

Universidade Federal de Santa Catarina
Curso de Pós-Graduação em Matemática e
Computação Científica

Métodos de Projeção para
Regularização com Informação a
Priori

Thiane Pereira Poncetta Coliboro

Orientador: Prof. Dr. Fermín Sinfórano Viloche
Bazán

Florianópolis, Fevereiro de 2011.

Universidade Federal de Santa Catarina
Curso de Pós-Graduação em Matemática e
Computação Científica

Métodos de Projeção para Regularização
com Informação a Priori

Dissertação apresentada ao Curso de Pós-Graduação em Matemática e Computação Científica, do Centro de Ciências Físicas e Matemáticas da Universidade Federal de Santa Catarina, para a obtenção do grau de Mestre em Matemática, com Área de Concentração em Matemática Aplicada.

Thiane Pereira Poncetta Coliboro

Florianópolis, Fevereiro de 2011.

Métodos de Projeção para Regularização com Informação a Priori

por

Thiane Pereira Poncetta Coliboro

Esta dissertação foi julgada para a obtenção do Título de Mestre em Matemática, Área de Concentração em Matemática Aplicada, e aprovada em sua forma final pelo Curso de Pós-Graduação em Matemática e Computação Científica.

Dr. Clóvis Caesar Gonzaga
Coordenador em Exercício da Pós-Graduação
em Matemática e Computação Científica

Comissão Examinadora

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

Prof^a. Dr^a. Elizabeth Wegner Karas (UFPR)

Prof. Dr. Juliano de Bem Francisco (UFSC)

Prof. Dr. Maicon Marques Alves (UFSC)

Florianópolis, 17 de fevereiro de 2011.

*Ao meu namorado
e à minha mãe*

Agradecimentos

À Deus acima de tudo, por zelar por mim sempre.

Aos meus pais, M^a Bernadete e Milton, por me incentivarem em mais esta etapa.

Ao meu namorado Rodrigo que desde o início da graduação está ao meu lado e que, principalmente nestes dois últimos anos, foi a calma em meio às tempestades.

Aos colegas das salas 106 e 107 por toda ajuda, pelas conversas descontraídas e pelo vôlei aos sábados.

Ao professor Fermín, principalmente. Desde a participação na banca do meu Trabalho de Conclusão de Curso até o término desta dissertação, agradeço pelos ensinamentos transmitidos e pelas valiosas conversas que contribuíram para a conclusão desta importante etapa.

Aos professores Elizabeth, Juliano e Maicon por aceitarem compor a banca, dedicarem tempo para a leitura do trabalho e contribuírem para seu aperfeiçoamento.

Ao Programa de Pós-Graduação em Matemática e Computação Científica da UFSC pelo auxílio na participação de eventos e, em especial, à secretária Elisa.

Ao CNPq, pelo auxílio financeiro nestes dois anos.

Resumo

Apresentamos três métodos de projeção para problemas discretos mal postos de grande porte que incorporam informação a priori da solução do problema. Os métodos são baseados em uma transformação do funcional de Tikhonov da forma geral (com uma seminorma como termo regularizante) para a forma padrão [26, 53].

Os dois primeiros métodos combinam o processo de bidiagonalização de Golub-Kahan [15] com a regularização de Tikhonov na forma geral, calculando soluções aproximadas em subespaços de Krylov. O parâmetro de regularização λ é escolhido pelo Método de Ponto Fixo (FP) de Bazán [3]. O terceiro método não depende da determinação do parâmetro λ sendo, portanto, uma alternativa para a Regularização de Tikhonov.

São apresentadas algumas generalidades sobre problemas inversos e problemas discretos mal-postos. Também é feito um estudo sobre projeções oblíquas, conceito essencial na transformação para a forma padrão. A performance dos métodos quando aplicados a problemas testes bem conhecidos e ao tratamento de imagens é ilustrada numericamente.

Palavras-chave: Métodos de projeção, Bidiagonalização de Golub e Kahan, Regularização de Tikhonov na forma geral, Problemas inversos de grande porte, Regularização iterativa.

Abstract

We present three projection methods for large-scale discrete ill-posed problems which use a priori information about the exact solution of the problem. These methods are based on a Tikhonov functional transformation from general form (with a seminorm as regularizing term) to a standard-form problem [26, 53].

