no canto de L é $\approx 10^{-11}$ com $\mathbf{T}_0 = \|\mathbf{r}\|_2 = 1,3965 \times 10^{-14}$, $\|\bar{\mathbf{x}} - \mathbf{x}_\alpha\|_2 = 4,1908 \times 10^{-5}$, e $\|\mathbf{x}_\alpha\|_2 \approx 10$.

2.3 Considerações finais

Outros métodos para resolução de problemas de regularização, não baseados na decomposição SVD, podem ser encontrados em (GOLUB; MATT, 1997), (CALVETTI et al., 2000) e (FROMMER; MAASS, 1999).

É importante ressaltar que o exemplo trabalhado nas Seções 2.2.2 e 2.2.3, está livre da presença de ruído no vetor b , caso houvesse a presença de ruído, os resultados obtidos não seriam tão satisfatórios. Ao analisar este exemplo, torna-se evidente que o problema do método de Tikhonov está no custo computacional e numérico envolvido no cálculo da decomposição SVD da matriz A , cálculo inviável no caso de matrizes de grande porte. Há também a dificuldade na escolha de um valor de α adequado para determinação de uma melhor solução regularizada.

3 Métodos de Pontos Interiores

O Método de Pontos Interiores (MPI) foi desenvolvido para resolver problemas de programação linear (WRIGHT, 1997). Com o decorrer dos anos, tem surgido muitas variantes deste método, dentre os quais, os mais conhecidos são os Métodos de Pontos Interiores primal-dual, desenvolvidos por (MEGIDDO, 1989). Embora uma iteração de um MPI primal-dual seja computacionalmente cara, a otimalidade é atingida após poucas iterações, o que torna o método atrativo. Além disso, eles tem melhores propriedades teóricas para análise de complexidade do pior caso, como pode ser observado em (MONTEIRO; ADLER, 1989) e (ZHANG; TAPIA, 1992). A maioria dos MPI encontrados na literatura podem ser vistos como uma variante do método de Newton. Também é importante ressaltar que na teoria de métodos de pontos interiores, as funções de barreira logarítmica possuem atributos convenientes, alguns dos quais utilizaremos em nosso método de pontos interiores.

3.1 Programação convexa

Para desenvolvermos o Método de Pontos Interiores (MPI) primal-dual, precisamos estar familiarizados com algumas definições e resultados importantes relacionados a problemas de programação convexa. No que segue, baseados em (BOYD; VANDENBERGHE, 1994), apresentamos estes resultados.

A forma mais geral de um problema de otimização é dada por:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.a.} \quad & h(x) = 0_m \\ & g(x) \leq 0_p, \end{aligned} \tag{3.1.1}$$

em que $x \in \mathbb{R}^n$ e as funções f, h_i, g_j são diferenciáveis para $i = 1, \dots, m, j = 1, \dots, p$, sendo $h = (h_1, h_2, \dots, h_m), g = (g_1, g_2, \dots, g_p)$.

Definição 1 (Solução ótima local). *Seja x^* um ponto factível do problema (3.1.1), ou seja, $x^* \in \mathcal{S} \cap \mathcal{G}$ onde $\mathcal{S} = \{x \in \mathbb{R}^n : h(x) = 0_m\}$ e $\mathcal{G} = \{x \in \mathbb{R}^n : g(x) \leq 0_p\}$. O ponto x^* é denominado solução ótima local de (3.1.1) se existe um escalar $\delta > 0$ tal que $f(x^*) \leq f(x)$ para todo x com $\|x - x^*\| < \delta$.*

Definição 2 (Direção factível). *Um vetor $d \in \mathbb{R}^n$ é uma direção factível no ponto x , onde $x \in \mathcal{S} \cap \mathcal{G}$, se existe $\bar{\alpha} \in \mathbb{R}_{++}$ tal que $x + \alpha d \in \mathcal{S} \cap \mathcal{G}$ para todo $\alpha \in [0, \bar{\alpha}]$.*

Definição 3 (Ponto regular). Um ponto $x^* \in \mathcal{S} \cap \mathcal{G}$ é um ponto regular se o conjunto

$$\{\nabla h_j(x^*) : j = 1, \dots, m\} \cup \{\nabla g_i(x^*) : i \in \mathcal{I}(x^*)\}$$

é linearmente independente, em que $\mathcal{I}(x^*) = \{i : g_i(x^*) = 0\}$.

Teorema 1 (Condições de Karush-Khun-Tucker (KKT)). Se o valor x^* é um ponto regular e solução local de (3.1.1), então existem $y^* \in \mathbb{R}^m$ e $z^* \in \mathbb{R}_+^p$ tais que:

$$\nabla_x \ell(x^*, y^*, z^*) = 0_n, \quad (3.1.2)$$

$$z_i^* g_i(x^*) = 0 \text{ para } i = 1, \dots, p, \quad (3.1.3)$$

onde

$$\ell(x, y, z) = f(x) + \sum_{j=1}^m y_j h_j(x) + \sum_{i=1}^p z_i g_i(x).$$

Uma demonstração das condições de KKT pode ser encontrada em (BRINKHUIS; TIKHOMIROV, 2011).

A função $\ell : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}_+^p \rightarrow \mathbb{R}$ definida no Teorema 1 é chamada função Lagrangeana associada ao problema (3.1.1).

Definição 4 (Conjunto Convexo). Um conjunto $\Omega \subset \mathbb{R}^n$ é dito convexo se e somente se, para todo $x, y \in \Omega$, $\lambda \in [0, 1]$, temos que $\lambda x + (1 - \lambda)y \in \Omega$, ou seja, para quaisquer dois pontos em Ω , o segmento que os une está contido em Ω .

Definição 5 (Função Convexa). Dada uma função f definida em um conjunto convexo Ω . Dizemos que f é convexa se e somente se, para todo $x, y \in \Omega$, $\lambda \in [0, 1]$, tem-se

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y).$$

Se para todo $\lambda \in (0, 1)$ e $x \neq y$ vale a desigualdade estrita, dizemos que f é estritamente convexa. Se $-f$ é convexa, dizemos que f é côncava.

Definição 6 (Problema de programação convexa). O problema

$$\begin{aligned} \min \quad & f(x) \\ \text{s.a.} \quad & h(x) = 0_m \\ & g(x) \leq 0_p, \end{aligned}$$

é chamado de problema de programação convexa se f é uma função convexa, as funções componentes de h são afins, ou seja, funções do tipo $a^T x + b_i$ com $a \in \mathbb{R}^n$ e $b_i \in \mathbb{R}$, e as funções componentes de g são convexas. Neste caso, o conjunto de pontos factíveis $\mathcal{S} \cap \mathcal{G}$ é um conjunto convexo, em que $\mathcal{S} = \{x \in \mathbb{R}^n : h(x) = 0_m\}$ e $\mathcal{G} = \{x \in \mathbb{R}^n : g(x) \leq 0_p\}$.

Teorema 2 (Suficiência das condições de KKT). *No problema (3.1.1) admita que:*

1. $h_j, j = 1, \dots, m$ são funções afins (\mathcal{S} é convexo);
2. $g_i, i = 1, \dots, p$ são funções convexas (\mathcal{G} é convexo);
3. f é convexa;

isto é, o problema (3.1.1) é de programação convexa. Se $x^* \in \mathcal{S} \cap \mathcal{G}$ é um ponto regular e satisfaz as condições KKT dadas em (3.1.2) e (3.1.3), então x^* é um mínimo global de f em $\mathcal{S} \cap \mathcal{G}$, onde $\mathcal{S} = \{x \in \mathbb{R}^n : h(x) = 0_m\}$ e $\mathcal{G} = \{x \in \mathbb{R}^n : g(x) \leq 0_p\}$.

Este teorema corresponde ao Teorema A.2 (i) de (WRIGHT, 1997), onde pode ser encontrado sua demonstração.

É importante ressaltar que em problemas de programação convexa, toda solução local é também solução global, como explicitado no seguinte teorema:

Teorema 3. *Suponha que*

$$\begin{aligned} \min \quad & f(x) \\ \text{s.a.} \quad & h(x) = 0_m \\ & g(x) \leq 0_p, \end{aligned} \tag{3.1.4}$$

com $x \in \mathbb{R}^n$, $h = (h_1, h_2, \dots, h_m)$ e $g = (g_1, g_2, \dots, g_p)$, sendo f, h_i, g_j funções diferenciáveis para $i = 1, \dots, m, j = 1, \dots, p$, seja um problema de programação convexa, veja Definição 6. O vetor x^* é ótimo local de (3.1.4) se, e somente se, x^* é um ótimo global de (3.1.4), isto é, $f(x^*) \leq f(x)$ para todo $x \in \mathcal{S} \cap \mathcal{G}$, onde $\mathcal{S} = \{x \in \mathbb{R}^n : h(x) = 0_m\}$ e $\mathcal{G} = \{x \in \mathbb{R}^n : g(x) \leq 0_p\}$.

A demonstração deste teorema de pode ser encontrada em (IZMAILOV; SOLODOV, 2009). Por este teorema temos que x^* é uma solução ótima global de um problema de programação convexa se, e somente se, as condições de KKT são satisfeitas. Mais adiante veremos como os métodos de pontos interiores usam uma variação do método de Newton nestas condições para encontrar a direção de busca.

3.2 Uma estrutura primal-dual para regularização de Tikhonov

Como mencionado na Subseção 2.2.3, o problema de regularização de Tikhonov, pode ser pensado como o problema de minimização

$$\min_x \|x\|_2, \text{ sujeito a } \|b - Ax\|_2 \leq \mathbf{T}_0 \quad (3.2.1)$$

Uma vez que o objetivo é obter soluções do sistema de equações lineares $Ax = b$, com norma pequena. Nós aproximamos o problema (3.2.1) pelo problema (3.2.2), que procura soluções com propriedades similares as requeridas pelo problema (3.2.1):

$$(P) \begin{cases} \min_x & \frac{\tau}{2}\|x\|_2^2 + \|u\|_1 + \|v\|_1 \\ \text{sujeito a} & Ax + u - v = b. \\ & (u, v) \geq 0 \text{ e } x \in \mathbb{R}^n. \end{cases} \quad (3.2.2)$$

ou ainda,

$$(P) \begin{cases} \min_x & \frac{\tau}{2}\|x\|_2^2 + e^T(u + v) \\ \text{sujeito a} & Ax + u - v = b. \\ & (u, v) \geq 0 \text{ e } x \in \mathbb{R}^n. \end{cases} \quad (3.2.3)$$

Trata-se de um problema de programação quadrática com restrições lineares, onde A é uma matriz de posto completo $m \times n$, b e x são vetores colunas de dimensões apropriadas. Além disso, e é um vetor com todas as entradas iguais a um e u e v são variáveis não negativas. O valor $\tau > 0$ representa um parâmetro de penalização para valores grandes de $\|x\|_2$.

Associado ao problema (3.2.3), temos o problema de programação quadrática dual:

$$(D) \begin{cases} \max_x & -\frac{\tau}{2}\|x\|^2 + y^T b \\ \text{sujeito a} & \tau x - A^T y = 0 \\ & e - y - z = 0 \\ & e + y - w = 0 \\ & (z, w) \geq 0 \text{ e } y \in \mathbb{R}^m. \end{cases} \quad (3.2.4)$$

O problema (3.2.3) é também um problema de programação convexa. Logo poderemos fazer uso dos resultados da seção anterior. Nesta seção, baseados em (BERTSIMAS; TSITSIKLIS, 1997), adaptamos para (3.2.3) e (3.2.4) alguns aspectos da Programação Linear (PL) que relacionam as formulações primal (P) e dual (D). Isto posto, dizemos que um vetor (x, u, v) pertencente ao conjunto

$$\mathcal{P} = \{(x, u, v) : Ax + u - v = b, (u, v) \geq 0\}$$

é um ponto primal factível. De forma semelhante, dizemos que um vetor (x, y, z, w) que pertence ao conjunto

$$\mathcal{D} = \{(x, y, z, w) : e - y - z = 0, e + y - w = 0, \tau x - A^T y = 0, (z, w) \geq 0\}$$

é um ponto dual factível. É importante ressaltar que a formulação primal ou dual de um problema de otimização é equivalente às formulações (3.2.3) e (3.2.4) de acordo com a Definição 7.

Definição 7. *Dois problemas de otimização*

$$(P1) \begin{cases} \min & F(x) \\ \text{s.a.} & x \in \mathcal{P}_1 \end{cases} \quad (P2) \begin{cases} \min & G(y) \\ \text{s.a.} & y \in \mathcal{P}_2 \end{cases}$$

são equivalentes se existir um ponto $y \in \mathcal{P}_2$, para cada $x \in \mathcal{P}_1$ tal que $F(x) = G(y)$, e de modo análogo, para cada $x \in \mathcal{P}_1$ existir $y \in \mathcal{P}_2$ tal que $G(y) = F(x)$.

Sendo assim, se $(P1)$ e $(P2)$ forem equivalentes e $(P1)$ possuir solução ótima com valor α , então $(P2)$ também possuirá solução ótima com valor α . A recíproca desta afirmação é também verdadeira.

Um problema de otimização como $(P1)$ é dito ilimitado quando para todo $\varepsilon \in \mathbb{R}$ existe $x \in \mathcal{P}_1$ tal que $F(x) < \varepsilon$. Além disso, $(P1)$ é dito infactível se o conjunto $\mathcal{P}_1 = \emptyset$. As mesmas considerações são válidas para o problema de otimização $(P2)$.

Estabelecidos esses conceitos, se $(P1)$ e $(P2)$ são equivalentes, então $(P1)$ é ilimitado se, e somente se, $(P2)$ é ilimitado. Outrossim, $(P1)$ é um infactível se, e somente se, $(P2)$ é infactível.

A fim de estabelecer uma relação entre os problemas primal (P) e dual (D) , exploramos um pouco da teoria de dualidade, começando pelo Teorema Dualidade Fraca.

Teorema 4 (Dualidade Fraca). *Se (x, u, v) é um ponto primal factível e (x, y, z, w) é um ponto dual factível, então*

$$-\frac{\tau}{2} \|x\|_2^2 + b^T y \leq \frac{\tau}{2} \|x\|_2^2 + e^T (u + v).$$

Demonstração. Com efeito, como $u^T z \geq 0$ e $v^T w \geq 0$, temos que:

$$\begin{aligned}
-\frac{\tau}{2}\|x\|_2^2 + b^T y &= -\frac{\tau}{2}\|x\|_2^2 + (Ax + u - v)^T y \\
&= -\frac{\tau}{2}\|x\|_2^2 + x^T A^T y + u^T y - v^T y \\
&= -\frac{\tau}{2}\|x\|_2^2 + \tau\|x\|_2^2 + u^T(e - z) - v^T(w - e) \\
&= \frac{\tau}{2}\|x\|_2^2 + u^T e - u^T z - v^T w + v^T e \\
&\leq \frac{\tau}{2}\|x\|_2^2 + e^T(u + v)
\end{aligned}$$

□

Apesar da simplicidade deste resultado, este já nos permite obter algumas conclusões relativas à relação entre os problemas primal e dual.

Corolário 1. *Se o problema primal (P) tem conjunto de pontos factíveis ilimitado, então o problema dual (D) é infactível. Analogamente, se o problema dual (D) tem conjunto de pontos factíveis ilimitado, então o problema primal (P) é infactível.*

Demonstração. Suponha que o problema dual é factível, isto é, existe $(x, \bar{y}, \bar{z}, \bar{w}) \in \mathcal{D}$. Então, pelo Teorema Dualidade Fraca:

$$-\frac{\tau}{2}\|x\|_2^2 + b^T \bar{y} \leq \frac{\tau}{2}\|x\|_2^2 + e^T(u + v), \text{ para todo } (x, u, v) \in \mathcal{P}.$$

Não obstante, da hipótese que \mathcal{P} é ilimitado, segue que para todo $\delta \in \mathbb{R}$ existe $(x, u, v) \in \mathcal{P}$ tal que $\frac{\tau}{2}\|x\|_2^2 + e^T(u + v) < \delta$. Em particular, se tomarmos, $\delta = -\frac{\tau}{2}\|x\|_2^2 + b^T \bar{y}$, existe $(x, u, v) \in \mathcal{P}$, tal que:

$$\frac{\tau}{2}\|x\|_2^2 + e^T(u + v) < -\frac{\tau}{2}\|x\|_2^2 + b^T \bar{y},$$

o que é um absurdo. Portanto, o problema dual (D) é infactível. Reciprocamente, suponhamos que o problema primal (P) é factível, ou seja, existe $(x, \bar{u}, \bar{v}) \in \mathcal{P}$. Logo pelo teorema fraco da dualidade,

$$-\frac{\tau}{2}\|x\|_2^2 + b^T y \leq \frac{\tau}{2}\|x\|_2^2 + e^T(\bar{u} + \bar{v}), \text{ para todo } (x, y, z, w) \in \mathcal{D}.$$

Como o problema Dual é ilimitado, para todo $\epsilon \in \mathbb{R}$ existe $(x, y, z, w) \in \mathcal{D}$ tal que $\epsilon < -\frac{\tau}{2}\|x\|_2^2 + b^T y$, em particular, se $\epsilon = \frac{\tau}{2}\|x\|_2^2 + e^T(\bar{u} + \bar{v})$, existe um (x, y, z, w) , tal que:

$$\frac{\tau}{2}\|x\|_2^2 + e^T(\bar{u} + \bar{v}) < -\frac{\tau}{2}\|x\|_2^2 + b^T\bar{y}$$

novamente uma contradição. Por conseguinte, (P) é infactível.

□

Corolário 2. *Sejam (x, u, v) um ponto primal factível e (x, y, z, w) um ponto dual factível que satisfazem*

$$-\frac{\tau}{2}\|x\|_2^2 + b^T y = \frac{\tau}{2}\|x\|_2^2 + e^T(u + v),$$

então (x, u, v) e (x, y, z, w) são soluções ótimas de seus respectivos problemas.

Demonstração. Seja $(\hat{x}, \hat{u}, \hat{v}) \in \mathcal{P}$ um ponto primal factível arbitrário, pelo Teorema Dualidade Fraca tem-se:

$$-\frac{\tau}{2}\|x\|_2^2 + b^T y \leq \frac{\tau}{2}\|\hat{x}\|_2^2 + e^T(\hat{u} + \hat{v}). \quad (3.2.5)$$

Como

$$-\frac{\tau}{2}\|x\|_2^2 + b^T y = \frac{\tau}{2}\|x\|_2^2 + e^T(u + v), \quad (3.2.6)$$

segue de (3.2.5) que

$$\frac{\tau}{2}\|x\|_2^2 + e^T(u + v) \leq \frac{\tau}{2}\|\hat{x}\|_2^2 + e^T(\hat{u} + \hat{v}),$$

para todo $(\hat{x}, \hat{u}, \hat{v}) \in \mathcal{P}$, ou seja, (x, u, v) é solução ótima do problema primal.

De modo análogo, seja $(\hat{x}, \hat{y}, \hat{z}, \hat{w}) \in \mathcal{D}$ um ponto dual factível arbitrário, logo pelo teorema fraco da dualidade temos:

$$-\frac{\tau}{2}\|\hat{x}\|_2^2 + b^T \hat{y} \leq \frac{\tau}{2}\|x\|_2^2 + e^T(u + v). \quad (3.2.7)$$

Da hipótese (3.2.6) e da desigualdade acima

$$-\frac{\tau}{2}\|\hat{x}\|_2^2 + b^T \hat{y} \leq -\frac{\tau}{2}\|x\|_2^2 + b^T y.$$

Portanto, (x, y, z, w) é solução ótima do problema dual.

□

Com o intuito de simplificar a notação, convencionamos o seguinte: se (P) tem solução ótima, dizemos que esta tem valor p^* , por sua vez, se (D) tem solução ótima dizemos que esta tem valor d^* . Também nomeamos *gap* de dualidade a diferença

$$\gamma = p^* - d^*.$$

Se o *gap* de dualidade é zero, dizemos que temos dualidade forte. Em geral, dualidade forte não é válida. Mas se o problema primal é convexo, isto é, da forma

$$\begin{aligned} \min \quad & f(x) \\ \text{s.a} \quad & g_i(x) \leq 0 \quad i = 1, \dots, m \\ & h_i(x) = 0 \quad i = 1, \dots, p \end{aligned} \tag{3.2.8}$$

com f, g_0, \dots, g_m convexas e h_1, \dots, h_p funções afins, usualmente (mas não sempre) temos dualidade forte. Existem muitas condições que asseguram dualidade forte para problemas convexos. Estas condições são chamadas *condição de qualificação de restrições*.