The two first methods combine the Golub-Kahan bidiagonalization algorithm[15] with the general-form Tikhonov regularization and calculate approximated solutions in the generated Krylov subspaces. The regularization parameter λ is chosen by the Fixed-point method (FP) by Bazán [3]. The third method is an alternative to Tikhonov regularization that does not require the computation of the parameter λ .

We first describe some generalities on inverse and discrete ill-posed problems. Next, we do a study concerning the oblique projections, an important idea to the standard-form transformation. The performance of the methods when applied to well-known test problems and to image deblurring and reconstruction problems are illustrated numerically.

Keywords: Projection Methods, Golub-Kahan Bidiagonalization, general-form Tikhonov regularization, Large-scale inverse problem, Iterative regularization.

Lista de Figuras

1.1	Modelagem matemática de um processo físico. . . .	7
1.2	Os 32 primeiros valores singulares σ_i de uma matriz A e valores singulares generalizados γ_i do par matricial (A, L)	17
1.3	Vetores singulares \mathbf{u}_i , $i = 1, 3, 5, 7$ e 9 (acima) e vetores singulares generalizados \mathbf{u}_i , $i = 32, 30, 28, 26$ e 24 (abaixo).	18
1.4	Condição discreta de Picard para o problema <i>Gravity</i> com $\mathbf{b}_{\text{exato}}$ (esquerda) e $\mathbf{b} = \mathbf{b}_{\text{exato}} + \mathbf{e}$ (direita).	21
1.5	Iterações λ_k para o problema teste <i>shaw</i> [24].	24
1.6	Solução do problema <i>Deriv2</i> de dimensão 1000 com 0,1% de ruído nos dados, utilizando regularização de Tikhonov na forma padrão, ou seja, $L = I$. Erros relativos: $E_{FP} = 14,56\%$, $E_{LC} = 14,39\%$ e $E_{GCV} = 14,39\%$	26
1.7	Solução do problema <i>Deriv2</i> de dimensão 1000 com 0,1% de ruído nos dados, utilizando regularização de Tikhonov na forma geral para $L = L_1$. Erros relativos: $E_{FP} = 1,64\%$, $E_{LC} = 1,43\%$ e $E_{GCV} = 1,05\%$	27
1.8	Solução do problema <i>Deriv2</i> de dimensão 1000 com 0,1% de ruído nos dados, utilizando regularização de Tikhonov na forma geral para $L = L_2$. Erros relativos: $E_{FP} = 0,86\%$, $E_{LC} = 0,57\%$ e $E_{GCV} = 0,28\%$	27
2.1	Solução aproximadas para $\mathbf{x}_{\text{exato}}$	30

3.1	Decomposição de um vetor \mathbf{s} arbitrário.	44
3.2	Ilustração da decomposição oblíqua $\mathbb{R}^n = \mathcal{N}(L) + \mathcal{N}(L)^{\perp A}$ para a transformação para a forma padrão.	59
4.1	Função $E(k)$	80
4.2	Comportamento da função $\Psi(k)$	81
5.1	<i>Gravity</i> : Solução exata.	87
5.2	<i>Gravity</i> : Avaliação da CDP para (A, L_1) e (A, L_2) . Coeficientes $ \mathbf{u}_i^T \mathbf{b} /\gamma_i$: \circ , Coeficientes de Fourier: \times , Valores singulares: \cdot	88
5.3	<i>Phillips</i> : Solução exata.	92
5.4	<i>Phillips</i> : Avaliação da CDP para (A, L_1) e (A, L_2) . Coeficientes $ \mathbf{u}_i^T \mathbf{b} /\gamma_i$: \circ , Coeficientes de Fourier: \times , Valores singulares: \cdot	93
5.5	<i>Deriv2</i> exemplo 2: Solução exata.	97
5.6	<i>Deriv2</i> exemplo 2: Avaliação da CDP para (A, L_1) e (A, L_2) . Coeficientes $ \mathbf{u}_i^T \mathbf{b} /\gamma_i$: \circ , Coeficientes de Fourier: \times , Valores singulares: \cdot	98
5.7	<i>Deriv2</i> exemplo 3: Solução exata.	102
5.8	<i>Deriv2</i> exemplo 3: Avaliação da CDP para (A, L_1) (esquerda) e (A, L_2) (direita). Coeficientes $ \mathbf{u}_i^T \mathbf{b} /\gamma_i$: \circ , Coeficientes de Fourier: \times , Valores singulares: \cdot	102
5.9	Imagem em alta resolução e duas em baixa resolu- ção.	107
5.10	Imagem <i>Tree</i> exata e em alta resolução.	109
5.11	Sequências de imagens em baixa resolução, com 2304 <i>pixels</i> cada.	110
5.12	Soluções obtidas para o problema <i>Tree</i> com 1% de ruído.	113
5.13	Função $\bar{\Psi}$ para o problema <i>Tree</i> com 1% de ruído.	114
5.14	Imagem exata e imagem em que foi aplicado o efeito blur.	116
5.15	Imagem rice.png da <i>toolbox</i> Image Processing	118
5.16	Imagens estimadas com 1% de ruído nos dados e a matriz L_{kron}	120
5.17	Função $\bar{\Psi}$ para o problema <i>Rice</i> com 1% de ruído.	121

B.1 Curva-L genérica	152
--------------------------------	-----

Lista de Tabelas

2.1	Algoritmo GKB-FP	40
3.1	Algoritmo <i>mkron1</i>	66
3.2	Algoritmo <i>mkron2</i>	66
4.1	Desenvolvimento dos produtos matriz-vetor envolvendo \bar{A} no caso não singular.	70
4.2	Desenvolvimento dos produtos matriz-vetor envolvendo \bar{A} no caso singular.	72
4.3	Algoritmo GGKB-FP.	73
4.4	Algoritmo da fatoração QR via Gram-Schmidt.	75
4.5	Algoritmo PROJ-FP.	77
4.6	Algoritmo G-LSQR.	82
5.1	Problemas teste do <i>toolbox</i> ReguralizationTools.	84
5.2	Parâmetros iniciais para os problemas testes.	86
5.3	Símbolos para descrição dos resultados numéricos.	86
5.4	Resultados obtidos para o problema <i>Gravity</i> após 50 execuções com 0,1% de ruído nos dados e usando a matriz L_1	89
5.5	Resultados obtidos para o problema <i>Gravity</i> após 50 execuções com 0,1% de ruído nos dados e usando a matriz L_2	89
5.6	Resultados obtidos para o problema <i>Gravity</i> após 50 execuções com 1% de ruído nos dados e usando a matriz L_1	90

5.7	Resultados obtidos para o problema <i>Gravity</i> após 50 execuções com 1% de ruído nos dados e usando a matriz L_2	90
5.8	Resultados obtidos para o problema <i>Gravity</i> após 50 execuções com 2,5% de ruído nos dados e usando a matriz L_1	91
5.9	Resultados obtidos para o problema <i>Gravity</i> após 50 execuções com 2,5% de ruído nos dados e usando a matriz L_2	91
5.10	Resultados obtidos para o problema <i>Phillips</i> após 50 execuções com 0,1% de ruído nos dados e usando a matriz L_1	94
5.11	Resultados obtidos para o problema <i>Phillips</i> após 50 execuções com 0,1% de ruído nos dados e usando a matriz L_2	94
5.12	Resultados obtidos para o problema <i>Phillips</i> após 50 execuções com 1% de ruído nos dados e usando a matriz L_1	95
5.13	Resultados obtidos para o problema <i>Phillips</i> após 50 execuções com 1% de ruído nos dados e usando a matriz L_2	95
5.14	Resultados obtidos para o problema <i>Phillips</i> após 50 execuções com 2,5% de ruído nos dados e usando a matriz L_1	96
5.15	Resultados obtidos para o problema <i>Phillips</i> após 50 execuções com 2,5% de ruído nos dados e usando a matriz L_2	96
5.16	Resultados obtidos para o problema <i>Deriv2 exemplo 2</i> após 50 execuções com 0,1% de ruído nos dados e L_1	99
5.17	Resultados obtidos para o problema <i>Deriv2 exemplo 2</i> após 50 execuções com 0,1% de ruído nos dados e L_2	99
5.18	Resultados obtidos para o problema <i>Deriv2 exemplo 2</i> após 50 execuções com 1% de ruído nos dados e L_1	100