Uma condição de qualificação simples é a *condição de Slater*. Dizemos que um problema satisfaz a condição de Slater se ele é estritamente factível, isto é, existe $x \in \mathcal{D}$ tal que:

$$g_i(x) < 0, \quad i = 1, \dots, m, \quad h_i(x) = 0, \quad i = 1, \dots, p.$$

A condição de Slater pode ser redefinida para *forma fraca* da condição de Slater, em que a factibilidade estrita não é requerida, sempre que as funções g_i são afim. Temos então:

Teorema 5 (Dualidade Forte via Condição de Slater). *Se o problema primal (3.2.8) é convexo, e satisfaz a condição de Slater fraca, então a Dualidade forte é válida, isto é, $p^* = d^*$.*

Teorema 6 (Problemas de Otimização Quadrática Convexa). *Se f é quadrática convexa, e as funções $g_1, \dots, g_m, h_1, \dots, h_p$ são todas afim, então o gap de dualidade é sempre zero, desde que um dos problemas, primal ou dual seja factível.*

Mais detalhes sobre a condição de Slater e as demonstrações dos teoremas acima, podem ser encontradas em (BOYD; VANDENBERGHE, 1994). O Teorema da Complementaridade é outro importante resultado da otimização. No que segue, apresentamos uma adaptação deste para nossos problemas em questão (3.2.3) e (3.2.4).

Teorema 7 (Complementaridade). *Considere os problemas (3.2.3) e (3.2.4) e sejam (x, u, v) um ponto primal factível e (x, y, z, w) um ponto dual factível. Então (x, u, v) e*

(x, y, z, w) são soluções dos problemas (3.2.3) e (3.2.4) respectivamente, se e somente se, $z_i u_i = 0$ e $w_i v_i = 0$.

Demonstração. Suponha que (x, u, v) e (x, y, z, w) sejam soluções ótimas de (3.2.3) e (3.2.4) respectivamente. Pelo Teorema Dualidade Forte, o gap de dualidade é zero, logo

$$\begin{aligned}
u^T z + v^T w &= u^T(e - y) + v^T(e + y) \\
&= e^T(u + v) + y^T(v - u) \\
&= e^T(u + v) + y^T(Ax - b) \\
&= e^T(u + v) + x^T A^T y - y^T b \\
&= e^T(u + v) + \tau \|x\|_2^2 - y^T b \\
&= \left(\frac{\tau}{2} \|x\|_2^2 + e^T(u + v) \right) - \left(-\frac{\tau}{2} \|x\|_2^2 + y^T b \right) \\
&= \gamma \\
&= 0.
\end{aligned} \tag{3.2.9}$$

Como $(u, v) \geq 0$ e $(z, w) \geq 0$ para cada $i = 1, \dots, m$, tem-se $u_i z_i \geq 0$ e $v_i w_i \geq 0$, segue da equação (3.2.9) que $\sum_{i=1}^m u_i z_i + \sum_{i=1}^m v_i w_i = u^T z + v^T w = 0$, donde concluímos que $u_i z_i = 0$ e $v_i w_i = 0$ para cada $i = 1, \dots, m$.

Suponha agora que $(x, u, v) \in \mathcal{P}$ e $(x, y, z, w) \in \mathcal{D}$ sejam tais que $u^T z + v^T w = 0$. Pela equação (3.2.9) tem-se $p^* = d^*$. Usando o Corolário 2, concluímos que (x, u, v) e (x, y, z, w) são soluções ótimas dos seus respectivos problemas. \square

O Teorema 2 pode ser aplicado ao problema primal (3.2.3), pois

$$f(x, u, v) = \frac{\tau}{2} \|x\|_2^2 + e^T(u + v)$$

e $g_1(u) = -u$, $g_2(v) = -v$ são funções convexas e

$$h(x, u, v) = b - Ax - u + v = b - [A \ I_m \ -I_m] \begin{bmatrix} x \\ u \\ v \end{bmatrix}$$

é uma função afim, em que I_m representa a matriz identidade de ordem m . Além disso, todo ponto factível primal (x, u, v) é regular desde que a matriz de restrições $[A \ I_m \ -I_m]$ seja de posto completo.

Seja $\mathbf{x} = (x, u, v)$ e $\mathbf{y} = (y, z, w)$. Um ponto $\mathbf{x}^* = (x^*, u^*, v^*)$ primal factível satisfaz as condições de KKT dadas em (3.1.2) e (3.1.3) se existem $y^* \in \mathbb{R}^m$ e $z^*, w^* \in \mathbb{R}_+^m$, tais que:

$$\nabla_{\mathbf{x}}\ell(\mathbf{x}^*, \mathbf{y}^*) = \begin{bmatrix} 0_n \\ 0_m \\ 0_m \end{bmatrix}$$

$$z_i^* u_i^* = 0 \text{ e } w_i^* v_i^* = 0 \text{ para } i = 1, \dots, m,$$

onde

$$\ell(\mathbf{x}, \mathbf{y}) = \frac{\tau}{2} \|x\|_2^2 + e^T(u + v) + y^T(b - Ax - u + v) + z^T(-u) + w^T(-v),$$

e

$$\nabla_{\mathbf{x}}\ell(\mathbf{x}, \mathbf{y}) = \begin{bmatrix} \tau x - A^T y \\ e - y - z \\ e + y - w \end{bmatrix} = \begin{bmatrix} 0_n \\ 0_m \\ 0_m \end{bmatrix},$$

Em outras palavras, \mathbf{x}^* satisfaz as condições de KKT se:

1. $Ax^* + u^* - v^* = b$, $x^* \in \mathbb{R}^n$ e $u^*, v^* \in \mathbb{R}_+^m$, além disso, existem $y^* \in \mathbb{R}^m$ e $z^*, w^* \in \mathbb{R}_+^m$, tais que:
2.
$$\begin{cases} \tau x^* - A^T y^* &= 0_n \\ e - y^* - z^* &= 0_m \\ e + y^* - w^* &= 0_m \end{cases},$$
3. $z_i^* u_i^* = 0$ e $w_i^* v_i^* = 0$ para $i = 1, \dots, m$.

Os vetores $y^* \in \mathbb{R}^m$ e $z^*, w^* \in \mathbb{R}_+^m$ são chamados *multiplicadores de Lagrange*. Note que (x^*, \mathbf{y}^*) é um ponto factível dual e a condição 3 é chamada *condição de complementaridade*. O vetor $(\mathbf{x}^*, \mathbf{y}^*)$ que satisfaz as condições 1, 2 e 3 é chamado solução ótima *primal-dual*.

Pelos Teoremas 1 e 2, as condições de KKT fornecem condições suficientes e necessárias para encontrar uma solução ótima \mathbf{x}^* local do problema primal (P) e uma solução ótima local \mathbf{y}^* do problema dual (D).

Pelo Teorema 3 temos que toda solução local de um problema de programação convexa é também uma solução global. Sendo assim, $(\mathbf{x}^*, \mathbf{y}^*)$ é uma solução ótima global de um problema de programação convexa se, e somente se, as condições de KKT são satisfeitas.

3.3 Um método de pontos interiores primal-dual para regularização de Tikhonov

Nesta seção apresentamos o Método de Pontos Interiores (MPI) primal-dual para o par primal-dual:

$$(P) \left\{ \begin{array}{l} \min_x \quad \frac{\tau}{2} \|x\|_2^2 + e^T(u+v) \\ \text{sujeito a} \quad Ax + u - v = b. \\ (u, v) \geq 0 \text{ e } x \in \mathbb{R}^n. \end{array} \right. \quad (D) \left\{ \begin{array}{l} \max_x \quad -\frac{\tau}{2} \|x\|^2 + y^T b \\ \text{sujeito a} \quad \tau x - A^T y = 0 \\ e - y - z = 0 \\ e + y - w = 0 \\ (z, w) \geq 0 \text{ e } y \in \mathbb{R}^m. \end{array} \right. \quad (3.3.1)$$

onde $x \in \mathbb{R}^n$, $u^*, v^* \in \mathbb{R}_+^m$, $y^* \in \mathbb{R}^m$, $z^*, w^* \in \mathbb{R}_+^m$ e A é uma matriz de tamanho $m \times n$ considerada de posto completo. Os conjuntos

$$\begin{aligned} \mathcal{P} &= \{(x, u, v) : Ax + u - v = b, (u, v) \geq 0\}, \\ \mathcal{D} &= \{(x, y, z, w) : e - y - z = 0, e + y - w = 0, \tau x - A^T y = 0, (z, w) \geq 0\} \text{ e} \\ \mathcal{F} &= \{(\mathbf{x}, \mathbf{y}) : Ax + u - v = b, e - y - z = 0, e + y - w = 0, \tau x - A^T y = 0, \\ &\quad (u, v, z, w) \geq 0\}, \end{aligned}$$

são chamados conjuntos de pontos primal, dual e primal-dual factíveis, respectivamente, dos problemas apresentados em (3.3.1). Além disso, os conjuntos

$$\begin{aligned} \mathcal{P}^\circ &= \{(x, u, v) : Ax + u - v = b, (u, v) > 0\}, \\ \mathcal{D}^\circ &= \{(x, y, z, w) : e - y - z = 0, e + y - w = 0, \tau x - A^T y = 0, (z, w) > 0\} \text{ e} \\ \mathcal{F}^\circ &= \{(\mathbf{x}, \mathbf{y}) : Ax + u - v = b, e - y - z = 0, e + y - w = 0, \tau x - A^T y = 0, \\ &\quad (u, v, z, w) > 0\}, \end{aligned} \quad (3.3.2)$$

são chamados conjuntos de pontos estritamente primal, dual e primal-dual factíveis, respectivamente, ou interiores.

De acordo com o Teorema 2, um ponto primal factível $\mathbf{x}^* = (x^*, u^*, v^*)$ é solução de (P) se existem $y^* \in \mathbb{R}^m$ e $z^*, w^* \in \mathbb{R}_+^m$, tais que:

$$\begin{aligned}
\tau x^* - A^T y^* &= 0_n \\
e - y^* - z^* &= 0_m \quad z^* \geq 0, \\
e + y^* - w^* &= 0_m \quad w^* \geq 0, \\
z_i^* u_i^* &= 0, \text{ para } i = 1, \dots, m \\
w_i^* v_i^* &= 0, \text{ para } i = 1, \dots, m
\end{aligned}$$

ou equivalentemente $\mathbf{x}^* = (x^*, u^*, v^*)$ é solução ótima de (P) se existem $y^* \in \mathbb{R}^m$ e $z^*, w^* \in \mathbb{R}_+^m$, tais que:

$$\begin{aligned}
Ax^* + u^* - v^* &= b, \quad (u, v) \geq 0; \\
\tau x^* - A^T y^* &= 0_n; \\
e - y^* - z^* &= 0_m, \quad z^* \geq 0; \\
e + y^* - w^* &= 0_m, \quad w^* \geq 0; \\
Z^* U^* e &= 0; \\
W^* V^* e &= 0;
\end{aligned} \tag{3.3.3}$$

onde $Z^* = \text{diag}(z_1^*, z_2^*, \dots, z_m^*)$, $U^* = \text{diag}(u_1^*, u_2^*, \dots, u_m^*)$, $W^* = \text{diag}(w_1^*, w_2^*, \dots, w_m^*)$ e $V^* = \text{diag}(v_1^*, v_2^*, \dots, v_m^*)$. Observe que (x^*, y^*, z^*, w^*) é solução ótima do problema dual (D) .

De forma semelhante a outros métodos iterativos de otimização, o MPI primal-dual requer dois componentes elementares: um procedimento para determinar a direção de busca e outro para determinar o tamanho do passo. A direção de busca $d = (\Delta x, \Delta u, \Delta v, \Delta y, \Delta z, \Delta w)$ é obtida usando uma variação do método de Newton nas equações (3.3.3). O tamanho de passo, α_k numa iteração k é obtido de tal forma que:

$$(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}) = (\mathbf{x}_k, \mathbf{y}_k) + \alpha_k d_k$$

onde

$$\begin{aligned}
(\mathbf{x}_k, \mathbf{y}_k) &= (x_k, u_k, v_k, y_k, z_k, w_k) \\
&\text{e} \\
d_k &= (\Delta x^k, \Delta u^k, \Delta v^k, \Delta y^k, \Delta z^k, \Delta w^k).
\end{aligned}$$

No que segue detalharemos como a direção de busca é obtida, e se faz importante ressaltar, embora já mencionado anteriormente, que o Método de Pontos Interiores primal-dual está relacionado com os métodos de barreira, (WRIGHT, 1997).

Se aplicarmos a penalização barreira logarítmica nas restrições de não negatividade, da formulação primal (P) dada em (3.3.1), temos:

$$\begin{aligned} \min \quad & \frac{\tau}{2} \|x\|_2^2 + e^T(u + v) - \mu \left(\sum_{i=1}^m \log(u_i) + \sum_{i=1}^m \log(v_i) \right) \\ \text{s.a.} \quad & Ax + u - v = b \quad (u, v) > 0, \end{aligned} \quad (3.3.4)$$

onde $\mu > 0$, e a expressão

$$\sum_{i=1}^m \log(u_i) + \sum_{i=1}^m \log(v_i)$$

é conhecida como *termo barreira*. O problema (3.3.4) é convexo e seu domínio é o conjunto estritamente factível

$$\mathcal{P}^\circ = \{(x, u, v) : Ax + u - v = b, (u, v) > 0\}.$$

O parâmetro μ controla a relação entre o termo de barreira e a função objetivo do problema (P), ou seja, se queremos dar maior prioridade a função objetivo, $\frac{\tau}{2} \|x\|_2^2 + e^T(u+v)$, basta tomarmos μ pequeno. No entanto, para efeito de que u e v sejam positivos, o MPI primal-dual inicia-se com μ grande, e ele é reduzido gradualmente para dar prioridade à minimização da função objetivo.

Como o problema (3.3.4) é convexo, pelo Teorema 2, as condições de otimalidade KKT, são necessárias e suficientes para encontrar uma solução ótima. Para tal finalidade, consideremos o lagrangeano ℓ do problema (3.3.4) :

$$\ell(\mathbf{x}, y) = \frac{\tau}{2} \|x\|_2^2 + e^T(u + v) + y^T(b - Ax - u + v) - \mu \left(\sum_{i=1}^m \log(u_i) + \sum_{i=1}^m \log(v_i) \right). \quad (3.3.5)$$

Suas derivadas parciais são dadas por:

1. $\nabla_x \ell(\mathbf{x}, y) = \tau x - A^T y$;
2. $\nabla_u \ell(\mathbf{x}, y) = e - y - \mu U^{-1} e$;
3. $\nabla_v \ell(\mathbf{x}, y) = e + y - \mu V^{-1} e$;
4. $\nabla_y \ell(\mathbf{x}, y) = b - Ax - u + v$;

onde $U^{-1} = \text{diag}(1/u_1, \dots, 1/u_m)$, $V^{-1} = \text{diag}(1/v_1, \dots, 1/v_m)$, e $e^T = (1, \dots, 1)^T \in \mathbb{R}^m$.

Se $z \in \mathbb{R}^m$ e $w \in \mathbb{R}^m$ são definidos respectivamente por $z = \mu U^{-1} e$ e $w = \mu V^{-1} e$, o gradiente do lagrangeano ℓ dado por (3.3.5), o qual denotamos por $\nabla \ell = (\nabla_x \ell, \nabla_u \ell, \nabla_v \ell, \nabla_y \ell)$ tem as seguintes componentes:

1. $\nabla_x \ell(\mathbf{x}, y) = \tau x - A^T y$;
2. $\nabla_u \ell(\mathbf{x}, y) = e - y - z$;
3. $\nabla_v \ell(\mathbf{x}, y) = e + y - w$;
4. $\nabla_y \ell(\mathbf{x}, y) = b - Ax - u + v$;

Para $\mu > 0$, pelos Teoremas 1 e 2, $(\mathbf{x}^*, y^*) = (\mathbf{x}^*(\mu), y^*(\mu))$ é solução ótima do problema (3.3.4) se, e somente se, existem $y^* = y^*(\mu) \in \mathbb{R}^m$, $z^* = z^*(\mu) \in \mathbb{R}_{++}^m$ e $w^* = w^*(\mu) \in \mathbb{R}_{++}^m$ que satisfazem:

$$\nabla \ell(\mathbf{x}, \mathbf{y}) = 0,$$

$$z = \mu U^{-1} e \quad \text{e} \quad w = \mu V^{-1} e,$$

ou

$$UZe = \mu e \quad \text{e} \quad VW = \mu e$$

Para ser mais exato, $(\mathbf{x}^*, y^*) = (\mathbf{x}^*(\mu), y^*(\mu))$ é solução ótima do problema dado por (3.3.4) se, e somente se, existem $y^* = y^*(\mu) \in \mathbb{R}^m$, $z^* = z^*(\mu) \in \mathbb{R}_{++}^m$ e $w^* = w^*(\mu) \in \mathbb{R}_{++}^m$ que satisfazem as seguintes equações:

$$Ax + u - v = b, \quad (u, v) > 0; \tag{3.3.6}$$

$$\tau x - A^T y = 0_n; \tag{3.3.7}$$

$$e - y - z = 0_m, \quad z > 0; \tag{3.3.8}$$

$$e + y - w = 0_m, \quad w > 0; \tag{3.3.9}$$

$$ZUe = \mu e; \tag{3.3.10}$$

$$WVe = \mu e; \tag{3.3.11}$$

onde $Z = \text{diag}(z_1, z_2, \dots, z_m)$, $U = \text{diag}(u_1, u_2, \dots, u_m)$, $W = \text{diag}(w_1, w_2, \dots, w_m)$ e $V = \text{diag}(v_1, v_2, \dots, v_m)$.

Para cada $\mu > 0$ existe um único ponto $(\mathbf{x}(\mu), \mathbf{y}(\mu))$ que satisfaz as seis equações (3.3.6)-(3.3.11). Este resultado é expresso na forma do seguinte teorema.

Teorema 8. *Suponha que o conjunto de pontos primal-dual estritamente factíveis dado por*

$$\mathcal{F}^\circ = \{(\mathbf{x}, \mathbf{y}) : Ax + u - v = b, \quad e - y - z = 0, \quad e + y - w = 0, \quad \tau x - A^T y = 0,$$

$$(u, v, z, w) > 0\}$$

seja não vazio, então para cada $\mu > 0$ existe uma única solução do sistema de equações dadas em (3.3.6)-(3.3.11).

Em vista da linearidade das equações que definem o conjunto \mathcal{F}° , sob algumas convenções, o teorema acima pode ser provado de forma semelhante ao Teorema 2.8 encontrado em (WRIGHT, 1997).

3.3.1 Trajetória central

O Teorema 8, por meio das equações (3.3.6)-(3.3.11), define implicitamente uma curva \mathcal{C} . (WRIGHT, 1997) define esta curva como *Trajétória Central* \mathcal{C} , uma curva de pontos estritamente factíveis, isto é, uma curva de pontos pertencentes a

$$\mathcal{F}^\circ = \{(\mathbf{x}, \mathbf{y}) : Ax + u - v = b, e - y - z = 0, e + y - w = 0, \tau x - A^T y = 0, (u, v, z, w) > 0\}.$$

Esta curva tem forma paramétrica:

$$\begin{aligned} \mathcal{C} : \mathbb{R}_{++} &\longrightarrow \mathbb{R}^n \times \mathbb{R}_{++}^{2m} \times \mathbb{R}^m \times \mathbb{R}_{++}^{2m} \\ \mu &\longrightarrow (x(\mu), u(\mu), v(\mu), y(\mu), z(\mu), w(\mu)) \end{aligned}$$

onde $(x(\mu), u(\mu), v(\mu), y(\mu), z(\mu), w(\mu))$ é obtida implicitamente do sistema de equações (3.3.6)-(3.3.11). Sendo assim, definimos a Trajetória Central como o conjunto

$$\mathcal{C} = \{(x(\mu), u(\mu), v(\mu), y(\mu), z(\mu), w(\mu)) : \mu > 0\}.$$

O MPI primal-dual seguidor de caminho segue a Trajetória Central na direção decrescente de μ . Como o sistema de equações (3.3.6)-(3.3.11) é contínuo, temos que $(x(\mu), u(\mu), v(\mu), y(\mu), z(\mu), w(\mu)) \approx (x^*, u^*, v^*, y^*, z^*, w^*)$ desde que $\mu \approx 0$.