5.19	Resultados obtidos para o problema <i>Deriv2 exemplo 2</i> após 50 execuções com 1% de ruído nos dados e L_2	100
5.20	Resultados obtidos para o problema <i>Deriv2 exemplo 2</i> após 50 execuções com 2,5% de ruído nos dados e L_1	101
5.21	Resultados obtidos para o problema <i>Deriv2 exemplo 2</i> após 50 execuções com 2,5% de ruído nos dados e L_2	101
5.22	Resultados obtidos para o problema <i>Deriv2 exemplo 3</i> após 50 execuções com 0,1% de ruído nos dados e L_1	103
5.23	Resultados obtidos para o problema <i>Deriv2 exemplo 3</i> após 50 execuções com 0,1% de ruído nos dados e L_2	103
5.24	Resultados obtidos para o problema <i>Deriv2 exemplo 3</i> após 50 execuções com 1% de ruído nos dados e L_1	104
5.25	Resultados obtidos para o problema <i>Deriv2 exemplo 3</i> após 50 execuções com 1% de ruído nos dados e L_2	104
5.26	Resultados obtidos para o problema <i>Deriv2 exemplo 3</i> após 50 execuções com 2,5% de ruído nos dados e L_1	105
5.27	Resultados obtidos para o problema <i>Deriv2 exemplo 3</i> após 50 execuções com 2,5% de ruído nos dados e L_2	105
5.28	Resultados obtidos para o problema <i>Tree</i> após 50 execuções com 0,1% de ruído nos dados e usando a matriz L_{kron}	111
5.29	Resultados obtidos para o problema <i>Tree</i> após 50 execuções com 1% de ruído nos dados e usando a matriz L_{kron}	112
5.30	Resultados obtidos para o problema <i>Tree</i> após 50 execuções com 2,5% de ruído nos dados e usando a matriz L_{kron}	114

5.31	Algoritmo <i>mkron</i>	119
5.32	Resultados obtidos para o problema <i>Rice</i> após 50 execuções com 0,1% de ruído nos dados e usando a matriz L_{kron}	121
5.33	Resultados obtidos para o problema <i>Rice</i> após 50 execuções com 1% de ruído nos dados e usando a matriz L_{kron}	122
5.34	Resultados obtidos para o problema <i>Rice</i> após 50 execuções com 2,5% de ruído nos dados e usando a matriz L_{kron}	122

Notações

\mathbb{R}^n	Conjunto dos vetores com n coordenadas reais
$\mathbb{R}^{m \times n}$	Conjunto das matrizes com entrada reais de m linhas e n colunas
\mathbf{x}	Um vetor de \mathbb{R}^n
$\ \mathbf{x}\ _P$	Norma vetorial $\ \mathbf{x}\ _P = (\mathbf{x}^T P \mathbf{x})^{\frac{1}{2}}$ para $\mathbf{x} \in \mathbb{R}^n$ e $P \in \mathbb{R}^{n \times n}$ simétrica positiva definida
I_n	Matriz identidade de ordem n
A^\dagger	Matriz pseudo-inversa de uma matriz A
$\text{diag}(d_1, \dots, d_p)$	Matriz diagonal de ordem $m \times n$ onde $d_{ij} = 0$ para todo $i \neq j$, $d_i = d_{ii}$ e $p = \min\{m, n\}$
$\mathcal{R}(A)$	Conjunto imagem do operador A
$\mathcal{N}(A)$	Núcleo do operador A
$\lambda(A)$	Conjunto de autovalores de uma matriz A

Sumário

Introdução	1
1 Problemas Mal Postos e Regularização	7
1.1 Problemas Inversos e sua Relação com Problemas Mal Postos	8
1.2 Método de Regularização	11
1.3 Problemas Discretos Mal Postos	12
1.4 Regularização de Tikhonov e GSVD	14
1.5 Condição Discreta de Picard	19
1.6 Determinação do Parâmetro de Regularização	22
1.6.1 Método de Ponto Fixo	23
2 Métodos Básicos para Problemas de Grande Porte	29
2.1 Métodos de Projeção	30
2.2 Bidiagonalização de Golub-Kahan	33
2.3 Método LSQR	35
2.4 Método GKB-FP	38
2.4.1 Algoritmo	39
2.4.2 Critério da Parada	40
3 Transformação para a Forma Padrão	43
3.1 Projeções Oblíquas	44
3.2 Transformação do Problema de Regularização de Tikhonov na Forma Geral para a Forma Padrão	56
3.3 Transformação e a GSVD	61
3.4 Matrizes Estruturadas	62

4	Métodos de Projeção para Regularização com Informação a Priori	67
4.1	GKB-FP para Tikhonov na Forma Geral	68
4.1.1	Caso Não Singular	69
4.1.2	Caso Singular	70
4.1.3	Critério de Parada	72
4.1.4	Algoritmo	72
4.2	PROJ-FP	73
4.2.1	Algoritmo	76
4.2.2	Critério de Parada	78
4.3	G-LSQR: Regularização Iterativa	78
4.3.1	Critério de Parada	79
4.3.2	Algoritmo	82
5	Resultados Numéricos	83
5.1	Problemas Teste	84
5.1.1	<i>Gravity</i>	86
5.1.2	<i>Phillips</i>	92
5.1.3	<i>Deriv2</i>	97
5.2	Tratamento de Imagens	106
5.2.1	Imagem <i>Tree</i>	108
5.2.2	Imagem <i>Rice</i>	117
	Considerações Finais	123
	Apêndice	124
A	SVD e GSVD: história e propriedades	125
A.1	A Decomposição em Valores Singulares	125
A.2	Generalizações para a SVD	131
A.2.1	Decomposição CS	132
A.2.2	A Primeira Versão de Van Loan	134
A.2.3	Melhoramentos por Paige e Saunders	140

B Métodos para Escolha do Parâmetro de Regularização	147
B.1 Método da Validação Cruzada Generalizada (GVC) e Weighted-GCV	147
B.2 Princípio da Discrepância	149
B.3 Curva L	150
Referências Bibliográficas	153

Introdução

Dada uma matriz $A \in \mathbb{R}^{m \times n}$ e um vetor $\mathbf{b}_{\text{exato}} \in \mathbb{R}^m$, considere o problema de encontrar $\mathbf{x}_{\text{exato}} \in \mathbb{R}^n$ que satisfaça

$$A\mathbf{x}_{\text{exato}} = \mathbf{b}_{\text{exato}}. \quad (1)$$

Muitas vezes, quando o problema (1) não tem solução, resolvemos

$$\mathbf{x}_{\text{exato}} = \underset{\mathbf{x} \in \mathbb{R}^n}{\operatorname{argmin}} \|\mathbf{b}_{\text{exato}} - A\mathbf{x}\|_2^2, \quad (2)$$

isto é, temos um problema de mínimos quadrados. Considere ainda que a matriz A é mal condicionada e que o vetor $\mathbf{b}_{\text{exato}}$ é desconhecido, ou seja, o lado direito de (1) é da forma $\mathbf{b} = \mathbf{b}_{\text{exato}} + \mathbf{e}$, em que \mathbf{e} é um vetor de ruídos. Neste caso, a solução usual de mínimos quadrados não é útil em aplicações práticas por estar contaminada por erros. Nosso objetivo é construir aproximações estáveis para $\mathbf{x}_{\text{exato}}$ em (2) a partir do problema

$$\underset{\mathbf{x} \in \mathbb{R}^n}{\operatorname{argmin}} \|\mathbf{b} - A\mathbf{x}\|_2^2. \quad (3)$$

Esse tipo de problema é conhecido como “problema discreto mal posto” e obter uma boa aproximação para a solução exata é de interesse em diversas áreas em ciência e tecnologia [14, 26, 29, 30, 35, 43, 48].

Buscando obter soluções tão próximas quanto possível da solução exata, usualmente utilizamos algum método de regularização. Dentre os métodos de regularização, o mais popular é o método de Tikhonov [53] que substitui o problema usual de

mínimos quadrados por

$$\mathbf{x}_\lambda = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^n} \left\{ \|\mathbf{b} - A\mathbf{x}\|_2^2 + \lambda^2 \|L\mathbf{x}\|_2^2 \right\}, \quad (4)$$

em que $\lambda > 0$ é o parâmetro de regularização e $L \in \mathbb{R}^{p \times n}$ uma matriz que incorpora informação a priori da solução exata no problema.

O caso mais frequente considera L como o operador identidade e método é conhecido como de Regularização de Tikhonov na forma padrão. A escolha $L \neq I$, denominada regularização de Tikhonov na forma geral, é justamente o caso de nosso interesse neste trabalho.