Se usarmos um método iterativo para resolver o sistema de equações:

$$Ax + u - v = b; \tag{3.3.12}$$

$$\tau x - A^T y = 0_n; \tag{3.3.13}$$

$$e - y - z = 0_m; \tag{3.3.14}$$

$$e + y - w = 0_m; \tag{3.3.15}$$

$$ZUe = 0; \tag{3.3.16}$$

$$WVe = 0; \tag{3.3.17}$$

$$u, v \geq 0; \tag{3.3.18}$$

$$z, w \geq 0; \tag{3.3.19}$$

gerado pelas condições de KKT do problema (P) , em geral são encontrados vetores (x, u, v, y, z, w) que satisfazem (3.3.12)-(3.3.19), no entanto, pode acontecer que os vetores u, v, z e w saiam prematuramente do ortante positivo, violando as condições de positividade (3.3.18) e (3.3.19).

A função da Trajetória Central é guiar a uma solução ótima ao longo de um caminho que mantém os produtos $z_i u_i$ e $w_i v_i$ estritamente positivos e os reduzindo a zero a uma mesma taxa. Portanto, em vez de resolver o sistema dado pelas condições de KKT de (P) , iremos resolver o sistema dado pelas equações (3.3.6)-(3.3.11) para diferentes valores positivos μ , com μ decrescente.

O sistema de equações (3.3.6)-(3.3.11) usado para encontrar a Trajetória Central é não linear pois as equações (3.3.10)-(3.3.11) são não lineares, por outro lado as equações (3.3.6)-(3.3.9) são lineares. Sendo assim, uma aproximação de primeira ordem irá fornecer uma boa aproximação de um ponto da Trajetória Central.

Considere a aplicação $F : \mathbb{R}^n \times \mathbb{R}_{++}^{2m} \times \mathbb{R}^m \times \mathbb{R}_{++}^{2m} \longrightarrow \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^{4m}$ dada por:

$$F(\mathbf{x}, \mathbf{y}) = \begin{pmatrix} Ax + u - v - b \\ \tau x - A^T y \\ e - y - z \\ e + y - w \\ UZe - \mu e \\ VWe - \mu e \end{pmatrix}$$

a aproximação de primeira ordem de F no ponto $\hat{X} = (\mathbf{x}, \mathbf{y})^T$ é dada por:

$$\hat{F}(X) = F(X) + J(\hat{X})(X - \hat{X}) \text{ para } X \approx \hat{X},$$

onde $J(\hat{X})$ é o Jacobiano de F no ponto \hat{X} .

Resolve-se $F(\hat{X}) + J(\hat{X})d = 0$, para encontrar a direção de busca $d = (\Delta x, \Delta u, \Delta v, \Delta y, \Delta z, \Delta w)^T$ no ponto \hat{X} , isto implica resolver o sistema de equações:

$$\begin{pmatrix} A & I_m & -I_m & 0 & 0 & 0 \\ \tau I_n & 0 & 0 & -A^T & 0 & 0 \\ 0 & 0 & 0 & -I_m & -I_m & 0 \\ 0 & 0 & 0 & I_m & 0 & -I_m \\ 0 & Z & 0 & 0 & U & 0 \\ 0 & 0 & W & 0 & 0 & V \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta u \\ \Delta v \\ \Delta y \\ \Delta z \\ \Delta w \end{pmatrix} = \begin{pmatrix} r_b \\ r_y \\ r_z \\ r_w \\ r_1 \\ r_2 \end{pmatrix} \quad (3.3.20)$$

em que

$$\begin{cases} r_b &= b - Ax - u + v \\ r_y &= A^T y - \tau x \\ r_z &= z + y - e \\ r_w &= w - y - e \\ r_1 &= \mu e - UZe \\ r_2 &= \mu e - VWe \end{cases}$$

e as matrizes U, Z, V e W são matrizes diagonais dadas por $U = \text{diag}(u_1, u_2, \dots, u_m)$, $Z = \text{diag}(z_1, z_2, \dots, z_m)$, $V = \text{diag}(v_1, v_2, \dots, v_m)$ e $W = \text{diag}(w_1, w_2, \dots, w_m)$ e $e^T = (1, 1, \dots, 1)^T \in \mathbb{R}^m$.

Para reduzir a condição de complementaridade, as diferentes variantes do MPI primal-dual seguidor de trajetória modificam o sistema (3.3.20) fazendo

$$r_1 = \sigma \mu e - UZe \text{ e } r_2 = \sigma \mu e - VWe, \quad (3.3.21)$$

onde $\sigma \in [0, 1]$ é chamado *parâmetro de centragem* e é dado como proposto por (ZHANG, 1993):

$$\sigma^k = \begin{cases} \frac{1}{(2n)^\gamma} & \text{se } \gamma^k > 1 \\ \frac{1}{(2n)^{3/2}} & \text{caso } \gamma^k \leq 1, \end{cases}$$

em que $\gamma^k = u_k^T z_k + v_k^T w_k$.

3.3.2 Etapas do método de pontos interiores primal-dual

Os primeiros MPIs primal-dual trabalhavam com sequências de pontos factíveis, atualmente exige-se apenas que cada ponto da sequência $(x_k, u_k, v_k, y_k, z_k, w_k)$ seja um ponto interior, isto é, $(u_k, v_k, z_k, w_k) > 0$.

Na literatura, a estrutura mais encontrada para MPIs primal-dual com pontos infactíveis é dada pelo Algoritmo 2.

O critério de parada do método é baseado nas condições de otimalidade relativas:

$$\frac{\|b - Ax^k - u^k + v^k\|}{\|b\| + 1} \leq \epsilon, \quad \frac{\|A^T y^k - x^k\|}{\|x^k\| + 1} \leq \epsilon, \quad \frac{\|z^k + y^k - e\|}{\|y^k\| + 1} \leq \epsilon, \quad \frac{\|w^k - y^k - e\|}{\|y^k\| + 1} \leq \epsilon,$$

$$\frac{2|\tau\| \|x^k\|_2^2 + e^T(u^k + v^k) - b^T y^k}{\|x^k\|_2^2 + 2e^T(u^k + v^k) + \|x^k\|_2^2 - 2b^T y^k + 2} \leq \epsilon. \quad (3.3.22)$$

Em geral, escolhe-se $\epsilon = 10^{-8}$.

Algoritmo 2 – Etapas do MPI primal-dual**Dados:** $(x^0, u^0, v^0, y^0, z^0, w^0)$ com $(u^0, v^0, z^0, w^0) > 0$ **Resultado:** Solução do MPI primal-dual**para** $k = 1, 2, \dots$ **faça**

1. Escolha $\sigma^k \in [0, 1]$ e calcule $\mu^k = \frac{(u^k)^T z^k + (v^k)^T w^k}{2n}$;

2. Resolva o sistema de equações (3.3.20) para encontrar as direções de busca $d_k = (\Delta x^k, \Delta u^k, \Delta v^k, \Delta y^k, \Delta z^k, \Delta w^k)$ usando μ_k e σ_k do passo 1 ;

3. Calcule o comprimento de passo α^k , dado por:

$$\alpha^k = \beta^k \min \{ \rho_u, \rho_v, \rho_w, \rho_z \} \text{ onde } \beta^k \in (0, 1), \text{ e}$$

$$\rho_u = \min_i \left\{ -\frac{u_i^k}{\Delta u_i^k} \mid \Delta u_i^k < 0 \right\}, \quad \rho_v = \min_i \left\{ -\frac{v_i^k}{\Delta v_i^k} \mid \Delta v_i^k < 0 \right\},$$

$$\rho_w = \min_i \left\{ -\frac{w_i^k}{\Delta w_i^k} \mid \Delta w_i^k < 0 \right\}, \quad \rho_z = \min_i \left\{ -\frac{z_i^k}{\Delta z_i^k} \mid \Delta z_i^k < 0 \right\};$$

4. Calcule o novo ponto

$$(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}) = (\mathbf{x}_k, \mathbf{y}_k) + \alpha^k d_k$$

onde

$$(\mathbf{x}_k, \mathbf{y}_k) = (x^k, u^k, v^k, y^k, z^k, w^k).$$

fim

Na Programação Linear o método Predictor-Corretor de Mehrotra (PCM), veja (MEHROTRA, 1992), é uma variante do MPI primal-dual seguidor de caminho e destaca-se por ser uma aproximação de segunda ordem de condições de otimalidade como as de (3.3.4). Na seção seguinte, apresentamos uma adaptação deste método para o problema (3.3.4).

3.3.3 Método predictor-corretor de Mehrotra

Muitos *softwares* voltados à resolução de problemas de Programação Linear, como BPMPD, CPLEX/Barrier, HOPDM, LIPSOL, PCx, assim como outros, utilizam uma implementação baseada neste método. Embora o método de Mehrotra não fuja à estrutura apresentada na Seção 3.3.2, sua diferença está na escolha das direções de buscas,

que são obtidas por resolver dois sistemas lineares. Primeiro, resolve-se

$$\begin{pmatrix} A & I_m & -I_m & 0 & 0 & 0 \\ \tau I_n & 0 & 0 & -A^T & 0 & 0 \\ 0 & 0 & 0 & -I_m & -I_m & 0 \\ 0 & 0 & 0 & I_m & 0 & -I_m \\ 0 & Z^k & 0 & 0 & U^k & 0 \\ 0 & 0 & W^k & 0 & 0 & V^k \end{pmatrix} \begin{pmatrix} \Delta x^{af} \\ \Delta u^{af} \\ \Delta v^{af} \\ \Delta y^{af} \\ \Delta z^{af} \\ \Delta w^{af} \end{pmatrix} = \begin{pmatrix} r_b^k \\ r_y^k \\ r_z^k \\ r_w^k \\ -U^k Z^k e \\ -V^k W^k e \end{pmatrix} \quad (3.3.23)$$

onde

$$\begin{cases} r_b^k &= b - Ax^k - u^k + v^k \\ r_y^k &= A^T y^k - \tau x^k \\ r_z^k &= z^k + y^k - e \\ r_w^k &= w^k - y^k - e \end{cases}$$

e $U^k = \text{diag}(u_1^k, u_2^k, \dots, u_m^k)$, $Z = \text{diag}(z_1^k, z_2^k, \dots, z_m^k)$, $V = \text{diag}(v_1^k, v_2^k, \dots, v_m^k)$ e $W = \text{diag}(w_1^k, w_2^k, \dots, w_m^k)$ e $e^T = (1, 1, \dots, 1)^T \in \mathbb{R}^m$.

A direção $d_k^{af} = (\Delta x^{af}, \Delta u^{af}, \Delta v^{af}, \Delta y^{af}, \Delta z^{af}, \Delta w^{af})^T$ é chamada direção afim-escala ou preditora em $(x^k, u^k, v^k, y^k, z^k, w^k)$ e consiste em um passo na direção de Newton puro.

O comprimento de passo nesta direção é dado por:

$$\alpha^{af} = \beta^k \min \{ \alpha_u^{af}, \alpha_v^{af}, \alpha_z^{af}, \alpha_w^{af} \} \text{ onde } \beta^k \in (0, 1) \text{ e}$$

$$\begin{aligned} \alpha_u^{af} &= \operatorname{argmax} \{ \alpha \in (0, 1] : u^k + \alpha \Delta u^{af} > 0 \}, \\ \alpha_v^{af} &= \operatorname{argmax} \{ \alpha \in (0, 1] : v^k + \alpha \Delta v^{af} > 0 \}, \\ \alpha_z^{af} &= \operatorname{argmax} \{ \alpha \in (0, 1] : z^k + \alpha \Delta z^{af} > 0 \}, \\ \alpha_w^{af} &= \operatorname{argmax} \{ \alpha \in (0, 1] : w^k + \alpha \Delta w^{af} > 0 \}. \end{aligned} \quad (3.3.24)$$

A fim de determinar a eficiência do passo na direção preditora, é calculada a medida de dualidade afim μ_{af} dada por:

$$\mu_{af}^k = \frac{(u^k + \alpha_u^{af} \Delta u^{af})(z^k + \alpha_z^{af} \Delta z^{af}) + (v^k + \alpha_v^{af} \Delta v^{af})(w^k + \alpha_w^{af} \Delta w^{af})}{2n}. \quad (3.3.25)$$

Para encontrar a direção preditora e de centragem, usaremos μ^k e μ_{af}^k , para calcular o parâmetro de centragem σ_k , a partir de duas considerações:

1. Se $\mu_{af}^k \ll \mu^k$, então a direção preditora é uma boa direção de busca pois permite uma redução significativa da medida de dualidade permanecendo interior, neste caso, não será necessária uma centragem muito grande, isto é, $\sigma_k \approx 0$.
2. Se $\mu_{af}^k \approx \mu^k$, então a direção preditora fez um pequeno progresso na redução da medida de dualidade, logo é preciso fazer uma centragem maior (mais perto da trajetória central) com o objetivo de que o ponto na iteração $k + 1$ esteja numa melhor posição para atingir um maior decréscimo da medida de dualidade na iteração seguinte.

Para encontrar o parâmetro de centragem σ_k da iteração k , Mehrotra sugere uma heurística bem sucedida dada por:

$$\sigma_k = \left(\frac{\mu_{af}^k}{\mu^k} \right)^3, \quad (3.3.26)$$

evidentemente, se $\mu_{af}^k \ll \mu^k$, então $\sigma_k \approx 0$, no entanto, se $\mu_{af}^k \approx \mu^k$, então $\sigma_k \approx 1$.

No sistema (3.3.23), suponha $\alpha_x^{af} = \alpha_u^{af} = \alpha_v^{af} = \alpha_y^{af} = \alpha_z^{af} = \alpha_w^{af} = 1$, ou seja, um passo otimista na direção preditora, então:

$$\begin{aligned} r_b^{k+1} &= b - Ax^{k+1} - u^{k+1} + v^{k+1} \\ &= b - A(x^k + \Delta x^{af}) - (u^k + \Delta u^{af}) + (v^k + \Delta v^{af}) \\ &= (b - Ax^k - u^k + v^k) - (A\Delta x^{af} + \Delta u^{af} - \Delta v^{af}) \\ &= r_b^k - r_b^k = 0; \\ r_y^{k+1} &= A^T y^{k+1} - \tau x^{k+1} \\ &= A^T (y^k + \Delta y^{af}) - \tau (x^k + \Delta x^{af}) \\ &= (A^T y^k - \tau x^k) - (-A^T \Delta y^{af} + \tau \Delta x^{af}) \\ &= r_y^k - r_y^k = 0 \\ r_z^{k+1} &= z^{k+1} + y^{k+1} - e \\ &= (z^k + \Delta z^{af}) + (y^k + \Delta y^{af}) - e \\ &= (z^k + y^k - e) - (-\Delta z^{af} - \Delta y^{af}) \\ &= r_z^k - r_z^k = 0; \\ r_w^{k+1} &= w^{k+1} - y^{k+1} - e \\ &= (w^k + \Delta w^{af}) - (y^k + \Delta y^{af}) - e \\ &= (w^k - y^k - e) - (-\Delta w^{af} + \Delta y^{af}) \\ &= r_w^k - r_w^k = 0. \end{aligned}$$

isto é,

$$r_b^{k+1} = r_y^{k+1} = r_z^{k+1} = r_w^{k+1} = 0. \quad (3.3.27)$$

Como podemos notar, a factibilidade é atingida em apenas um passo, no entanto, para as folgas complementares, temos:

$$\begin{aligned}(u_i + \Delta u_i^{af})(z_i + \Delta z_i^{af}) &= u_i z_i + u_i \Delta z_i^{af} + \Delta u_i^{af} z_i + \Delta u_i^{af} \Delta z_i^{af} = \Delta u_i^{af} \Delta z_i^{af}, \\ (v_i + \Delta v_i^{af})(w_i + \Delta w_i^{af}) &= v_i w_i + v_i \Delta w_i^{af} + \Delta v_i^{af} w_i + \Delta v_i^{af} \Delta w_i^{af} = \Delta v_i^{af} \Delta w_i^{af},\end{aligned}$$

isto é, as folgas complementares não se anulam. Sendo assim, a otimalidade não é atingida pois um passo de Newton puro é apenas uma aproximação linear de primeira ordem das condições de otimalidade. Chegamos assim à segunda etapa da direção de busca, em que tentamos encontrar uma direção corretora e de centragem, que tenta compensar a não linearidade da direção preditora com uma escolha adaptativa do parâmetro de centragem σ_k .

Nesta etapa, a ideia natural é tentar criar uma direção que faça $\Delta u_i^{af} \Delta z_i^{af} = 0$ e $\Delta v_i^{af} \Delta w_i^{af} = 0$. Não obstante, a contrapartida é o risco de que $(u_i + \Delta u_i^{af})(z_i + \Delta z_i^{af})$ e $(v_i + \Delta v_i^{af})(w_i + \Delta w_i^{af})$ tendam a zero com taxas de convergência diferentes. Assim, usa-se o parâmetro de centragem σ_k definido por (3.3.26), visto que este depende da medida de dualidade e da posição do ponto da iteração corrente no ortante positivo. Portanto, a direção corretora de centragem, a qual denotamos por $d_k^{cc} = (\Delta x^{cc}, \Delta u^{cc}, \Delta v^{cc}, \Delta y^{cc}, \Delta z^{cc}, \Delta w^{cc})$, pretende obter:

$$\Delta u_i^{af} \Delta z_i^{af} = \sigma_k \mu^k \quad \text{e} \quad \Delta z_i^{af} \Delta w_i^{af} = \sigma_k \mu^k \quad \text{para todo } i = 1, \dots, m. \quad (3.3.28)$$

De (3.3.3) e (3.3.28), esta direção nos leva a:

$$(u_i + \Delta u_i^{af})(z_i + \Delta z_i^{af}) = \sigma_k \mu^k \quad \text{e} \quad (v_i + \Delta v_i^{af})(w_i + \Delta w_i^{af}) = \sigma_k \mu^k.$$

Das equações (3.3.27) e (3.3.28), concluímos que a direção corretora e de centragem é obtida pela resolução do seguinte sistema linear:

$$\begin{pmatrix} A & I_m & -I_m & 0 & 0 & 0 \\ \tau I_n & 0 & 0 & -A^T & 0 & 0 \\ 0 & 0 & 0 & -I_m & -I_m & 0 \\ 0 & 0 & 0 & I_m & 0 & -I_m \\ 0 & Z^k & 0 & 0 & U^k & 0 \\ 0 & 0 & W^k & 0 & 0 & V^k \end{pmatrix} \begin{pmatrix} \Delta x^{cc} \\ \Delta u^{cc} \\ \Delta v^{cc} \\ \Delta y^{cc} \\ \Delta z^{cc} \\ \Delta w^{cc} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \sigma_k \mu^k e - \Delta U^{af} \Delta Z^{af} e \\ \sigma_k \mu^k e - \Delta V^{af} \Delta W^{af} e \end{pmatrix} \quad (3.3.29)$$

onde ΔU^{af} , ΔZ^{af} , ΔV^{af} e ΔW^{af} são matrizes diagonais dadas por: $\Delta U^{af} = \text{diag}(\Delta u^{af})$, $\Delta Z^{af} = \text{diag}(\Delta z^{af})$, $\Delta V^{af} = \text{diag}(\Delta v^{af})$ e $\Delta W^{af} = \text{diag}(\Delta w^{af})$.

Para concluir, a soma $d^k = d_k^{af} + d_k^{cc}$, nos fornece a direção de busca do método PCM.

Esta variante reduz significativamente o número de iterações para o método primal-dual. Esta redução é obtida ao preço de resolver dois sistemas lineares por iteração. No entanto, estes sistemas tem a matriz de coeficientes em comum. No caso da Programação Linear, constatou-se na prática que o trabalho de resolver o sistema linear extra é compensado pela economia na redução do número de iterações (WRIGHT, 1997).

3.3.4 Etapas do MPI primal-dual na sua variante PCM

Algoritmo 3 – Etapas do PCM

Dados: $(x^0, u^0, v^0, y^0, z^0, w^0)$ com $(u^0, v^0, z^0, w^0) > 0$

Resultado: Solução do PCM

para $k = 1, 2, \dots$ **faça**

1. Escolha $\sigma^k \in [0, 1]$ e calcule $\mu^k = ((u^k)^T z^k + (v^k)^T w^k)/2n$;
2. Calcule a direção afim escala ou preditora d_k^{af} do sistema (3.3.23);
3. Calcule $\alpha_u^{af}, \alpha_v^{af}, \alpha_z^{af}$ e α_w^{af} de acordo com (3.3.24);
4. Calcule μ_{af} a partir de (3.3.25);
5. A partir de (3.3.26), determine o parâmetro de centragem σ_k ;
6. Calcule a direção corretora e de centragem d_k^{cc} de acordo com (3.3.29);
7. Obtenha a direção de descida $d^k = d_k^{af} + d_k^{cc}$;

8. Calcule o comprimento de passo $\alpha^k = \beta^k \min \{\rho_u, \rho_v, \rho_w, \rho_z\}$ onde:

$$\beta^k \in (0, 1), \text{ e } \rho_u = \min_i \left\{ -\frac{u_i^k}{\Delta u_i^k} \mid \Delta u_i^k < 0 \right\}, \rho_v = \min_i \left\{ -\frac{v_i^k}{\Delta v_i^k} \mid \Delta v_i^k < 0 \right\},$$

$$\rho_w = \min_i \left\{ -\frac{w_i^k}{\Delta w_i^k} \mid \Delta w_i^k < 0 \right\}, \rho_z = \min_i \left\{ -\frac{z_i^k}{\Delta z_i^k} \mid \Delta z_i^k < 0 \right\};$$

9. Calcule o novo ponto

$$(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}) = (\mathbf{x}_k, \mathbf{y}_k) + \alpha^k d^k$$

onde

$$(\mathbf{x}_k, \mathbf{y}_k) = (x^k, u^k, v^k, y^k, z^k, w^k).$$

fim

O critério de convergência é baseado nas condições de otimalidade relativas dadas em (3.3.22).