A escolha de L depende das características do problema contínuo que deu origem ao problema de mínimos quadrados. Assim, cada problema deve ser analisado de modo a se determinar a matriz que produzirá, juntamente com o parâmetro de regularização, uma boa aproximação para $\mathbf{x}_{\text{exato}}$. Neste trabalho utilizamos dois tipos de matriz L que acrescentam ao problema informações acerca da suavidade da solução. São aproximações discretas da primeira e segunda derivadas de um operador.

Além da matriz L , o parâmetro de regularização λ também deve ser escolhido na regularização de Tikhonov. Existem diversos métodos para encontrar tal parâmetro, dos quais podemos citar a Curva L de Hansen [23, 27], a Validação Cruzada Generalizada (GCV) de Golub, Heath e Wahba [17], o princípio da discrepância de Morozov [38, 39] e o algoritmo de Ponto Fixo (FP) de Bazán [3]. Para a escolha de λ utilizamos o método FP que calcula o parâmetro de regularização para o problema na forma geral como sendo um ponto fixo da função

$$\varphi_\mu(\lambda) = \sqrt{\mu} \frac{\|\mathbf{b} - A\mathbf{x}_\lambda\|_2}{\|L\mathbf{x}_\lambda\|_2}, \quad \mu > 0. \quad (5)$$

Recentemente, foi introduzido por Bazán e Borges [5] o método GKB-FP, um método iterativo que busca implementar FP para problemas de grande porte, utilizando para isso o processo de Bidiagonalização de Golub-Kahan (GKB) [15] e o Método

Least Square QR (LSQR) [41] para resolver o problema de mínimos quadrados. No entanto, o método GKB-FP foi desenvolvido apenas para o caso em que $L = I$.

O primeiro método proposto neste trabalho surge como a complementação do método GKB-FP, abordando também o caso da regularização de Tikhonov na forma geral. Intitulado GGKB-FP, o método utiliza a estratégia de transformar a regularização na forma geral dada na equação (4) para a forma padrão

$$\mathbf{y}_\lambda = \operatorname{argmin}_{\mathbf{y} \in \mathbb{R}^p} \left\{ \|\bar{\mathbf{b}} - \bar{A}\mathbf{y}\|_2^2 + \lambda^2 \|\mathbf{y}\|_2^2 \right\}, \quad (6)$$

em que $\bar{A} = AL_A^\dagger \in \mathbb{R}^{m \times p}$ e L_A^\dagger é a matriz pseudo inversa oblíqua de L . A solução \mathbf{y}_λ é calculada pelo método GKB-FP e usado para obter \mathbf{x}_λ .

Mesmo sendo uma excelente ferramenta teórica, a dificuldade da transformação é que ela pode ter um grande custo computacional quando A e L são matrizes de grande porte. Sendo assim, buscamos metodologias para fazer tal transformação de forma eficiente, incorporando estas estratégias em GGKB-FP.

Preocupados com o custo computacional, propomos uma segunda abordagem para a regularização na forma geral. O método PROJ-FP constrói soluções aproximadas para o problema (4) como

$$\mathbf{x}_\lambda^{(k)} = V_k \mathbf{y}_\lambda^{(k)},$$

com

$$\mathbf{y}_\lambda^{(k)} = \operatorname{argmin}_{\mathbf{y} \in \mathbb{R}^k} \left\{ \|\beta - B_k \mathbf{y}\|_2^2 + \lambda^2 \|L_k \mathbf{y}\|_2^2 \right\} \quad (7)$$

em que $\beta \in \mathbb{R}^{k+1}$, e $V_k \in \mathbb{R}^{n \times k}$ e $B_k \in \mathbb{R}^{(k+1) \times k}$ são obtidos no k -ésimo passo do processo de bidiagonalização e $L_k \in \mathbb{R}^{p \times k}$ com k muito menor do que p . O parâmetro de regularização é calculado por FP.

Devido ao fato da escolha do parâmetro λ ser um problema à parte na regularização de Tikhonov, exploramos uma nova estratégia de regularização que não necessita de tal parâmetro, denominado método G-LSQR ou Regularização Iterativa. Utilizando a transformação para a forma padrão, a proposta do

método consiste em calcular a solução do problema discreto mal posto como

$$\mathbf{x}^{(k)} = L_A^\dagger \mathbf{y}^{(k)} + x_{\mathcal{N}} \quad \text{com} \quad \mathbf{y}^{(k)} = \underset{\mathbf{y} \in \mathcal{V}_k \subset \mathbb{R}^p}{\operatorname{argmin}} \|\bar{\mathbf{b}} - \bar{A}\mathbf{y}\| \quad (8)$$

onde $x_{\mathcal{N}} \in \mathcal{N}(L)$ e \bar{A} , $\bar{\mathbf{b}}$ são como na equação (6).