3.4 Cálculo da direção de busca

Como visto nas seções anteriores, o MPI primal dual revolve um ou dois sistemas lineares para encontrar a direção de busca. A maior parte do tempo total de processamento do MPI primal-dual corresponde ao método que resolve estes sistemas lineares. Segundo (GONDZIO, 1996), o cálculo da direção de busca tem custo de 60% à 90% do tempo total de processamento. Na subseção que segue, apresentamos duas estratégias para a determinação da direção de busca do MPI primal-dual.

3.4.1 Direção de busca para o MPI primal-dual

Para obter a direção do MPI primal-dual, devemos resolver o sistema (3.3.20), para tanto, vamos reduzi-lo a duas formas, semelhante ao que se faz para problemas de programação linear. Com efeito, da terceira e quarta equação do sistema (3.3.20), temos:

$$\Delta z^k = -r_z^k - \Delta y^k, \quad (3.4.1)$$

$$\Delta w^k = \Delta y^k - r_w^k. \quad (3.4.2)$$

Substituindo as igualdades (3.4.1) e (3.4.2), na quinta e na sexta equação do sistema (3.3.20), obtemos:

$$\Delta u^k = (Z^k)^{-1} (r_1^k - U^k \Delta z^k) = (Z^k)^{-1} r_1^k + (Z^k)^{-1} U^k r_z^k + (Z^k)^{-1} U^k \Delta y^k, \quad (3.4.3)$$

$$\Delta v^k = (W^k)^{-1} (r_2^k - V^k \Delta w^k) = (W^k)^{-1} r_2^k - (W^k)^{-1} V^k \Delta y^k + (W^k)^{-1} V^k r_w^k. \quad (3.4.4)$$

Substituindo as igualdades (3.4.3) e (3.4.4), na primeira equação do sistema (3.3.20), obtemos:

$$\begin{aligned} A\Delta x^k &= r_b^k - \Delta u^k + \Delta v^k \\ &= r_b^k - (Z^k)^{-1} r_1^k - (Z^k)^{-1} U^k r_z^k - (Z^k)^{-1} U^k \Delta y^k + (W^k)^{-1} r_2^k - (W^k)^{-1} V^k \Delta y^k \\ &\quad + (W^k)^{-1} V^k r_w^k, \end{aligned} \quad (3.4.5)$$

ou ainda

$$A\Delta x^k = q^k - (\Theta^k)^{-1} \Delta y^k, \quad (3.4.6)$$

onde

$$q^k = r_b^k - (Z^k)^{-1}(r_1^k + U^k r_z^k) + (W^k)^{-1}(r_2^k + V^k r_w^k), \quad (3.4.7)$$

$$(\Theta^k)^{-1} = (Z^k)^{-1}U^k + (W^k)^{-1}V^k. \quad (3.4.8)$$

Usando a segunda equação do sistema (3.3.20) e a equação (3.4.6) temos o que denominamos *Sistema Aumentado*:

$$\begin{pmatrix} (\Theta^k)^{-1} & A \\ A^T & -\tau I_n \end{pmatrix} \begin{pmatrix} \Delta y^k \\ \Delta x^k \end{pmatrix} = \begin{pmatrix} q^k \\ -r_y^k \end{pmatrix}, \quad (3.4.9)$$

em analogia ao que é feito na Programação Linear. A matriz (3.4.9) de tamanho $(m+n) \times (m+n)$ é simétrica e indefinida.

As componentes Δu^k , Δv^k , Δz^k e Δw^k são obtidas de (3.4.1), (3.4.2), (3.4.3) e (3.4.4).

Conhecida na Programação Linear como *Sistema de Equações Normais*, a segunda forma de encontrar a direção de busca para o nosso método é dada pela primeira equação do sistema (3.4.9) obtemos

$$A^T \Theta^k A \Delta x^k = A^T \Theta^k q - A^T \Delta y^k. \quad (3.4.10)$$

Da segunda equação do sistema (3.4.9), temos que:

$$A^T \Delta y^k = -r_y^k + \tau \Delta x^k, \quad (3.4.11)$$

agora, substituindo (3.4.11) na equação (3.4.10), resulta em:

$$A^T \Theta^k A \Delta x^k = A^T \Theta^k q + (r_y^k - \tau \Delta x^k), \quad (3.4.12)$$

ou ainda

$$(A^T \Theta^k A + \tau I_n) \Delta x^k = A^T \Theta^k q + r_y^k. \quad (3.4.13)$$

Ao sistema de equações acima denominamos *Sistema de Equações Normais* na k -ésima iteração do MPI, este nome deve-se à semelhança com as equações normais para o problema de quadrados mínimos. A matriz do sistema (3.4.13) é simétrica e definida positiva de tamanho n .

Após obtermos Δx^k do Sistema de Equações Normais (3.4.13), o vetor Δy^k é dado por:

$$\Delta y^k = \Theta^k(q^k - A\Delta x^k), \quad (3.4.14)$$

e as componentes Δu^k , Δv^k , Δz^k e Δw^k são obtidas de (3.4.1), (3.4.2), (3.4.3) e (3.4.4).

3.4.2 Cálculo da direção de busca do PCM

Nesta subseção, faremos considerações similares as feitas na subseção (3.4.1), para o PCM. Para obter a direção de busca do PCM primal-dual, devemos resolver dois sistemas lineares, o sistema (3.3.23) para obter a direção preditora e o sistema (3.3.29) para obter a direção corretora e de centragem. É importante observar que ambos os sistemas têm a mesma matriz dos coeficientes, alterando somente o lado direito. Observe que o lado direito do sistema (3.3.29) necessita da direção preditora. Sendo assim, é necessário resolver dois sistemas mesmo que eles compartilhem a mesma matriz dos coeficientes. A partir daqui vamos buscar um Sistema Aumentado para a direção preditora. Repetindo os procedimentos da subseção anterior, mas desta vez para o sistema (3.3.23), chegamos ao sistema aumentado:

$$\begin{pmatrix} (\Theta^k)^{-1} & A \\ A^T & -\tau I_n \end{pmatrix} \begin{pmatrix} \Delta y^{af} \\ \Delta x^{af} \end{pmatrix} = \begin{pmatrix} \dot{q}^k \\ -r_y^k \end{pmatrix}, \quad (3.4.15)$$

novamente uma matriz de tamanho $(m+n) \times (n+n)$ simétrica e indefinida. Mas desta vez

$$\dot{q}^k = r_b^k + U^k(e - (Z^k)^{-1}r_z^k) + V^k(-e^k + (W^k)^{-1}r_w^k), \quad (3.4.16)$$

$$(\Theta^k)^{-1} = (Z^k)^{-1}(U^k) + (W^k)^{-1}V^k, \quad (3.4.17)$$

e as componentes Δu^{af} , Δv^{af} , Δz^{af} e Δw^{af} são dadas por:

$$\Delta z^{af} = -r_z^k - \Delta y^{af}, \quad (3.4.18)$$

$$\Delta w^{af} = \Delta y^{af} - r_w^k, \quad (3.4.19)$$

$$\Delta u^{af} = (Z^k)^{-1}(-U^k Z^k e - U^k \Delta z^{af}), \quad (3.4.20)$$

$$\Delta v^{af} = (W^k)^{-1}(-V^k W^k e - V^k \Delta w^{af}). \quad (3.4.21)$$

Por sua vez o Sistema de Equações normais tem a forma

$$(A^T \Theta^k A + \tau I_n) \Delta x^{af} = A^T \Theta^k \dot{q}^k + r_y^k. \quad (3.4.22)$$

A matriz do sistema (3.4.22) é simétrica e definida positiva de tamanho n . Após obtermos Δx^{af} a partir deste Sistema de Equações Normais, o vetor Δy^{af} é dado por:

$$\Delta y^{af} = \Theta^k (\dot{q}^k - A \Delta x^{af}), \quad (3.4.23)$$

e as componentes Δu^{af} , Δv^{af} , Δz^{af} e Δw^{af} são obtidas de (3.4.18), (3.4.19), (3.4.20) e (3.4.21).

Obtida a direção preditora, $d_k^{af} = (\Delta x^{af}, \Delta u^{af}, \Delta v^{af}, \Delta y^{af}, \Delta z^{af}, \Delta w^{af})$, através do Sistema Aumentado ou pelo Sistema de Equações Normais, a direção corretora e de centragem d_k^{cc} é determinada como segue:

Primariamente, use o vetor d_k^{af} para obter o lado direito do sistema (3.3.29), ou seja, para construir $\Delta U^{af} = \text{diag}(\Delta u^{af})$, $\Delta Z^{af} = \text{diag}(\Delta z^{af})$, $\Delta V^{af} = \text{diag}(\Delta v^{af})$ e $\Delta W^{af} = \text{diag}(\Delta w^{af})$. Repetindo os procedimentos da subseção anterior, mas desta vez para o sistema (3.3.29), chegamos ao sistema aumentado:

$$\begin{pmatrix} (\Theta^k)^{-1} & A \\ A^T & -\tau I_n \end{pmatrix} \begin{pmatrix} \Delta y^{cc} \\ \Delta x^{cc} \end{pmatrix} = \begin{pmatrix} \dot{q}^k \\ 0 \end{pmatrix}, \quad (3.4.24)$$

em que

$$\ddot{q} = (Z^k)^{-1} (-\sigma_k \mu^k e + \Delta U^{af} \Delta Z^{af} e) + (W^k)^{-1} (\sigma_k \mu^k e - \Delta V^{af} \Delta W^{af} e), \quad (3.4.25)$$

$$(\Theta^k)^{-1} = (Z^k)^{-1} U^k + (W^k)^{-1} V^k. \quad (3.4.26)$$

Obtidas as soluções Δx^{cc} e Δy^{cc} a partir do Sistema Aumentado dado por (3.4.24), as componentes Δu^{cc} , Δv^{cc} , Δz^{cc} e Δw^{cc} da direção de centragem $d_k^{cc} = (\Delta \mathbf{x}^{cc}, \Delta \mathbf{y}^{cc})$, são obtidas das equações

$$\Delta z^{cc} = -\Delta y^{cc}, \quad (3.4.27)$$

$$\Delta w^{cc} = \Delta y^{cc}, \quad (3.4.28)$$

$$\Delta u^{cc} = (Z^k)^{-1} (\sigma_k \mu^k e - \Delta U^{af} \Delta Z^{af} e + U^k \Delta y^{cc}), \quad (3.4.29)$$

$$\Delta v^{cc} = (W^k)^{-1} (\sigma_k \mu^k e - \Delta V^{af} \Delta W^{af} e - V^k \Delta y^{cc}). \quad (3.4.30)$$

Por sua vez, o sistema de equações normais tem a forma

$$(A^T \Theta^k A + \tau I_n) \Delta x^{cc} = A^T \Theta^k \ddot{q}^k. \quad (3.4.31)$$

Após obtermos Δx^{cc} do Sistema de Equações Normais (3.4.31), o vetor Δy^{cc} é dado por:

$$\Delta y^{cc} = \Theta^k (\ddot{q}^k - A \Delta x^{cc}), \quad (3.4.32)$$

e as componentes Δu^{cc} , Δv^{cc} , Δz^{cc} e Δw^{cc} são obtidas de (3.4.27), (3.4.28), (3.4.29) e (3.4.30). Para concluir, a direção de descida d^k na k -ésima iteração do método PCM é dada por $d_k = d_k^{af} + d_k^{cc}$.

Não obstante, na prática, ao invés de calcular d_k^{af} e d_k^{cc} e depois somá-los para obter d^k , após obter a direção preditora d_k^{af} , resolvemos o sistema linear:

$$\begin{pmatrix} A & I_m & -I_m & 0 & 0 & 0 \\ \tau I_n & 0 & 0 & -A^T & 0 & 0 \\ 0 & 0 & 0 & -I_m & -I_m & 0 \\ 0 & 0 & 0 & I_m & 0 & -I_m \\ 0 & Z^k & 0 & 0 & U^k & 0 \\ 0 & 0 & W^k & 0 & 0 & V^k \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta u \\ \Delta v \\ \Delta y \\ \Delta z \\ \Delta w \end{pmatrix} = \begin{pmatrix} r_b \\ r_y \\ r_z \\ r_w \\ \sigma_k \mu^k e - \Delta U^{af} \Delta Z^{af} e - U^k Z^k e \\ \sigma_k \mu^k e - \Delta V^{af} \Delta W^{af} e - V^k W^k e \end{pmatrix}, \quad (3.4.33)$$

que trata-se da soma dos sistemas (3.3.23) e (3.3.29). A solução do sistema (3.4.33) nos fornecerá d^k .

Ao tratar de problemas de grande porte, mesmo no caso de problemas esparsos, a resolução via métodos diretos do Sistema Aumentado ou de Equações Normais infelizmente demanda excessivo uso de memória, comprometendo a eficiência dos método diretos.

4 Sistema de Equações Normais

O Sistema de Equações Normais, $A^T\Theta A + \tau I_n$ é recorrente nos métodos apresentados no capítulo anterior. Em geral, mesmo quando A é esparsa, a matriz $A^T\Theta A$ é densa. Usando produto externo, observe que :

$$A^T\Theta A = \sum_{i=1}^m \theta_i A(i, :)^T A(i, :) = \sum_{i=1}^m \left(\frac{u_i}{z_i} + \frac{v_i}{w_i} \right)^{-1} A(i, :)^T A(i, :),$$

ou seja, basta que uma linha $A(i, :)$ seja densa para que a matriz $A^T\Theta A$ perca esparsidade e consequentemente $A^T\Theta A + \tau I_n$ também. Todavia, existem problemas em que a matriz $A^T\Theta A$ é mais esparsa que A . A fim de evitar produtos entre matrizes, para resolver este tipo de problema, são usados métodos iterativos, visto que estes necessitam apenas de produtos do tipo matriz por vetor. Para sistemas grandes com matrizes simétricas e definidas positivas, o Método dos Gradientes Conjugados Precondicionado (MGCP) é uma boa escolha. No entanto, a matriz diagonal Θ fica muito mal condicionada ao longo das iterações do MPI, o que causa o mal condicionamento da matriz $A\Theta A + \tau I_n$. Neste ponto, é importante ressaltar que o êxito do MGCP depende da escolha de um bom preconditionador, visto que é bem conhecido que este método é sensível ao número de condição da matriz do sistema linear, isto é, não tem um bom desempenho quando utilizado com matrizes mal condicionadas.

4.1 Análise espectral do Sistema de Equações Normais

Como já mencionado o Sistema de Equações Normais, $A^T\Theta A + \tau I_n$ é recorrente nos métodos apresentados. Fazendo $W = \Theta^{1/2}A$ estes sistemas podem ser reescritos como $\tau I_n + W^T W$. Feito isso, além da simetria destes sistemas, podemos verificar os seguintes resultados.

Proposição 1. *Seja W uma matriz de tamanho $m \times n$, então as matrizes $\tau I_n + W^T W$ e $\tau I_m + W W^T$ têm o mesmo espectro.*

Demonstração. Seja (λ, x) um autopar de $\tau I_m + W W^T$, então $(\lambda, W^T x)$ é um autopar da matriz $\tau I_n + W^T W$. Com efeito,

$$(\tau I_n + W^T W)W^T x = W^T(\tau I_m + W W^T)x = W^T(\lambda x) = \lambda(W^T x).$$

De maneira análoga, se (α, y) é um autopar de $\tau I_n + W^T W$ então (α, Wy) é um autopar da matriz $\tau I_m + WW^T$. De fato

$$(\tau I_m + WW^T)Wy = W(\tau I_n + W^T W)y = W(\alpha y) = \alpha(Wy).$$

□

Corolário 3. Se $\kappa(A)$ denota o número de condição de uma matriz arbitrária A na norma 2, então

$$\kappa(\tau I_n + W^T W) = \kappa(\tau I_m + WW^T),$$

onde W é uma matriz de tamanho $m \times n$.

Demonstração. Sejam λ_{max} e λ_{min} o maior e o menor autovalor da matriz $\tau I_n + W^T W$ respectivamente, da proposição anterior, λ_{max} e λ_{min} são também o maior e o menor autovalor de $(\tau I_m + WW^T)$. Visto que as matrizes $\tau I_n + W^T W$ e $\tau I_m + WW^T$ são simétricas, os seus autovalores são reais, e mais do que isso, o número de condição de ambas é dado por:

$$\kappa = \frac{\lambda_{max}}{\lambda_{min}}.$$

□

Para matriz $\tau I_n + W^T W$, temos que $\lambda_{min} \geq \tau$. De fato, se (λ, v) é um autopar de $\tau I_n + W^T W$, então $\tau v + W^T W v = \lambda v$. Multiplicando pela esquerda por v^T , tem-se $\tau \|v\|_2^2 + \|Wv\|_2^2 = \lambda \|v\|_2^2$. Logo,

$$\lambda = \tau + \frac{\|Wv\|_2^2}{\|v\|_2^2} \geq \tau. \quad (4.1.1)$$

Isto prova que os autovalores de $\tau I_n + W^T W$ e $\tau I_m + WW^T$ estão limitados inferiormente por τ . Lembre-se que estamos supondo $\tau > 0$, logo estes sistemas além de simétricos são definidos positivos.

Proposição 2. Os autovalores λ da matriz $\tau I_n + W^T W$ são da forma $\lambda = \tau + \alpha$ onde α é um autovalor de $W^T W$.

Demonstração. Seja (λ, v) um autopar de $\tau I_n + W^T W$, ou seja, $\tau v + W^T W v = \lambda v$. Então $W^T W v = (\lambda - \tau)v$, ou seja, (α, v) é um autopar de $W^T W$, onde $\alpha = \lambda - \tau$. □

Proposição 3. *Suponha que $A \in \mathbb{R}^{m \times n}$, onde*

$$\|A\|_2 = \begin{cases} \alpha_{max}, & \text{se } A = A^T \\ \sigma_{max}, & \text{se } A \neq A^T \end{cases},$$

onde α_{max} é o maior autovalor de A em módulo, e σ_{max} é o maior valor singular de A , sendo $\sigma_{max} = \sqrt{\alpha_{max}(A^T A)}$.

Uma demonstração da proposição acima, pode ser encontrada em [Golub e Loan \(1996\)](#).

Pela Proposição 2, temos que

$$\lambda_{max} = \tau + \alpha_{max} \quad (4.1.2)$$

onde α_{max} é o maior autovalor de $W^T W$. Por sua vez, da Proposição 3, fazendo $A = W$ vemos que $W \neq W^T$, logo $\|W\|_2 = \sigma_{max} = \sqrt{\alpha_{max}(W^T W)}$, ou ainda, $\alpha_{max}(W^T W) = \|W\|_2^2$. Equivalentemente

$$\alpha_{max} = \|W\|_2^2. \quad (4.1.3)$$

Das Proposições 2 e 3 temos que $\lambda_{max} = \tau + \|W\|_2^2$. E do Corolário 3, segue que $\kappa = \kappa(\tau I_n + W^T W) = \kappa(\tau I_m + W W^T) = \frac{\lambda_{max}}{\lambda_{min}}$. Logo, no caso particular em que $\tau \geq 1$ temos que:

$$\kappa = \frac{\lambda_{max}}{\lambda_{min}} \leq \lambda_{max} = \tau + \|W\|_2^2 \leq \tau + \|W\|_F^2 = \tau + \|W^T\|_F^2. \quad (4.1.4)$$

Lembrando que $W = \Theta^{1/2} A$, segue das equações (4.1.1)-(4.1.3) que

$$\tau \leq \lambda \leq \tau + \|\Theta^{1/2} A\|_2^2,$$

onde λ denota os autovalores da matriz $A^T \Theta A + \tau I_n$. Além disso, concluímos da Equação 4.1.4 que:

$$\kappa(A^T \Theta A + \tau I_n) \leq \tau + \|(\Theta^{1/2} A)^T\|_F^2,$$

ou seja o número de condição de $A^T \Theta A + \tau I_n$ depende da norma de $(\Theta^{1/2} A)^T$. Esta ferramenta de análise do número de condição do Sistema de Equações Normais é importante, visto que pretendemos usar o MGCP para resolver os Sistemas de Equações Normais, e como mencionado anteriormente, este método é sensível ao número de condição da matriz do sistema linear.