Assim, as informações fornecidas pela matriz L estão incorporados ao problema original por meio da matriz \bar{A} . Por se tratar de um método iterativo, adotamos um critério de parada que avalia o comportamento de $\|\mathbf{x}^{(k)}\|_2$ e $\|\mathbf{b} - A\mathbf{x}^{(k)}\|_2$ à medida que k aumenta.

O trabalho está dividido em cinco capítulos. O Capítulo 1 inicia-se com generalidades sobre problemas inversos e problemas mal postos, mostrando como eles podem surgir naturalmente, por exemplo, da resolução de equações integrais de primeira espécie. Em seguida, introduzimos a Teoria de Regularização, bem como o método de Tikhonov, um dos mais populares da área.

Após uma introdução sobre problemas mal postos, apresentamos os problemas discretos mal posto e a Regularização de Tikhonov. Relembramos a decomposição em valores singulares generalizados de um par matricial (A, L) e como ela pode ser utilizada como um método de regularização. Finalizamos discutindo alguns métodos para a escolha do parâmetro de regularização como Curva L [23], GCV [17], Princípio de Discrepância [38] e Método do Ponto Fixo [3, 4]. O Método de Ponto-Fixo é apresentado com detalhes por ser utilizado nos métodos propostos neste trabalho.

No segundo capítulo apresentamos métodos básicos para problemas de grande porte, começando pelos Métodos de Projeção, que calculam aproximações $\mathbf{x}^{(k)}$ para a solução exata do problema, onde $\mathbf{x}^{(k)}$ está em um subespaço de dimensão k de \mathbb{R}^n , com k muito menor que n . Seguindo, introduzimos o Processo de Bidiagonalização de Golub-Kahan e o método LSQR. Mostramos como LSQR pode ser utilizado muito eficazmente quando aplicado à regularização de Tikhonov na forma padrão. Também discutimos o método GKB-FP, apresentando seu critério de parada com detalhes.

No Capítulo 3 fazemos um estudo detalhado sobre projeções oblíquas e pseudo-inversas oblíquas, essenciais para a transformação da forma geral para a forma padrão. Tal definição é precedida por uma série de conceitos pouco discutidos na literatura como a noção de complemento oblíquo de um subespaço \mathcal{X} de \mathbb{R}^n com relação a uma matriz A , simbolizado por $\mathcal{X}^{\perp A}$, e a decomposição do \mathbb{R}^n que pode ser feita utilizando $\mathcal{X}^{\perp A}$. Um ponto importante do capítulo é a demonstração de que um resultado da literatura [30, Teorema 4, Cap. 8], enunciado como válido no caso geral, é verdadeiro apenas sob hipóteses particulares. Finalizamos mostrando algumas matrizes com estruturas especiais, como Toeplitz e de Banda, e fazendo algumas considerações sobre o Produto de Kronecker [37].

No penúltimo capítulo surge a contribuição desta pesquisa, onde introduzimos três novos métodos para tratar problemas discretos mal postos de grande porte. Apresentamos os métodos de projeção GGKB-FP, PROJ-FP e G-LSQR discutidos anteriormente de forma detalhada, exibindo os principais passos dos algoritmos de forma a minimizar o custo computacional.

No Capítulo 5, são exibidos os resultados numéricos dos testes realizados com problemas bem conhecidos da toolbox *Regularization Tools* [24] trabalhando com sistemas de 1024 variáveis que surgem da discretização de equações integrais. Para fins de comparação, exibimos também os resultados obtidos com outros métodos que podem ser utilizados em problemas na forma geral como Curva L e GCV. Além disso abordamos o problema de super resolução e de reconstrução de imagens. O tratamento de imagem é uma das muitas aplicações interessantes de problemas inversos [43, 57, 29].