4.2 Abordagens para resolver o Sistema de Equações Normais

Nesta seção apresentamos a principal ferramenta que utilizamos na resolução dos sistemas de equações normais apresentados anteriormente, o Método dos Gradientes Conjugados Precondicionado.

4.2.1 Método dos Gradientes Conjugados

O Método dos Gradientes Conjugados (MGC) foi desenvolvido em 1952 por Hestenes e Stiefel, veja ([TREFETHEN; BAU, 1997](#)). Sistemas de equações lineares simétricos e definidos positivos podem ser resolvidos de forma eficiente por este método, desde que os autovalores da matriz do sistema estejam bem distribuídos. Afim de exemplificar o MGC, suponha que A é uma matriz simétrica definida positiva de ordem n , $b \in \mathbb{R}^n$, x_0 é um chute inicial dado e ϵ é a precisão requerida. O Algoritmo 4 nos mostra os passos do MGC.

Algoritmo 4 – Método dos Gradientes Conjugados

Dados: $x = x_0$, $k = 0$, $r_0 = b - Ax_0$, $\rho_0 = \|r_0\|_2^2$

Resultado: Solução do sistema $Ax = b$

enquanto $(\sqrt{\rho_k} > \epsilon \|b\|_2)$ e $(k < k_{\max})$ **faça**

$k = k + 1$;

se $k = 1$ **então**

$p_1 = r_0$;

senão

$\beta_k = r_{k-1}^T r_{k-1} / r_{k-2}^T r_{k-2}$;

$p_k = r_k + \beta_k p_{k-1}$;

fim

$w_k = Ap_k$;

$\alpha_k = r_{k-1}^T r_{k-1} / p_k^T w_k$;

$x_k = x_{k-1} + \alpha_k p_k$;

$r_k = r_{k-1} - \alpha_k w_k$;

$\rho_k = \|r_k\|_2^2$;

fim

$x = x_k$

Como pode ser visto no Algoritmo 4, o MGC gera uma sequência $\{x^k\}$ e utiliza a cada iteração apenas operações do tipo vetor por vetor e apenas um produto do tipo matriz por vetor que em geral é o passo mais caro do método. Em aritmética exata, é bem conhecido que, sob certas condições da distribuição de autovalores de A o método dos gradientes conjugados chega a solução do sistema linear em no máximo n iterações, sendo que a solução pode ser alcançada antes disso. Na verdade, segundo ([GOLUB; LOAN, 1996](#)), se A possuir r autovalores distintos, então o método de gradientes conjugados encontra a solução em no máximo r iterações.

Outro resultado útil para análise do comportamento do MGC é a seguinte desigualdade:

$$\|x^k - x^*\|_A \leq 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k \|x^0 - x^*\|_A, \quad (4.2.1)$$

onde a norma matricial $\|\cdot\|_A$ é definida por $\|w\|_A = \sqrt{w^T A w}$ para todo $w \in \mathbb{R}^n$, além disso, $k_2(A)$ denota o número de condição da matriz A na norma Euclidiana. Em geral, a precisão de $\{x^k\}$ é melhor que o indicado pelo limitante em (4.2.1). Segundo (TREFETHEN; BAU, 1997), à medida que $\sqrt{\kappa(A)} \rightarrow \infty$, tem-se

$$\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \simeq 1 - \frac{2}{\kappa(A)}.$$

Daí, se $\kappa(A)$ é grande, mas não muito grande, para tolerância especificada a convergência do método é da ordem de $\sqrt{\kappa(A)}$ iterações. Ou seja, pode-se dizer que o método é eficiente quando A é bem condicionada. Mais detalhes sobre a convergência do MGC podem ser encontrados em (KELLEY, 1999).

4.2.2 Precondicionamento

Métodos iterativos para resolver sistemas lineares podem ter convergência acelerada via precondicionamento da matriz do sistema linear. O precondicionamento de um sistema linear consiste em modificá-lo em um sistema linear com melhor condicionamento da matriz de coeficientes e cuja solução é facilmente obtida a partir da solução do sistema precondicionado. O precondicionamento pode ser efetuado pela esquerda, direita ou por ambos lados de um sistema $Ax = b$, como segue:

- $MAy = Mb; x = y;$
- $AMy = b; x = My;$
- $MAM^T z = Mb; x = M^T z.$

A matriz M é denominada *precondicionador*. É importante observar que $M^{-1}(MA)M = AM$, isto é, as matrizes MA e AM são matrizes equivalentes e consequentemente tem o mesmo espectro.

Note que, se $M = A$, temos $M^{-1}A = I_n$, donde $k(M^{-1}A) = 1$ e o método dos gradientes conjugados converge em apenas uma iteração. No entanto, isto é impraticável, visto que no Método dos Gradientes Conjugados Precondicionados que mostraremos em seguida, isto equivaleria a utilizar a própria matriz A como precondicionador e implicaria em resolver um sistema do tipo $Az = r$, o que não traz nenhuma vantagem em relação a resolver o sistema original. Se por outro lado, escolhermos o precondicionador *escala*

diagonal, dado por $M = \text{diag}(A)$, cujos elementos são aqueles da diagonal principal de A , mesmo em face da simplicidade deste preconditionador, podemos ter resultados bastante satisfatórios, como pode ser visto em (TREFETHEN; BAU, 1997). Observar estes dois preconditionadores nos dá uma visão do caminho a ser seguido na construção de preconditionadores adequados para o sistema $Ax = b$: um balanço entre a eficiência de $M = A$ e o baixo custo computacional de $M = \text{diag}(A)$.

No que segue, consideremos o Método dos Gradientes Conjugados Precondicionado (MGCP). Supondo que A é uma matriz simétrica definida positiva de ordem n , $b \in \mathbb{R}^n$, x_0 é um valor inicial dado, ϵ é a precisão requerida e M é um preconditionador que está disponível. O Algoritmo 5 nos mostra os passos do MGCP.

Algoritmo 5 – Método dos Gradientes Conjugados Precondicionado

Dados: $x = x_0$, $k = 0$, $r_0 = b - Ax_0$, $\rho_0 = \|r_0\|_2^2$

Resultado: Solução do sistema $Ax = b$

enquanto ($\sqrt{\rho_k} > \epsilon \|b\|_2$) e ($k < k_{\max}$) **faça**

 Resolva $Mz_k = r_k$;

$k = k + 1$;

se $k = 1$ **então**

$p_1 = z_0$;

senão

$\beta_k = r_{k-1}^T z_{k-1} / r_{k-2}^T z_{k-2}$;

$p_k = z_{k-1} + \beta_k p_{k-1}$;

fim

$w_k = Ap_k$;

$\alpha_k = r_{k-1}^T z_{k-1} / p_k^T w_k$;

$x_k = x_{k-1} + \alpha_k p_k$;

$r_k = r_{k-1} - \alpha_k w_k$;

$\rho_k = \|r_k\|_2^2$;

fim

$x = x_k$

Para que o MGCP seja eficiente é necessário que o sistema linear $Mz = r$ possa ser facilmente resolvido. Ou seja, é fundamental o papel exercido pelo preconditionador M para convergência do método. A dificuldade, quando se trata de preconditionamento, é a construção de preconditionadores que demandem baixo custo computacional para sua construção e que ao mesmo tempo sejam eficientes em acelerar os métodos iterativos. Segundo (BENZI, 2002) o componente mais intrínseco na implementação eficiente de problemas da computação científica é o preconditionamento.

Neste trabalho utilizamos o MGCP detalhado no Algoritmo 5 para resolver os sistemas de equações normais oriundos dos Métodos de Pontos Interiores desenvolvidos no

capítulo anterior. Estes sistemas serão preconditionados da seguinte forma:

$$M^{-1}(A^T \Theta^k A + I_n) M^{-T} \Delta \bar{x}^k = M^{-1}(A^T \Theta^k q^k + r_y^k). \quad (4.2.2)$$

onde M é o preconditionador e $\Delta \bar{x}^k = M^T \Delta x^k$.

4.2.3 Fatoração Controlada de Cholesky

Uma das estratégias para construir a matriz M baseia-se na Fatoração Controlada de Cholesky (FCC), proposta por (CAMPOS, 1995). Vamos considerar a fatoração de Cholesky e a fatoração incompleta de Cholesky da matriz $A^T \Theta A + \tau I_n$, ou seja:

$$LL^T = A^T \Theta A + \tau I_n = \tilde{L} \tilde{L}^T + R$$

onde L é o fator de Cholesky, \tilde{L} é a matriz da fatoração incompleta e R é matriz do resíduo. Se $E = L - \tilde{L}$, temos

$$\tilde{L}(A^T \Theta A + \tau I_n) \tilde{L}^{-T} = (\tilde{L}^{-1} L)(\tilde{L}^{-1} L)^T = (I_n + \tilde{L}^{-1} E)(I_n + \tilde{L}^{-1} E)^T.$$

À medida que \tilde{L} se aproxima de L , a matriz E tende à matriz nula, e conseqüentemente, pela última igualdade acima, $\tilde{L}(A^T \Theta A + \tau I_n) \tilde{L}^{-T}$ se aproxima da identidade I_n . Note que

$$\begin{aligned} R &= LL^T - \tilde{L} \tilde{L}^T \\ &= LL^T - L \tilde{L}^T + L \tilde{L}^T - \tilde{L} \tilde{L}^T \\ &= L(L^T - \tilde{L}^T) + (L - \tilde{L}) \tilde{L}^T \\ &= LE^T + E \tilde{L}^T, \end{aligned} \quad (4.2.3)$$

logo, se $\|E\|_F$ se aproxima de zero, segue da igualdade (4.2.3) que $\|R\|_F$ se aproxima de zero. Este raciocínio motiva a construção do preconditionador FCC tomando como base a minimização da norma de Frobenius da matriz E . Isto nos leva ao problema:

$$\min \|E\|_F^2. \quad (4.2.4)$$

Se definirmos $c_j = \sum_{i=1}^n |l_{ij} - \tilde{l}_{ij}|^2$, para $j = 1, \dots, n$, resolver o problema

$$\min \sum_{j=1}^n c_j \quad (4.2.5)$$

equivale a resolver o problema (4.2.4). Além disso, podemos reescrever o problema (4.2.5) como:

$$\min \sum_{j=1}^n \left(\sum_{k=1}^{n_j+\eta} |l_{ikj} - \tilde{l}_{ikj}|^2 + \sum_{k=m_j+\eta+1}^n |l_{ikj}|^2 \right), \quad (4.2.6)$$

Nesta equação, n_j é o número de componentes não nulos abaixo da diagonal na j -ésima coluna da matriz $A^T \Theta A + \tau I_n$, η representa o número de elementos extras permitido por coluna e n é a ordem da matriz $A^T \Theta A + \tau I_n$. Vamos resolver o problema (4.2.4) utilizando os seguintes passos:

1. Aumentamos o fator η , permitindo mais preenchimento. Desta forma, c_j decresce, pois a primeira soma em (4.2.6) contém mais elementos.
2. Para cada η fixo escolhemos os maiores $n_j + \eta$ elementos de \tilde{L} em valor absoluto.
3. Uma coluna do preconditionador é calculada por vez sendo armazenados os maiores valores em valor absoluto.

Por conseguinte, apresentamos as principais características do preconditionador FCC:

1. A FCC não considera o padrão de esparsidade da matriz original. Os elementos são escolhidos por valor, não por posição.
2. O preconditionador FCC pode ser considerado uma generalização do preconditionador proposto por (JONES; PLASSMANN, 1995) que escolhe um número fixo de elementos não nulos em cada linha ou coluna.
3. A FCC, assim como as fatorações incompletas, não está livre da presença de pivôs muito pequenos ou não positivos. Este problema é contornado utilizando-se um incremento exponencial. Segundo (BOCANEGRA; CAMPOS; OLIVEIRA, 2007), esse incremento pode ser calculado da seguinte forma: quando um elemento da diagonal é muito pequeno ou não positivo, pode-se incrementar na diagonal principal da matriz, o valor dado por $\alpha_i = 5,0 \times 10^{-4} 2^{i-1}$, para $i = 1, \dots, n$ e recomeça a fatoração. A fatoração poderá ser reiniciada muitas vezes, podendo até mesmo tornar inviável a resolução do mesmo.
4. Armazenamento previsível. Cada uma das matrizes é armazenada em três vetores: um vetor de dimensão $n + 1$ para os ponteiros das colunas, um vetor de inteiros de dimensão n para os índices das linhas e um vetor de dimensão n para os elementos. Outro vetor é usado como espaço de trabalho para construção de \tilde{L} . Como apenas

$n_j + \eta$ elementos são mantidos em cada coluna j do preconditionador, o espaço de armazenamento necessário para o preconditionador é previsível e controlável.

5. A FCC é um preconditionador versátil visto que o parâmetro η pode variar de $-n$ até n . Se $\eta = -n$ obtemos preconditionador escala diagonal. No caso de $\eta = 0$, obtemos um fator de Cholesky aproximado \tilde{L} que tem a mesma quantidade de armazenamento da fatoração de Cholesky incompleta sem preenchimentos, visto que o padrão de esparsidade da matriz original é ignorado e apenas os maiores fatores não nulos, obtidos com a fatoração, são mantidos no fator incompleto. Por fim, se $\eta = n$, temos o fator de Cholesky completo L e, conseqüentemente, o armazenamento do preconditionador pode ser maior do que o da matriz original.

A eficiência do preconditionador FCC depende do valor inicial e final de η sendo que denotamos estes respectivamente por η_0 e η_{max} . Estes valores determinam o máximo de preenchimento permitido em cada coluna do preconditionador. Em problemas de PL, na primeira iteração, o número de entradas não nulas permitidas em cada coluna depende da esparsidade do problema a ser resolvido, daí η_0 é dado pela relação:

$$\eta_0 = \begin{cases} -\frac{|A^T\Theta A|}{n}, & \text{se } \frac{|A^T\Theta A|}{n} > 10, \\ \frac{|A^T\Theta A|}{n}, & \text{caso contrário.} \end{cases} \quad (4.2.7)$$

Sendo assim, da igualdade (4.2.7), $\eta_0 < 0$ se a matriz $A^T\Theta A$ é relativamente densa e $0 \leq \eta_0 < 10$ caso a matriz $A^T\Theta A$ seja esparsa. Ainda é alvo de estudo valores para η_{max} que dependam dos dados do problema de PL. No entanto, esta estratégia para determinação de η_0 não apresentou bons resultados quando utilizado em conjuntos com os métodos apresentados no Capítulo 2. As matrizes no qual aplicamos os métodos desenvolvidos no Capítulo 2 são em geral densas e extremamente mal condicionadas. A densidade dessas matrizes induz o preconditionador escala diagonal, e a ineficiência deste por sua vez, induz várias atualizações do preconditionador no decorrer das iterações dos MPI's, tornando estes métodos demasiadamente lentos. Em vista disso, adotamos a seguinte estratégia: escolhemos $\eta_0 = 0$ o que implica que o fator \tilde{L} é equivalente a fatoração incompleta de Cholesky proposta por (JONES; PLASSMANN, 1995). Por sua vez tomamos $\eta_{max} = n/4$, sendo que $\eta_k = \eta_{k-1} + 10$ cada vez que é necessário atualizar o preconditionador, ou seja, permitimos que o preconditionador FCC seja atualizado no máximo $n/4$ vezes no decorrer dos MPI's, e somente quando o número de iterações do MGCP excede $n/4$. A escolha dos valores de η_k segundo esta estratégia se mostraram mais promissores quando comparados a estratégia adotada para problemas de PL.

4.2.4 Precondicionador Separador

A segunda estratégia de preconditionamento do sistema de equações normais baseia-se na fatoração LU. A matriz dos coeficientes do sistema de equações normais

$$(A^T \Theta A + \tau I_n) \Delta x = A^T \Theta q + r_y$$

pode ser reescrita da seguinte maneira

$$[A^T \ I_n] \begin{bmatrix} \Theta & 0 \\ 0 & \tau I_n \end{bmatrix} \begin{bmatrix} A \\ I_n \end{bmatrix}. \quad (4.2.8)$$

Seja P uma matriz de permutação tal que $P[A^T \ I_n]^T = [A_1^T \ A_2^T]^T$ onde $A_1 \in \mathbb{R}^{n \times n}$, $A_2 \in \mathbb{R}^{m \times n}$ e de tal modo que se tenha $A_1 = LU$. Então o sistema (4.2.8) pode ser reescrito na forma

$$[A_1^T \ A_2^T] \begin{bmatrix} \Theta_1 & 0 \\ 0 & \Theta_2 \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} = A_1^T \Theta_1 A_1 + A_2^T \Theta_2 A_2, \quad (4.2.9)$$

em que

$$P \begin{bmatrix} \Theta & 0 \\ 0 & \tau I_n \end{bmatrix} P^T = \begin{bmatrix} \Theta_1 & 0 \\ 0 & \Theta_2 \end{bmatrix}.$$

Isto posto, propomos que o novo preconditionador para o sistema de equações normais seja $M^{-1} = \Theta_1^{-1/2} A_1^{-T}$. Aplicando este preconditionador ao sistema de equações normais com a disposição dada por (4.2.9), obtemos

$$M^{-1}(A^T \Theta A + \tau I_n)M^{-T} = I_n + \Theta_1^{-1/2} A_1^{-T} A_2^T \Theta_2 A_2 A_1^{-1} \Theta_1^{-T/2} = I_n + W^T W \quad (4.2.10)$$

onde $W = \Theta_2^{1/2} A_2 A_1^{-1} \Theta_1^{-T/2}$.

A matriz preconditionada obtida é positiva definida e tem autovalores maiores ou iguais a 1, ou seja, não tem autovalores próximos a zero. Sendo assim, se escolhermos a matriz de permutação P de tal modo que Θ_1^{-1} e Θ_2 se aproximam da matriz nula a medida que nos aproximamos de uma solução ótima, a matriz (4.2.10) se aproxima da matriz identidade, e conseqüentemente tanto os maiores autovalores dessa matriz como seu número de condição se aproximam do número 1. Embora possa não ser simples encontrar uma matriz de permutação que nos forneça essa disposição adequada, calcular a fatoração LU da matriz A_1 requer menos esforço do que calcular a fatoração de Cholesky de $A_1^T A_1$. Para detectar as colunas de A_1 executamos uma fatoração LU da matriz $[A^T \ I_n]^T$. Esta abordagem de preconditionamento é baseada no preconditionador separador proposto por (OLIVEIRA; SORENSEN, 2005).

5 Resultados Numéricos

Para verificar a eficiência dos métodos de pontos interiores desenvolvidos no Capítulo 3, neste capítulo aplicamos estes métodos à resolução de alguns exemplos. Nomeamos cada método segundo o preconditionador utilizado por este da seguinte maneira:

1. MPIFCC: Método dos pontos interiores preconditionado com a fatoração controlada de Cholesky;
2. MPILU: Método dos pontos interiores preconditionado com o Preconditionador Separador;
3. MPCFCC: Método preditor-corretor preconditionado com a fatoração controlada de Cholesky;
4. MPCLU: Método preditor-corretor preconditionado com o Preconditionador Separador;

Todos os resultados numéricos foram desenvolvidos em MATLAB R2013b com sistema operacional 64-bit Windows 10, processador Intel Core I7-8550U, 1.99 Ghz, 16 GB de memória RAM. A maioria dos códigos para calcular a discretização dos exemplos desta tese provêm dos pacotes disponibilizados por (HANSEN, 2007) e (GAZZOLA; HANSEN; NAGY, 2019).

5.1 Hilbert

Começamos por retomar o exemplo tratado nas subseções (2.2.2) e (2.2.3), a matriz de Hilbert. Porém, desta vez consideramos que o lado direito do sistema está contaminado por ruídos, ou seja, $b = \bar{b} + \mathbf{e}$, em que \mathbf{e} refere-se a um vetor aleatório normalizado escolhido de tal forma que $\|\mathbf{e}\|/\|\bar{b}\| = 10^{-2}$. Neste e nos próximos exemplos iremos nos referir ao quociente $NL = \|\mathbf{e}\|/\|\bar{b}\|$ como nível de ruído. Aplicamos os métodos MPIFCC, MPILU, MPCFCC e MPCLU a este exemplo e apresentamos os resultados nas Tabelas 1, 2, 3 e 4.

Tabela 1 – MPIFCC aplicado à matriz de Hilbert.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
250	13	194	$5,190 \times 10^{-2}$	$2,400 \times 10^{-3}$	$3,854 \times 10^{-1}$
500	19	208	$3,740 \times 10^{-2}$	$1,000 \times 10^{-3}$	$7,531 \times 10^{-1}$
1000	24	281	$5,640 \times 10^{-2}$	$1,100 \times 10^{-3}$	$3,681 \times 10^{+0}$
2000	28	830	$5,184 \times 10^{-2}$	$7,920 \times 10^{-4}$	$3,647 \times 10^{+1}$
5000	44	5655	$4,310 \times 10^{-2}$	$4,217 \times 10^{-4}$	$1,898 \times 10^{+3}$

Tabela 2 – MPILU aplicado à matriz de Hilbert.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
250	17	172	$9,580 \times 10^{-2}$	$1,000 \times 10^{-3}$	$2,512 \times 10^{-1}$
500	18	180	$6,439 \times 10^{-2}$	$4,294 \times 10^{-4}$	$5,856 \times 10^{-1}$
1000	23	256	$3,490 \times 10^{-2}$	$2,879 \times 10^{-4}$	$2,738 \times 10^{+0}$
2000	30	328	$5,210 \times 10^{-2}$	$3,835 \times 10^{-4}$	$1,240 \times 10^{+1}$
5000	43	496	$4,310 \times 10^{-2}$	$4,228 \times 10^{-4}$	$6,584 \times 10^{+1}$

Tabela 3 – MPCFCC aplicado à matriz de Hilbert.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
250	7	150	$7,166 \times 10^{-2}$	$1,774 \times 10^{-3}$	$4,354 \times 10^{-1}$
500	9	217	$5,486 \times 10^{-2}$	$5,703 \times 10^{-4}$	$8,839 \times 10^{-1}$
1000	9	241	$7,826 \times 10^{-2}$	$6,598 \times 10^{-4}$	$3,729 \times 10^{+0}$
2000	11	316	$9,933 \times 10^{-2}$	$8,382 \times 10^{-4}$	$2,840 \times 10^{+1}$
5000	11	345	$4,110 \times 10^{-2}$	$3,091 \times 10^{-4}$	$6,492 \times 10^{+2}$

Tabela 4 – MPCLU aplicado à matriz de Hilbert.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
250	11	157	$5,280 \times 10^{-2}$	$9,149 \times 10^{-4}$	$2,717 \times 10^{-1}$
500	12	192	$3,740 \times 10^{-2}$	$6,362 \times 10^{-4}$	$5,810 \times 10^{-1}$
1000	13	196	$5,760 \times 10^{-2}$	$5,989 \times 10^{-4}$	$2,460 \times 10^{+0}$
2000	12	196	$2,830 \times 10^{-2}$	$1,822 \times 10^{-4}$	$1,213 \times 10^{+1}$
5000	14	232	$4,440 \times 10^{-2}$	$4,216 \times 10^{-4}$	$9,459 \times 10^{+1}$

Nas Tabelas 1, 2, 3 e 4 temos respectivamente os dados do MPIFCC, MPILU, MPCFCC e do MPCLU aplicados a matrizes de Hilbert de diferentes dimensões. A primeira coluna das tabelas nos informa a dimensão das matrizes de Hilbert, a segunda coluna fornece o número de iterações de métodos de pontos interiores que foram necessárias para resolver o problema, ao passo que a terceira fornece o número de iterações do MGCP necessárias durante toda a execução do método de pontos interiores em questão. Uma vez que a solução exata \mathbf{x} é conhecida, as colunas Erd e Eri representam respectivamente os erros relativos cometidos: $Erd = \|\mathbf{x}^k - \mathbf{x}\|_2 / \|\mathbf{x}\|_2$ e $Eri = \|A\mathbf{x}^k - b\|_2 / \|b\|_2$. Por fim, t_{cpu} representa o tempo em segundos demandado pelo método para resolução do problema.

Ao compararmos as Tabelas 1 e 2, observarmos que a ordem de precisão dos resultados é praticamente o mesmo para ambos os métodos. No que diz respeito ao número de iterações de MPI, os resultados estão próximos. Uma discrepância da eficiência entre esses dois métodos pode ser notada quando aplicamos os métodos a matriz de Hilbert de ordem $n = 5000$. Essa discrepância se evidencia quando comparamos o número de iterações de MGCP e o tempo computacional t_{cpu} destes métodos para ordem $n = 5000$. Note que em todos os casos o número iterações de MGCP necessárias para atingir a solução foi maior para o MPIFCC.

Os resultados obtidos pelo MPCFCC expostos na Tabela 3 mostram que este método teve um número menor de iterações de MPI do que os métodos MPIFCC e MPILU além de ter alcançado a mesma ordem de precisão destes métodos. No entanto, quando tomamos a dimensão da matriz de Hilbert como $n = 5000$ podemos novamente notar uma discrepância entre os métodos MPIFCC, MPILU e MPCFCC no tempo de execução computacional t_{cpu} . Para esta dimensão, novamente o método MPILU se mostrou mais eficiente. No entanto, o MPCFCC necessitou de um número menor de iterações de MGCP do que o MPILU.

Por fim, consideremos a Tabela 4 referente ao método MPCLU. Com ordem de precisão semelhante aos resultados das tabelas anteriores, este método também apresentou um número menor de iterações de MPI do que os métodos MPIFCC e MPILU. Na maioria dos casos este método também precisou de menos iterações de MGCP para alcançar a solução do que os métodos MPIFCC e MPILU, mesmo para $n = 5000$. Apesar disso, para $n = 5000$, o MPILU ainda obteve melhor tempo computacional do que o MPCLU.

5.2 Baart

Prosseguindo com os exemplos, a equação integral de Fredholm de primeiro tipo

$$\int_0^{\pi} \exp(s \cos(t)x(t))x(t)dt = 2\frac{\text{sen}(s)}{s}, \quad 0 \leq s \leq \frac{\pi}{2},$$

discutida por (BAART, 1982), tem solução $x(t) = \text{sen}(t)$. Esta equação integral é discretizada por um método de Galerkin usando a função Baart de (HANSEN, 2007) que gera uma matriz não simétrica e mal condicionada que iremos nos referir como matriz de Baart. Uma pequena perturbação do lado direito deste sistema pode causar uma grande perturbação na solução. Em (PARK, 2019), este problema foi resolvido por um método baseado em SVD para ordem $n = 2000$ com $b = \bar{b} + \mathbf{e}$, em que $NL = 10^{-3}$. O erro relativo obtido em (PARK, 2019) foi de $1,1 \times 10^{-1}$. As Tabelas 5, 6, 7 e 8 mostram os resultados obtidos por nossos métodos.

A ordem de precisão obtida pelos métodos MPIFCC e MPCFCC para $n = 2000$ é a mesma obtida em (PARK, 2019). Já os métodos MPILU e MPCLU apresentaram uma sutil melhora na precisão para esta dimensão.

Tabela 5 – MPIFCC aplicado à matriz de Baart.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
250	15	107	$2,210 \times 10^{-1}$	$5,669 \times 10^{-4}$	$3,275 \times 10^{-1}$
500	23	151	$8,701 \times 10^{-2}$	$8,393 \times 10^{-5}$	$7,315 \times 10^{-1}$
1000	24	149	$9,032 \times 10^{-2}$	$7,535 \times 10^{-5}$	$3,440 \times 10^{+0}$
2000	26	254	$1,381 \times 10^{-1}$	$2,393 \times 10^{-5}$	$2,872 \times 10^{+1}$
5000	100	10005	$9,960 \times 10^{-2}$	$2,455 \times 10^{-5}$	$3,432 \times 10^{+3}$

Tabela 6 – MPILU aplicado à matriz de Baart.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
250	32	269	$1,476 \times 10^{-1}$	$2,303 \times 10^{-4}$	$3,051 \times 10^{-1}$
500	39	366	$1,095 \times 10^{-1}$	$6,602 \times 10^{-5}$	$9,754 \times 10^{-1}$
1000	41	688	$9,740 \times 10^{-2}$	$1,001 \times 10^{-4}$	$5,761 \times 10^{+0}$
2000	69	1062	$7,290 \times 10^{-2}$	$2,846 \times 10^{-5}$	$3,985 \times 10^{+1}$
5000	68	913	$1,000 \times 10^{-1}$	$2,839 \times 10^{-5}$	$1,730 \times 10^{+1}$

Tabela 7 – MPCFCC aplicado à matriz de Baart.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
250	8	148	$1,158 \times 10^{-1}$	$2,403 \times 10^{-4}$	$2,809 \times 10^{-1}$
500	9	129	$1,201 \times 10^{-1}$	$1,635 \times 10^{-4}$	$6,022 \times 10^{-1}$
1000	10	131	$7,760 \times 10^{-2}$	$5,332 \times 10^{-5}$	$2,841 \times 10^{+0}$
2000	10	131	$1,058 \times 10^{-1}$	$4,386 \times 10^{-4}$	$2,312 \times 10^{+1}$
5000	11	147	$1,101 \times 10^{-1}$	$2,134 \times 10^{-5}$	$5,811 \times 10^{+2}$

Tabela 8 – MPCLU aplicado à matriz de Baart.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
250	9	148	$1,145 \times 10^{-1}$	$2,204 \times 10^{-4}$	$4,127 \times 10^{-1}$
500	11	158	$1,300 \times 10^{-1}$	$1,657 \times 10^{-4}$	$5,723 \times 10^{-1}$
1000	10	164	$1,573 \times 10^{-1}$	$6,239 \times 10^{-5}$	$1,742 \times 10^{+0}$
2000	12	201	$7,130 \times 10^{-2}$	$3,557 \times 10^{-5}$	$8,885 \times 10^{+0}$
5000	10	165	$8,060 \times 10^{-2}$	$3,120 \times 10^{-5}$	$5,365 \times 10^{+1}$

De um modo geral, para este exemplo, o MPILU precisou de um maior número de iterações de MPI e de MGCP do que o MPIFCC para convergir, exceto para $n = 5000$. Para $n = 5000$ o método MPILU também se sobressaiu ao método MPIFCC, no que diz respeito ao tempo computacional t_{cpu} . Quando comparamos o método MPILU com o método MPCFCC, percebemos que em todos os casos o número de iterações de MPI e o número de iterações de MGCP do MPCFCC foram menores que os apresentados pelo método MPILU. Também, quando comparamos os tempos de execução computacional t_{cpu} do MPILU com MPCFCC, o método MPCFCC só não se mostrou melhor para $n = 5000$, o que se deve ao fato do preconditionador ter necessitado de duas atualizações.

Nos requisitos iterações de MPI e iterações de MGCP o método MPCFCC se mostrou competitivo com o método MPCLU, ele também se mostrou competitivo no quesito tempo computacional, exceto para ordem $n = 5000$.

Embora os resultados obtidos não sejam tão precisos, para $n = 2000$, esses são coerentes com os apresentados por (PARK, 2019). Em (REICHEL; SADOK; SHYSHKOV, 2007) este problema foi resolvido para $n = 500$ e também apresentou precisão equivalente à obtida por nossos métodos. Estes resultados se devem ao alto mal-condicionamento desta

matriz, o que implica que, mesmo sem qualquer perturbação do lado direito do sistema, os resultados não seriam tão satisfatórios. No entanto, apesar das dificuldades envolvidas na resolução deste sistema, este nos permite mostrar uma das vantagens alcançadas pelo método MPCLU. Para $n = 18000$ e para as especificações de máquina enunciadas no início deste capítulo, não foi possível calcular a decomposição SVD da matriz de Baart devido a limitações de memória da máquina, ou seja, caso fosse necessário, métodos baseados em SVD não seriam capazes de resolver este problema. Por outro lado, o método MPCLU foi capaz de resolvê-lo obtendo os resultados da Tabela 9.

Tabela 9 – MPCLU aplicado à matriz de Baart de ordem $n=18000$.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
18000	12	140	$1,536 \times 10^{-1}$	$7,415 \times 10^{-4}$	$2,981 \times 10^3$

Note que, apesar do tamanho da matriz, de sua densidade e de seu mal condicionamento, o MPCLU precisou de poucas iterações de MPI e de MGCP para alcançar um resultado coerente com a precisão dos demais resultados apresentados.

5.3 Blur

O problema teste de desfocagem de imagem surge em conexão com a degradação de imagens digitais por turbulência atmosférica, modelada por uma função Gaussiana de espalhamento pontual

$$h(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right).$$

Para discretizá-lo utilizamos a função Blur (HANSEN, 2007). A matriz A gerada na discretização deste problema é uma matriz simétrica duplamente Toeplitz $N^2 \times N^2$ armazenada em formato esparsa. O vetor x é uma versão empilhada em colunas de uma imagem de teste simples gerada pela própria função Blur, enquanto b mantém uma versão empilhada em colunas da imagem borrada, ou seja, $b = Ax$. Mais detalhes sobre este problema podem ser encontrados em (HANKE; HANSEN, 1993).

As Tabelas 10, 11, 12 e 13 exibem os resultados obtidos por nossos métodos para o problema de desfocagem de imagem.

Tabela 10 – MPIFCC aplicado à matriz de Blur.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
256	8	258	$5,421 \times 10^{-11}$	$3,621 \times 10^{-12}$	$2,052 \times 10^{-1}$
576	8	470	$1,091 \times 10^{-09}$	$6,874 \times 10^{-11}$	$4,730 \times 10^{-1}$
1024	9	689	$4,609 \times 10^{-11}$	$2,716 \times 10^{-12}$	$2,771 \times 10^{+0}$
2116	9	682	$6,774 \times 10^{-08}$	$3,886 \times 10^{-09}$	$7,052 \times 10^{+0}$
5184	10	2023	$3,227 \times 10^{-09}$	$1,755 \times 10^{-10}$	$1,045 \times 10^{+2}$

Tabela 11 – MPILU aplicado à matriz de Blur.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
256	8	167	$5,420 \times 10^{-11}$	$3,621 \times 10^{-12}$	$6,000 \times 10^{-2}$
576	8	211	$1,091 \times 10^{-09}$	$6,874 \times 10^{-11}$	$1,384 \times 10^{-1}$
1024	9	240	$4,531 \times 10^{-11}$	$2,670 \times 10^{-12}$	$7,192 \times 10^{-1}$
2116	9	278	$6,779 \times 10^{-08}$	$3,889 \times 10^{-09}$	$8,152 \times 10^{-1}$
5184	10	345	$3,227 \times 10^{-09}$	$1,755 \times 10^{-10}$	$5,283 \times 10^{+0}$

Tabela 12 – MPCFCC aplicado à matriz de Blur.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
256	5	276	$8,760 \times 10^{-10}$	$6,251 \times 10^{-11}$	$3,205 \times 10^{-1}$
576	5	450	$2,137 \times 10^{-09}$	$1,033 \times 10^{-10}$	$8,381 \times 10^{-1}$
1024	6	700	$3,684 \times 10^{-12}$	$1,611 \times 10^{-13}$	$2,500 \times 10^{+0}$
2116	7	1014	$6,158 \times 10^{-11}$	$2,776 \times 10^{-12}$	$1,220 \times 10^{+1}$
5184	7	1796	$5,302 \times 10^{-09}$	$3,833 \times 10^{-10}$	$1,357 \times 10^{+2}$

Tabela 13 – MPCLU aplicado à matriz de Blur.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
256	5	190	$6,633 \times 10^{-10}$	$4,839 \times 10^{-11}$	$8,890 \times 10^{-2}$
576	5	244	$1,880 \times 10^{-09}$	$9,090 \times 10^{-11}$	$1,490 \times 10^{-1}$
1024	5	276	$7,203 \times 10^{-08}$	$3,150 \times 10^{-09}$	$2,479 \times 10^{-1}$
2116	6	333	$5,659 \times 10^{-11}$	$2,578 \times 10^{-12}$	$8,480 \times 10^{-1}$
5184	6	424	$5,288 \times 10^{-09}$	$3,815 \times 10^{-10}$	$5,287 \times 10^{+0}$

No que diz respeito à precisão, os quatro métodos alcançaram resultados bastante satisfatórios. Por outro lado, quando observamos o número de iterações de MPI necessários para obtenção da solução, percebemos que os métodos MPCFCC e MPCLU tiveram uma leve vantagem sobre os outros dois métodos, o que é relevante dada a quantidade de processamento envolvido em uma iteração de ponto interior. No entanto, os métodos MPILU e MPCLU necessitaram de menos iterações de MGCP do que os métodos MPIFCC e MPCFCC para alcançar a solução. Essa vantagem do MPILU e do MPCLU também se refletiu no tempo computacional t_{cpu} quando comparado aos outros dois métodos. Quando comparamos os métodos MPILU e MPCLU, também percebemos que não há muita diferença entre seus tempos computacionais t_{cpu} para as dimensões dadas. Nas Figuras (2a), (2b) e (2c) temos um comparativo entre uma imagem de teste original, sua imagem borrada pela função Blur e a imagem recuperada por nosso método MPILU. Como pode ser observado, a solução obtida é muito próxima da solução original.

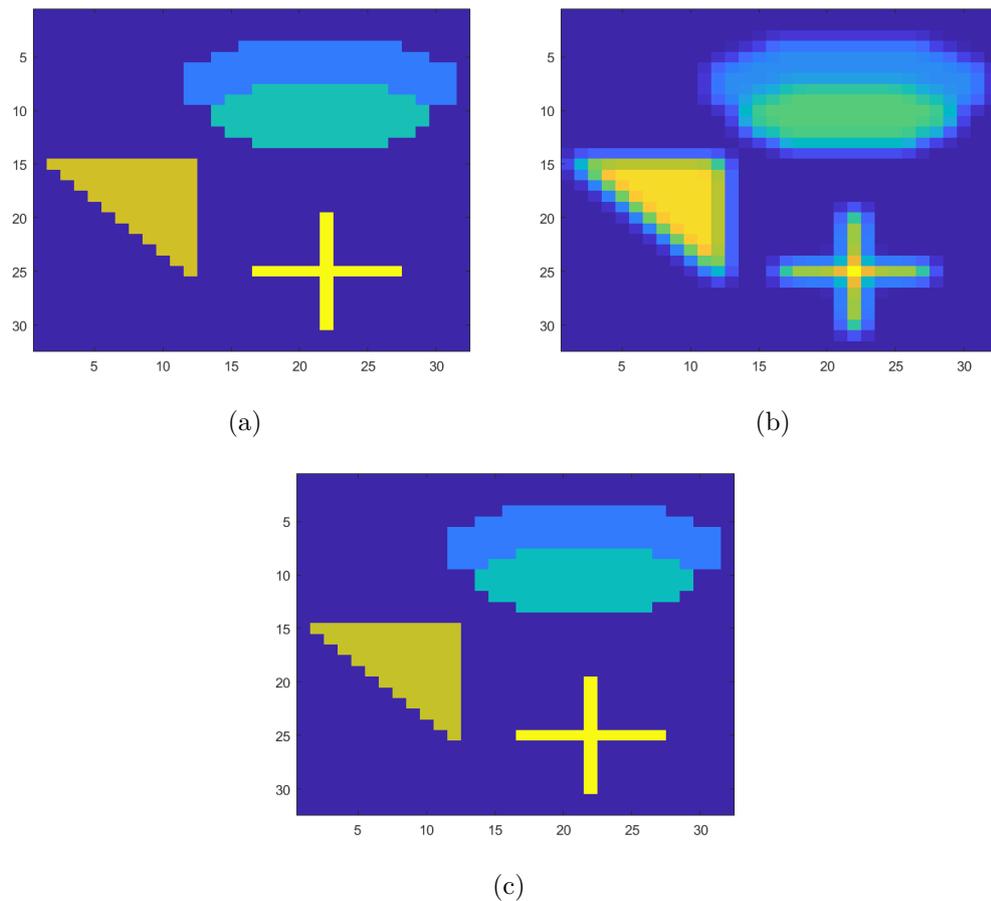


Figura 2 – Problema de teste Blur: (a) Imagem de teste original, (b) Imagem de teste borrada, (c) Imagem recuperada pelo MPILU.

5.4 Shaw

Considere agora a equação integral de Fredholm

$$\int_{-\pi/2}^{\pi/2} K(s, t)x(t)dt = g(s), \quad -\pi/2 \leq s, t \leq \pi/2,$$

que representa um modelo de reconstrução de imagem unidimensional. O núcleo K e a solução $x(t)$ são dados respectivamente por

$$\begin{aligned} K(s, t) &= (\cos(s) + \cos(t))^2 \left(\frac{\text{sen}(u)}{u} \right)^2 \\ u &= \pi(\text{sen}(s) + \text{sen}(t)) \\ x(t) &= 2 \exp(-6(t - 0, 8)^2) + \exp(-2(t + 0, 5)^2). \end{aligned}$$

A discretização deste problema pelo método de Nyström baseado na regra de quadratura de ponto médio é obtida usando a função shaw em (HANSEN, 2007). A matriz resultante é simétrica, indefinida e numericamente singular, além de severamente

mal condicionada, veja (SHAW, 1972). O lado direito do problema é obtido pelo produto $Ax = b$. As Tabelas 14, 15, 16 e 17 exibem respectivamente os resultados dos métodos MPIFCC, MPILU, MPCFCC e MPCLU para $NL = 10^{-4}$.

Tabela 14 – MPIFCC aplicado à matriz de Shaw.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
250	46	787	$2,200 \times 10^{-2}$	$9,980 \times 10^{-6}$	$3,817 \times 10^{-1}$
500	82	1650	$3,290 \times 10^{-2}$	$9,074 \times 10^{-6}$	$2,093 \times 10^{+0}$
1000	88	2100	$3,270 \times 10^{-2}$	$7,396 \times 10^{-6}$	$1,796 \times 10^{+1}$
2000	95	1443	$3,340 \times 10^{-3}$	$6,734 \times 10^{-6}$	$5,901 \times 10^{+1}$
3000	100	1159	$3,290 \times 10^{-2}$	$6,044 \times 10^{-6}$	$1,175 \times 10^{+2}$

Tabela 15 – MPILU aplicado à matriz de Shaw.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
250	56	1139	$2,370 \times 10^{-2}$	$6,453 \times 10^{-6}$	$6,572 \times 10^{-1}$
500	92	1896	$3,240 \times 10^{-2}$	$7,777 \times 10^{-6}$	$2,879 \times 10^{+0}$
1000	100	1747	$3,370 \times 10^{-2}$	$7,805 \times 10^{-6}$	$9,058 \times 10^{+0}$
2000	100	1728	$3,380 \times 10^{-2}$	$1,280 \times 10^{-5}$	$2,988 \times 10^{+1}$
3000	100	1651	$3,410 \times 10^{-2}$	$3,922 \times 10^{-5}$	$6,707 \times 10^{+1}$

Tabela 16 – MPCFCC aplicado à matriz de Shaw.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
250	13	332	$3,250 \times 10^{-2}$	$1,474 \times 10^{-5}$	$3,200 \times 10^{-1}$
500	13	332	$3,360 \times 10^{-2}$	$9,887 \times 10^{-6}$	$7,339 \times 10^{-1}$
1000	14	354	$3,340 \times 10^{-2}$	$8,180 \times 10^{-6}$	$3,478 \times 10^{+0}$
2000	14	343	$3,350 \times 10^{-2}$	$6,793 \times 10^{-6}$	$2,598 \times 10^{+1}$
3000	15	372	$3,310 \times 10^{-2}$	$6,273 \times 10^{-7}$	$8,920 \times 10^{+1}$

Tabela 17 – MPCLU aplicado à matriz de Shaw.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
250	12	344	$3,170 \times 10^{-2}$	$1,283 \times 10^{-5}$	$4,074 \times 10^{-1}$
500	12	337	$3,450 \times 10^{-2}$	$1,144 \times 10^{-5}$	$8,760 \times 10^{-1}$
1000	15	396	$3,230 \times 10^{-2}$	$5,710 \times 10^{-6}$	$4,019 \times 10^{+0}$
2000	14	356	$3,310 \times 10^{-2}$	$7,223 \times 10^{-6}$	$1,500 \times 10^{+1}$
3000	15	382	$3,360 \times 10^{-2}$	$6,546 \times 10^{-6}$	$3,363 \times 10^{+2}$

Para este exemplo, à medida que aumentamos a dimensão da matriz, os métodos MPIFCC e MPILU atingiram o número máximo de iterações de ponto interior preestabelecido para o código. Além disso, demandaram maior número de iterações de MGCP do que os métodos MPCFCC e MPCLU, mas mostraram ordem de precisão dos resultados equivalentes a estes métodos. Em termos de número de iterações de ponto

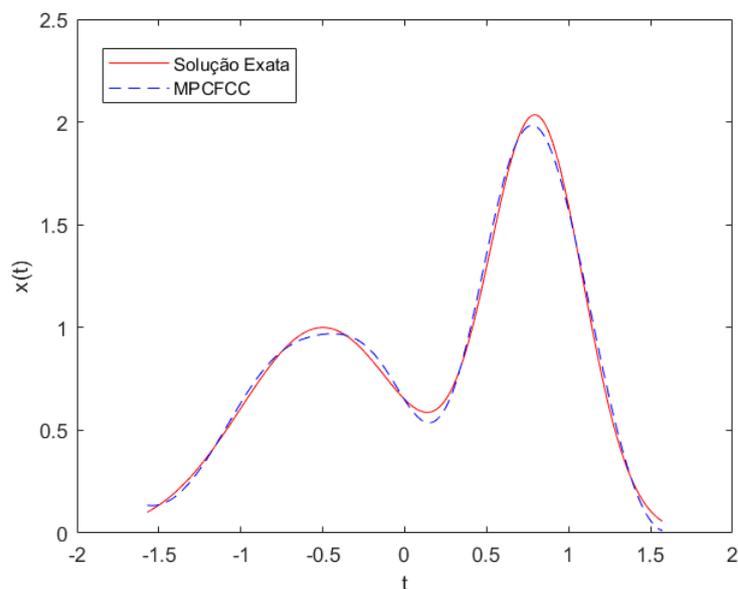


Figura 3 – Comparação entre a solução exata e a solução obtida pelo MPCFCC para o problema Shaw com $NL = 10^{-4}$

interior e de MGCP, os métodos MPCFCC e MPCLU tiveram resultados bem próximos e podemos dizer o mesmo com relação à precisão dos resultados alcançados pelos mesmos. Na maior parte dos casos, o tempo de processamento do MPCFCC obteve resultados melhores do que o método MPCLU. Na Figura 3 temos um comparativo entre a solução exata deste problema e a solução numérica obtida pelo método MPCFCC, para ordem $n = 1000$. Como pode ser observado por esta figura, o comportamento da solução numérica está próximo da solução exata. Este exemplo é particularmente interessante, pois neste caso os MPI's combinados ao preconditionador FCC forneceram resultados tão satisfatórios quanto os combinados ao preconditionador separador.

5.5 Phillips

Prosseguindo, considere a equação integral de Fredholm de primeiro tipo

$$\int_{-6}^6 K(s, t)x(t)dt = g(s), \quad -6 \leq s, t \leq 6,$$

em que a solução $x(t)$ é dada por

$$x(t) = \begin{cases} 1 + \cos\left(\frac{\pi t}{3}\right), & |t| < 3, \\ 0, & |t| \geq 3, \end{cases}$$

o núcleo $K(s, t) = k(s - t)$ e $g(s) = (6 - |s|) \left(1 + \frac{1}{2} \cos\left(\frac{\pi s}{3}\right)\right) + \frac{9}{2\pi} \operatorname{sen}\left(\frac{\pi |s|}{3}\right)$. A

discretização desta equação integral pelo método de Galerkin produz uma matriz simétrica e indefinida, veja (PHILLIPS, 1962). Os resultados obtidos por nossos métodos para este problema discretizado e com $NL = 10^{-3}$ estão dispostos nas Tabelas 18, 19, 20 e 21.

Os resultados de tempo de processamento t_{cpu} do MPILU foram, para todas as dimensões apresentadas, melhores do que o MPIFCC. No entanto, para quase todos os casos, o número de iterações de MGCP foram maiores do que o MPIFCC. Os resultados de tempo de processamento t_{cpu} do MPCFCC se tornaram maiores do que os do MPCLU a partir da ordem $n = 2048$, novamente devido à necessidade de atualizar o preconditionador.

Tabela 18 – MPIFCC aplicado à matriz de Phillips.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
252	23	473	$1,390 \times 10^{-2}$	$4,338 \times 10^{-4}$	$2,741 \times 10^{-1}$
504	22	425	$1,390 \times 10^{-2}$	$4,935 \times 10^{-4}$	$8,595 \times 10^{-1}$
1024	29	640	$1,160 \times 10^{-2}$	$3,154 \times 10^{-4}$	$3,852 \times 10^{+0}$
2048	39	2242	$6,700 \times 10^{-3}$	$1,386 \times 10^{-4}$	$7,554 \times 10^{+1}$
4096	37	846	$9,400 \times 10^{-3}$	$2,927 \times 10^{-4}$	$7,069 \times 10^{+2}$

Tabela 19 – MPILU aplicado à matriz de Phillips.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
252	20	560	$1,640 \times 10^{-2}$	$6,053 \times 10^{-4}$	$2,854 \times 10^{-1}$
504	26	808	$5,400 \times 10^{-3}$	$4,206 \times 10^{-4}$	$6,139 \times 10^{-1}$
1024	36	1235	$1,470 \times 10^{-2}$	$3,749 \times 10^{-4}$	$2,691 \times 10^{+0}$
2048	34	1293	$1,060 \times 10^{-2}$	$3,922 \times 10^{-4}$	$1,076 \times 10^{+1}$
4096	42	1165	$1,360 \times 10^{-2}$	$3,718 \times 10^{-4}$	$4,775 \times 10^{+1}$

Tabela 20 – MPCFCC aplicado à matriz de Phillips.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
252	9	285	$1,180 \times 10^{-2}$	$6,052 \times 10^{-4}$	$2,707 \times 10^{-1}$
504	11	377	$1,440 \times 10^{-2}$	$5,890 \times 10^{-4}$	$6,451 \times 10^{-1}$
1024	12	417	$1,480 \times 10^{-2}$	$3,644 \times 10^{-4}$	$3,200 \times 10^{+0}$
2048	13	502	$1,210 \times 10^{-2}$	$4,740 \times 10^{-4}$	$2,557 \times 10^{+1}$
4096	12	458	$1,120 \times 10^{-2}$	$4,235 \times 10^{-4}$	$2,876 \times 10^{+2}$

Tabela 21 – MPCLU aplicado à matriz de Phillips.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
252	10	385	$1,000 \times 10^{-2}$	$3,143 \times 10^{-4}$	$3,817 \times 10^{-1}$
504	11	429	$1,250 \times 10^{-2}$	$3,563 \times 10^{-4}$	$9,546 \times 10^{-1}$
1024	12	458	$9,000 \times 10^{-3}$	$3,182 \times 10^{-4}$	$4,547 \times 10^{+0}$
2048	13	512	$1,170 \times 10^{-2}$	$3,919 \times 10^{-4}$	$2,302 \times 10^{+1}$
4096	12	466	$1,160 \times 10^{-2}$	$5,213 \times 10^{-4}$	$1,038 \times 10^{+2}$

Os resultados de tempo de processamento t_{cpu} do MPILU foram, para todas as dimensões apresentadas, melhores do que o MPIFCC. No entanto, para quase todos os casos, o número de iterações de MGCP foram maiores do que o MPIFCC. Os resultados de tempo de processamento t_{cpu} do MPCFCC se tornaram maiores do que os do MPCLU a partir da ordem $n = 2048$, novamente devido à necessidade de atualizar o preconditionador.

Em geral, o método MPCFCC obteve resultados mais satisfatórios do que o método MPIFCC. Por outro lado o método MPILU atingiu resultados de tempo de processamento t_{cpu} mais satisfatórios do que o método MPCLU.

5.6 Problemas de teste: Foxgood, Gravity, Odftomo, Sphericaltomo e PRtomo

Os exemplos a seguir tem dimensões especificadas na Tabela 22, são mal condicionados e também provêm de (HANSEN, 2007) e (GAZZOLA; HANSEN; NAGY, 2019).

Tabela 22 – Dimensões dos Problemas de Testes

Problema	m	n
Foxgood	3000	3000
Gravity	2500	2500
Odftomo	2304	1024
Sphericaltomo	8100	1024
PRtomo	8100	1024

Nós resolvemos estes problemas para três níveis de ruído $NL=10^{-4}, 10^{-3}, 10^{-2}$. Por simplicidade, iremos nos referir ao problema Sphericaltomo apenas como Spherical. As Tabelas 23, 25 e 27 mostram os resultados obtidos pelos métodos MPIFCC e MPILU. Nas Tabelas 24, 26 e 28 temos os resultados obtidos pelos métodos MPCFCC e MPCLU.

Tabela 23 – Resultados obtidos pelos MPI's para $NL=10^{-4}$

NL= 10^{-4}	MPIFCC				MPILU			
	It	Itgcp	Erd	t_{cpu}	It	Itgcp	Erd	t_{cpu}
Foxgood	100	642	$4,20 \times 10^{-3}$	$1,07 \times 10^{+2}$	100	641	$7,00 \times 10^{-3}$	$2,99 \times 10^{+1}$
Gravity	31	3205	$8,90 \times 10^{-2}$	$1,53 \times 10^{+2}$	58	3160	$8,90 \times 10^{-2}$	$8,15 \times 10^{+1}$
Odftomo	31	2642	$7,83 \times 10^{-4}$	$2,39 \times 10^{+1}$	100	101803	$7,59 \times 10^{-4}$	$2,19 \times 10^{+2}$
Spherical	65	5458	$1,92 \times 10^{-4}$	$4,52 \times 10^{+1}$	83	57407	$1,92 \times 10^{-4}$	$2,80 \times 10^{+1}$
PRtomo	50	4564	$6,61 \times 10^{-4}$	$5,18 \times 10^{+1}$	100	102703	$4,43 \times 10^{-4}$	$3,78 \times 10^{+2}$

Tabela 24 – Resultados obtidos pelos MPC's para $NL=10^{-4}$

NL= 10^{-4}	MPCFCC				MPCLU			
	Problema	It	Itgcp	Erd	t_{cpu}	It	Itgcp	Erd
Foxgood	12	157	$3,70 \times 10^{-3}$	$9,55 \times 10^{+1}$	14	199	$3,70 \times 10^{-3}$	$9,58 \times 10^{+0}$
Gravity	11	474	$8,90 \times 10^{-3}$	$6,41 \times 10^{+1}$	12	1005	$7,60 \times 10^{-3}$	$2,59 \times 10^{+0}$
Odfotomo	15	2414	$8,17 \times 10^{-4}$	$2,47 \times 10^{+1}$	4	6869	$7,78 \times 10^{-4}$	$1,82 \times 10^{+1}$
Spherical	14	1832	$1,92 \times 10^{-4}$	$2,07 \times 10^{+1}$	12	5461	$2,04 \times 10^{-4}$	$5,60 \times 10^{+0}$
PRtomo	14	2514	$3,98 \times 10^{-4}$	$2,23 \times 10^{+1}$	1	2356	$4,93 \times 10^{-4}$	$1,12 \times 10^{+1}$

Tabela 25 – Resultados obtidos pelos MPI's para $NL=10^{-3}$

NL= 10^{-3}	MPIFCC				MPILU			
	Problema	It	Itgcp	Erd	t_{cpu}	It	Itgcp	Erd
Foxgood	100	4498	$1,45 \times 10^{-2}$	$2,31 \times 10^{+2}$	100	563	$1,69 \times 10^{-2}$	$2,69 \times 10^{+1}$
Gravity	29	3160	$2,68 \times 10^{-2}$	$6,36 \times 10^{+1}$	42	1586	$2,68 \times 10^{-2}$	$4,10 \times 10^{+1}$
Odfotomo	32	3590	$7,50 \times 10^{-3}$	$2,55 \times 10^{+1}$	100	101760	$7,40 \times 10^{-3}$	$2,17 \times 10^{+3}$
Spherical	56	4927	$2,00 \times 10^{-3}$	$4,34 \times 10^{+1}$	100	83518	$2,00 \times 10^{-3}$	$4,10 \times 10^{+1}$
PRtomo	59	5959	$4,00 \times 10^{-3}$	$6,37 \times 10^{+1}$	100	102708	$4,30 \times 10^{-3}$	$3,79 \times 10^{+2}$

Tabela 26 – Resultados obtidos pelos MPC's para $NL=10^{-3}$

NL= 10^{-3}	MPCFCC				MPCLU			
	Problema	It	Itgcp	Erd	t_{cpu}	It	Itgcp	Erd
Foxgood	13	163	$6,60 \times 10^{-3}$	$9,56 \times 10^{+1}$	13	165	$2,90 \times 10^{-3}$	$9,30 \times 10^{+0}$
Gravity	10	374	$2,63 \times 10^{-2}$	$6,09 \times 10^{+1}$	12	814	$2,81 \times 10^{-2}$	$2,17 \times 10^{+1}$
Odfotomo	17	2380	$7,30 \times 10^{-3}$	$2,25 \times 10^{+1}$	4	7012	$7,80 \times 10^{-3}$	$1,85 \times 10^{+1}$
Spherical	17	2880	$1,90 \times 10^{-3}$	$2,44 \times 10^{+1}$	12	7642	$1,90 \times 10^{-3}$	$7,58 \times 10^{+0}$
PRtomo	15	3083	$5,00 \times 10^{-3}$	$2,40 \times 10^{+1}$	1	2358	$3,90 \times 10^{-3}$	$1,12 \times 10^{+1}$

Tabela 27 – Resultados obtidos pelos MPI's para $NL=10^{-2}$

NL= 10^{-2}	MPIFCC				MPILU			
	Problema	It	Itgcp	Erd	t_{cpu}	It	Itgcp	Erd
Foxgood	100	9098	$3,39 \times 10^{-2}$	$3,81 \times 10^{+2}$	50	291	$3,32 \times 10^{-2}$	$1,47 \times 10^{+1}$
Gravity	27	625	$1,11 \times 10^{-1}$	$6,37 \times 10^{+1}$	31	1008	$1,15 \times 10^{-1}$	$2,67 \times 10^{+1}$
Odfotomo	35	4082	$7,46 \times 10^{-2}$	$3,04 \times 10^{+1}$	100	102035	$7,44 \times 10^{-2}$	$2,23 \times 10^{+2}$
Spherical	65	5609	$1,89 \times 10^{-2}$	$5,13 \times 10^{+1}$	100	90773	$1,89 \times 10^{-2}$	$4,48 \times 10^{+1}$
PRtomo	60	5292	$5,01 \times 10^{-2}$	$5,68 \times 10^{+1}$	100	102281	$4,92 \times 10^{-2}$	$3,77 \times 10^{+2}$

Tabela 28 – Resultados obtidos pelos MPC's para $NL=10^{-2}$

NL= 10^{-2}	MPCFCC				MPCLU			
	It	Itgcp	Erd	t_{cpu}	It	Itgcp	Erd	t_{cpu}
Foxgood	12	116	$4,12 \times 10^{-2}$	$9,51 \times 10^{+1}$	12	138	$1,79 \times 10^{-2}$	$7,12 \times 10^{+0}$
Gravity	9	332	$8,61 \times 10^{-2}$	$6,02 \times 10^{+1}$	10	551	$4,64 \times 10^{-2}$	$1,49 \times 10^{+1}$
Odfotomo	18	2274	$7,82 \times 10^{-2}$	$1,98 \times 10^{+1}$	4	6813	$7,33 \times 10^{-2}$	$1,81 \times 10^{+1}$
Spherical	21	3639	$1,93 \times 10^{-2}$	$3,18 \times 10^{+1}$	10	6335	$1,94 \times 10^{-2}$	$6,65 \times 10^{+0}$
PRtomo	18	5435	$8,13 \times 10^{-2}$	$3,29 \times 10^{+1}$	1	2358	$3,52 \times 10^{-2}$	$1,12 \times 10^{+1}$

Apesar do problema Foxgood não satisfazer a condição discreta de Picard para valores singulares pequenos, veja (BAKER, 1977), nossos métodos se mostraram eficientes em obter uma solução numérica aproximada para este problema. Em especial os métodos do tipo preditor-corretor, necessitaram de menos iterações tanto de pontos interiores quanto de gradientes conjugados para alcançar convergência

Para o problema Gravity é perceptível, observando as Tabelas 23, 24, 25, 26, 27 e 28, que os métodos alcançaram resultados com precisão bastante similares. Os métodos do tipo preditor-corretor mais uma vez se mostraram eficientes em diminuir o número de iterações de MPI e de MGCP para se obter convergência. Também, na maior parte dos casos, os métodos preconditionados com a fatoração LU obtiveram os melhores tempos de processamento t_{cpu} .

Odfotomo, Spherical e PRtomo são problemas de reconstrução tomográfica bidimensionais. PRtomo, por exemplo, é usado para gerar problemas de teste que modelam a tomografia de atenuação de raios-X, geralmente chamada de tomografia computadorizada. Para estes três exemplos, para cada nível de ruído $NL = 10^{-4}, 10^{-3}, 10^{-2}$, a ordem de precisão obtida pelos métodos foi respectivamente a mesma. O MPILU alcançou o número máximo de iterações preestabelecido para os métodos mais vezes que o método MPIFCC. Também, na maioria dos casos, o número iterações de MGCP do MPILU superou em muito o número de iterações de MGCP do MPIFCC. A superioridade do MPIFCC sobre o MPILU para estes três exemplos também se refletiu na maior parte dos tempos de processamento t_{cpu} .

Ainda para os exemplos Odfotomo, Spherical e PRtomo, em todos os casos o número de iterações de pontos interiores do método MPCFCC foi maior do que o método MPCLU. Por outro lado, com exceção do problema PRtomo, o MPCFCC precisou de menos iterações de MGCP do que o MPCLU. No que diz respeito ao tempo de processamento computacional, para os três níveis de ruídos apresentados, o MPCLU se mostrou mais eficiente em todos os casos.

5.7 MPCLU vs SVD

Na Seção 5.1, para o exemplo Baart de ordem $n = 18000$, não foi possível calcular a decomposição SVD da matriz de Baart devido a limitações de memória da máquina. Por outro lado, o método MPCLU foi capaz de resolvê-lo. Nesta seção, a fim de demonstrar a superioridade de nosso método para resolução de problemas de grande porte quando comparado a métodos baseados em decomposição SVD, faremos mais alguns comparativos. Para tanto, utilizamos a rotina `tikhonov` em (HANSEN, 2007) que resolve o problema de regularização de Tikhonov por meio da decomposição SVD. O parâmetro de regularização $\alpha > 0$ é obtido por meio da rotina `l_curve` também de (HANSEN, 2007). Tal rotina também depende de dados da decomposição SVD da matriz do sistema linear. Nós consideramos as dimensões das matrizes de teste da Seção 5.1 e os níveis de ruído como dados pela Tabela 29. Relembramos que, no caso do problema de desfocagem Blur, o vetor b já corresponde à uma versão empilhada em colunas da imagem borrada.

Tabela 29 – Dimensões dos Problemas de Testes

Problema	m	n	NL
Hilbert	10000	10000	10^{-2}
Baart	5000	5000	10^{-3}
Blur	5184	5184	–
Shaw	15000	15000	10^{-4}
Phillips	18000	15000	10^{-3}
Foxgood	15000	15000	10^{-3}
Gravity	15000	15000	10^{-2}
Odfotomo	18000	62500	10^{-4}
Spherical	34560	18496	10^{-2}
PRtomo	32580	16384	10^{-3}

Na Tabela 30 temos um comparativo entre os resultados obtidos pela rotina `tikhonov` e os resultados obtidos pelo método MPCLU. Nesta tabela, para os problemas Phillips, Odfotomo, Spherical e PRtomo utilizamos o símbolo “–” para indicar que para as especificações de máquina enunciadas no início deste capítulo, não foi possível calcular a decomposição SVD. Por outro lado, mesmo mediante a alta dimensão destes problemas, o método MPCLU obteve êxito em solucioná-los, apesar de uma certa perda de precisão na maioria dos casos quando comparados aos exemplos da Seção 5.1. Para o problema teste de desfocagem Blur, o método MPCLU alcançou uma precisão muito superior à obtida pela rotina `tikhonov`. Esta maior eficiência se repetiu ao compararmos os tempos de processamento computacional destes dois métodos.

Tabela 30 – MPCLU vs SVD

Problema	SVD		MPCLU			
	Erd	t_{cpu}	It	Itgcp	Erd	t_{cpu}
Hilbert	$4,320 \times 10^{-2}$	$2,420 \times 10^{+2}$	13	377	$4,490 \times 10^{-2}$	$1,726 \times 10^{+2}$
Baart	$1,122 \times 10^{-1}$	$4,236 \times 10^{+1}$	11	185	$7,770 \times 10^{-2}$	$2,387 \times 10^{+1}$
Blur	$2,250 \times 10^{-2}$	$5,430 \times 10^{+1}$	6	414	$5,312 \times 10^{-9}$	$4,697 \times 10^{+0}$
Shaw	$2,810 \times 10^{-2}$	$9,039 \times 10^{+2}$	14	417	$3,570 \times 10^{-2}$	$4,596 \times 10^{+2}$
Phillips	–	–	14	887	$4,500 \times 10^{-3}$	$8,086 \times 10^{+2}$
Foxgood	$3,000 \times 10^{-3}$	$8,859 \times 10^{+2}$	14	172	$4,000 \times 10^{-3}$	$2,131 \times 10^{+2}$
Gravity	$2,170 \times 10^{-2}$	$8,826 \times 10^{+2}$	2	52	$2,670 \times 10^{-2}$	$7,202 \times 10^{+1}$
Odfotomo	–	–	13	275046	$5,500 \times 10^{-3}$	$9,230 \times 10^{+3}$
Spherical	–	–	14	3963	$7,459 \times 10^{-1}$	$7,367 \times 10^{+1}$
PRtomo	–	–	11	36416	$2,060 \times 10^{-2}$	$7,294 \times 10^{+2}$

Para nosso problema inicial, a matriz de Hilbert e para os problemas Shaw, Foxgood e Gravity, o MPCLU obteve a mesma ordem de precisão das soluções regularizadas obtidas pelo método `tikhonov`, além de ter obtido melhor tempo de processamento em todos os casos.

Para matriz do problema de Baart, não só o tempo de processamento computacional obtido pelo MPCLU foi melhor do que o obtido pela rotina `tikhonov`, como também obtivemos uma sutil melhora na precisão da solução regularizada.



Figura 4 – Problema de teste PRtomo: (a) Imagem de teste original, (b) Imagem recuperada.

Para finalizar, resolvemos o problema PRtomo para $m = 65160$, $n = 65536$ e $NL = 10^{-3}$. Para tanto, utilizamos a imagem de teste Shepp–Logan phantom, veja (SHEPP; LOGAN, 1974). Após 14 iterações de pontos interiores, 291246 iterações de gradientes conjugados e $2,071 \times 10^{+4}$ segundos, o método MPCLU alcançou uma solução regularizada com erro relativo dado por $Erd = 1,165 \times 10^{-1}$. Nas Figuras (4a) e (4b) temos um comparativo entre a solução verdadeira e a solução recuperada pelo MPCLU após a perturbação de nível $NL = 10^{-3}$ no lado direito do sistema .

6 Conclusões e Perspectivas Futuras

6.1 Conclusões

Neste trabalho apresentamos o problema de regularização de Tikhonov, tendo como motivação sua aplicação a sistemas lineares mal-postos provenientes de problemas inversos lineares.

Reformulamos o problema de regularização de Tikhonov como um problema de programação quadrática por meio de uma formulação Primal-Dual e então propusemos dois métodos, um método de pontos interiores Primal-Dual e um método de pontos interiores do tipo Preditor-Corretor, ambos com barreira logarítmica. O cálculo das direções de busca destes métodos nos levaram a sistemas de equações normais onde o parâmetro $\tau > 0$ que representa a penalização para valores grandes de $\|x\|_2$ na formulação Primal-Dual passou a exercer o papel de parâmetro de regularização.

Para encontrar as direções de busca os sistemas de equações normais foram resolvidos pelo Método dos Gradientes Conjugados Precondicionado. Nós dividimos os dois métodos em quatro variantes conforme o preconditionador escolhido para resolução do sistema de equações normais.

Para os cinco primeiros exemplos apresentados a ordem de precisão das soluções encontradas com os quatro métodos foi muito similar e coerente a resultados encontrados na literatura. Ainda com relação aos cinco primeiros exemplos, no que diz respeito ao tempo de convergência, na maior parte dos casos os métodos preconditionados com o Precondicionador Separador obtiveram melhores tempos para alcançar uma solução regularizada do que os preconditionados com a Fatoração Controlada de Cholesky.

Com o objetivo de continuar com nossa análise e comparar a eficiência dos métodos com os dois preconditionadores propostos, prosseguimos de uma forma simplificada e aplicamos nossos métodos a mais cinco exemplos e com três níveis de ruídos $NL = 10^{-4}, 10^{-3}, 10^{-2}$. Na maior parte dos casos, o método preditor-corretor preconditionado com o Precondicionador Separador alcançou os melhores resultados em números de iterações de pontos interiores e de tempo de processamento, tendo alcançado diferenças significativas em alguns casos. Por este motivo, na Seção 5.7 escolhemos este método para resolver os problemas de teste com dimensões aumentadas. O método preditor-corretor preconditionado com o Precondicionador Separador se mostrou bastante promissor não só em resolver problemas de grande porte, quando não foi possível calcular decomposição SVD, mas também obteve melhores resultados de processamento do que o método `tikhonov` baseado em decomposição SVD, nos casos em que foi possível utilizá-lo.

De um modo geral podemos dizer que todos os métodos obtiveram ordem de precisão da solução bastante similares. Também, na maior parte dos casos as variantes do método Preditor-Corretor realizaram menos iterações do que as variantes do método Primal-Dual, mas nem sempre obtiveram os melhores resultados de tempo computacional.

6.2 Perspectivas Futuras

Embora os resultados obtidos com a implementação dos métodos no software MATLAB sejam eficientes, pretendemos melhorar os resultados obtidos por meio da implementação dos métodos em linguagem C, do que esperamos proporcionar uma maior eficiência no tempo de processamento dos métodos.

Nos exemplos resolvidos no Capítulo 5, mantivemos o parâmetro $\tau > 0$ fixo com valor $5,0 \times 10^{-3}$, valor escolhido dado sua eficiência em uma variedade de testes. No entanto, os resultados obtidos podem ser melhorados por meio de uma base teórica que permita entender como deve ser escolhido este parâmetro.

Em problemas de Programação Linear, com o objetivo de melhorar o condicionamento dos sistemas de equações normais e reduzir o tempo computacional em cada iteração do MPI, são usados métodos iterativos preconditionados. Com este objetivo, uma Abordagem Híbrida foi proposta por (BOCANEGRA; CAMPOS; OLIVEIRA, 2007). Esta abordagem é dividida em duas fases, na primeira é usado o preconditionador Fatoração Controlada de Cholesky (FCC) proposto por (CAMPOS, 1995), já na segunda fase, trabalha-se com o preconditionador Separador (PS) proposto por (OLIVEIRA; SORENSEN, 2005).

Como resultados promissores foram alcançados pela Abordagem Híbrida proposta por (BOCANEGRA; CAMPOS; OLIVEIRA, 2007), poderíamos aplicar uma abordagem similar aos métodos de pontos interiores apresentados nesta tese. Podemos fazer isso por combinar através de uma abordagem híbrida os preconditionadores apresentados no Capítulo 4, com o objetivo de reduzir o número de iterações de MGCP necessárias para alcançar uma solução regularizada e para reduzir o tempo computacional em cada iteração dos métodos de pontos interiores.

Poderão ser investigadas estratégias mais adequadas de atualização e de preenchimento do preconditionador Fatoração Controlada de Cholesky que sejam mais eficientes para os sistemas de equações normais apresentados no Capítulo 4.

Uma implementação do MPIFCC e do MPCFCC que leve em conta o armazenamento de matrizes esparsas pode apresentar melhores resultados de tempo computacional, além de nos permitir abordar problemas esparsos de grande porte fazendo menos uso de memória.

Referências

- BAART, M. L. The use of auto-correlation for pseudo-rank determination in noisy ill-conditioned linear least-squares problems. *IMA J. Numer. Anal.*, 2, p. 241–247, 1982.
- BAKER, C. H. The numerical treatment of integral equations. *Clarendon Press, Oxford*, p. 665, 1977.
- BENZI, M. Preconditioning techniques for large linear systems: A survey. *Journal of Computational Physics*, vol. 182, n. 2, Elsevier, p. 418–477, 2002.
- BERTSIMAS, D.; TSITSIKLIS, J. *Introduction to Linear Optimization*. [S.l.]: Athena Scientific, 1997. (Athena Scientific series in optimization and neural computation). ISBN 9781886529199.
- BJORCK, A. A bidiagonalization algorithm for solving large and sparse ill-posed systems of linear equations. *BIT* 18, p. 659–670, 1988.
- BOCANEGRA, S.; CAMPOS, F. F.; OLIVEIRA, A. R. L. Using a hybrid preconditioner for solving large-scale linear systems from interior point methods. *Computational Optimization and Applications*, 36, p. 149–164, 2007.
- BOYD, S.; VANDENBERGHE, L. *Convex Optimization*. [S.l.]: Cambridge University Press, 1994. ISBN 9780521833783.
- BRINKHUIS, J.; TIKHOMIROV, V. *Optimization: Insights and Applications*. [S.l.]: Princeton University Press, 2011. (Princeton Series in Applied Mathematics). ISBN 9781400829361.
- CALVETTI, D.; MORIGI, S.; REICHEL, L.; SGALLARI, F. Tikhonov regularization and l-curve for large discrete ill-posed problems. *Journal of Computational and Applied Mathematics*, v 123, p. 423–446, 2000.
- CAMPOS, F. F. *Analysis of conjugate gradient-type methods for solving linear equations*. Tese (Doutorado) — University of Oxford, 1995.
- ELDÉN, L. Algorithms for the regularization of ill-conditioned least squares problems. *BIT*, 17, p. 134–145, 1977.
- FRANKLIN, J. N. Well-posed stochastic extensions of ill-posed linear problems. *J. Math. Anal Appl.*, 31, p. 682–716, 1970.
- _____. Minimum principles for ill-posed problems. *SIAM J. Math. Anal.*, 9, p. 638–650, 1978.
- FROMMER, A.; MAASS, P. Fast CG-based methods for Tikhonov-Phillips regularization. *SIAM J. Sci. Comput.*, 20, p. 1831–1850, 1999.
- GAZZOLA, S.; HANSEN, P. C.; NAGY, J. G. Ir tools: A matlab package of iterative regularization methods and large-scale test problems. *Numerical Algorithms*, 81, p. 773–811, 2019.

- GOLUB, G. H. Numerical methods for solving linear least squares problems. *Numer. Math.*, 7, p. 206–216, 1965.
- GOLUB, G. H.; LOAN, C. F. V. *Matrix Computations*. [S.l.]: Johns Hopkins University Press, 1996. (Johns Hopkins Studies in the Mathematical Sciences). ISBN 9780801854149.
- GOLUB, G. H.; MATT, U. V. Quadratically constrained least squares and quadratic problems. *Numer. Math.*, 59, p. 561–579, 1991.
- _____. Tikhonov regularization for large scale problems. *Scientific Computing, Springer, Berlin*, p. 3–26, 1997.
- GONDZIO, J. Multiple centrality corrections in a primal-dual method for linear programming. *Computational Optimization and Applications*, v. 6, n. 2, p. 137–156, 1996.
- HANKE, M.; HANSEN, P. C. Regularization methods for large-scale problems. *Surv. Math. Ind.* 3, p. 253–315, 1993.
- HANKE, M.; VOGEL, C. R. Two-level preconditioners for regularized inverse problems i: Theory. *Numer. Math.*, 83, p. 385–402, 1999.
- HANSEN, P. C. The discrete picard condition for discrete ill-posed problems. *BIT*, 30, p. 658–672, 1990.
- _____. Rank-deficient and discrete ill-posed problems. *SIAM, Philadelphia*, p. 658–672, 1998.
- _____. The l-curve and its use in the numerical treatment of inverse problems. *in Computational Inverse Problems in Electrocardiology*, ed. P. Johnston, *Advances in Computational Bioengineering*, WIT Press, p. 119–142, 2000.
- _____. Regularization tools version 4.0 for matlab 7.3. *Numer. Algorithms*, 46, p. 189–194, 2007.
- HOANG, N. S.; RAMM, A. G. Solving ill-conditioned linear algebraic systems by the dynamical systems method. *Inverse Problems in Science and Engineering*, v. 16, p. 617–630, 2008.
- HOERL, A. E.; KENNARD, R. W. Ridge regression. applications to nonorthogonal problems. *Technometrics*, 12, p. 69–82, 1970.
- _____. Ridge regression. biased estimation for nonorthogonal problems. *Technometrics*, 12, p. 55–67, 1970.
- IZMAILOV, A.; SOLODOV, M. *Otimização - Condições de Otimalidade, Elementos de Análise Convexa e de Dualidade*, vol. 1. [S.l.]: Impa, 2009. ISBN 9788524402388.
- JONES, M. T.; PLASSMANN, P. E. An improved incomplete Cholesky factorization. *ACM Transactions on Mathematical Software (TOMS), ACM*, v. 21, n.1, p. 5–17, 1995.
- KELLEY, C. T. *Iterative Methods for Optimization*. [S.l.]: Society for Industrial and Applied Mathematics (SIAM), 1999. (Frontiers in Applied Mathematics). ISBN 9780898714333.

- MARQUARDT, D. W. Generalized inverses, ridge regression, biased linear estimation, and nonlinear estimation. *Technometrics*, 12, p. 591–612, 1970.
- MEGIDDO, N. Pathways to the optimal set in linear programming. *Progress in Mathematical Programming, Interior-Point and Related Methods, Springer-Verlag, New York. ORSA Journal on Computing, Vol. 2*, p. 131–158, 1989.
- MEHROTRA, S. On the implementation of a primal-dual interior point method. *SIAM Journal on optimization, SIAM, v.2 n. 4*, p. 575–601, 1992.
- MEYER, C. D. *Matrix Analysis and Applied Linear Algebra*. [S.l.]: SIAM, 2010. ISBN 9780898714548.
- MONTEIRO, R. C.; ADLER, I. Interior path-following primal-dual algorithms, part 1: Linear programming. *Mathematical Programming, Vol. 44*, p. 131–158, 1989.
- OLIVEIRA, A. R. L.; SORENSEN, D. A new class of preconditioners for large-scale linear systems from interior point methods for linear programming. *Linear Algebra and its applications, Elsevier, v. 394*, p. 1–24, 2005.
- PARK, Y. *Parameter selection rules for ill-posed problems*. Tese (Doutorado) — Kent State University, 2019.
- PHILLIPS, D. L. A technique for the numerical solution of certain integral equations of the first kind. *J. Assoc. Comput. Mach.*, 9, p. 84–97, 1962.
- PINA, H. *Métodos Numéricos*. [S.l.]: Escolar Editora, 2010. ISBN 9789725922842.
- REICHEL, L.; SADOK, H.; SHYSHKOV, A. Greedy tikhonov regularization for large linear ill-posed problems. *International Journal of Computer Mathematics, vol. 84*, p. 1151–1166, 2007.
- REZGHI, M.; HOSSEINI, S. M. A new variant of l-curve for tikhonov regularization. *Journal of Computational and Applied Mathematics, Elsevier*, p. 914–924, 2009.
- RILEY, J. D. Solving systems of linear equations with a positive definite, symmetric, but possibly ill-conditioned matrix. *Math. Tables Aids comput.*, 9, p. 96–101, 1955.
- SHAW, C. B. Improvements of the resolution of an instrument by numerical solution of an integral equation. *J. Math. Anal. Appl.* 37, p. 83–112, 1972.
- SHEPP, L.; LOGAN, B. F. The fourier reconstruction of a head section. *IEEE Transactions on Nuclear Science. NS-21 (3)*, p. 21–43, 1974.
- TIKHONOV, A. N. Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl.*, 4, p. 1035–1038, 1963.
- TIKHONOV, A. N.; ARSENIN, V. Y. Solution of ill-posed problems. *J. Wiley*, 1977.
- TREFETHEN, L. N.; BAU, D. *Numerical Linear Algebra*. [S.l.]: Society for Industrial and Applied Mathematics, 1997. ISBN 9780521833783.
- TWOMEY, S. On the numerical solution of Fredholm integral equations of the first kind by inversion of the linear system produced by quadrature. *J. Assoc. Comput. Mach.*, 19, p. 97–101, 1963.

VOGEL, C. R. *Computational Methods for Inverse Problems*. [S.l.]: SIAM, 2002. (Frontiers in Applied Mathematics). ISBN 9780898717570.

WRIGHT, S. J. *Primal-Dual Interior-Point Methods*. [S.l.]: Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 1997. (SIAM e-books). ISBN 9781611971453.

ZHANG, Y. A primal-dual interior point approach for computing l_1 and l_∞ solutions of overdetermined linear systems. *J. Optim. Theory Appl.* 77 (2), p. 323–341, 1993.

ZHANG, Y.; TAPIA, R. A. Superlinear and quadratic convergence of primal-dual interior methods for linear programming revisited. *Journal of Optimization Theory and Applications*, Vol. 73, p. 229–242, 1992.