No final do trabalho apresentamos as considerações finais e dois apêndices. O primeiro contém uma retrospectiva histórica sobre a Decomposição em Valores Singulares (SVD) e, com mais detalhes, sobre a Decomposição em Valores Singulares Generalizada (GSVD). A GSVD é das grandes ferramentas teóricas para a análise de problemas discretos mal postos. Apresentamos também algumas propriedades da pseudo-inversa usual, também conhecida

como pseudo-inversa de Moore-Penrose. O apêndice B traz os métodos para escolha do parâmetro de regularização GCV e W-GCV [9], o Princípio da Discrepância de Morozov e a Curva L.

Capítulo 1

Problemas Mal Postos e Regularização

Muitos sistemas ou processos físicos das ciências naturais e/ou engenharia expressos através de modelos são a fonte de inúmeros problemas em diversos ramos das ciências aplicadas. O termo *sistema* refere-se a uma entidade (física, econômica, etc.) capaz de produzir algum efeito g (saída/output) como consequência de um estímulo ou causa f (entrada/input).

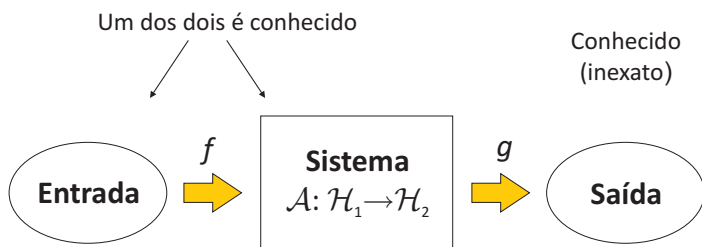


Figura 1.1: Modelagem matemática de um processo físico.

A conexão entre o processo físico e sua modelagem matemática é usualmente descrita por meio de equações do tipo

$$\mathcal{A}f = g \tag{1.1}$$

onde $\mathcal{A} : \mathcal{H}_1 \rightarrow \mathcal{H}_2$ é um operador (diferencial, integral, etc),

possivelmente não linear, $f \in \mathcal{H}_1$ é a função “entrada”, $g \in \mathcal{H}_2$ é a função “dados”, com \mathcal{H}_1 e \mathcal{H}_2 sendo espaços normados, chamados espaço de soluções e espaço de dados, respectivamente. Este modelo é representado na Figura 1.1.

O problema de calcular a resposta (ou saída) de um sistema, dado o modelo e a função de entrada, é relativamente simples de ser resolvido e é conhecido como *problema direto*. Já o *problema inverso*, consiste em determinar a causa (ou entrada) desconhecida a partir de efeitos desejados ou observados. Em outras palavras, sendo conhecida a função g e o operador \mathcal{A} , o problema inverso consiste em determinar f , o que corresponde à inversão do operador, caso exista.

Em situações práticas, a função g é obtida experimentalmente, portanto sujeita a erros de medição. Este fato é frequente quando a função g provém da leitura de sinais usando aparelhos, portanto, carentes de precisão absoluta. Nestas condições somos forçados a trabalhar com uma versão aproximada g_δ com certo nível de ruído, isto é,

$$\|g_\delta - g\| \leq \delta, \quad (1.2)$$

onde δ depende da precisão do instrumento e a escolha da norma é feita levando-se em conta características intrínsecas do processo físico envolvido. Detalhes a respeito de problemas inversos e aplicações podem ser encontrados em [14, 26, 35, 48, 56].

1.1 Problemas Inversos e sua Relação com Problemas Mal Postos

No processo de resolução de um problema inverso, deve ser feita uma análise prévia com relação a questões de existência, unicidade e estabilidade de soluções. Tal análise é de vital importância pois, se a função g_δ não pertence ao espaço imagem do operador, então (1.1) não tem solução. Outra possibilidade é que \mathcal{A} seja não injetor (o qual significa que \mathcal{A}^{-1} não existe), caso em que podem existir muitas soluções.

Por outro lado, precisamos saber se a solução depende continuamente dos dados. A estabilidade é necessária se desejamos assegurar que pequenas variações nos dados produzam pequenas mudanças na solução. As considerações que acabamos de apresentar são o ponto central da discussão e análise de problemas mal postos.

A noção de problema bem-posto surgiu no início do século XX quando o matemático francês J. S. Hadamard estudava problemas de valor na fronteira, com a ideia de que nenhum fenômeno natural poderia ser refletido por algum modelo que conduza a um problema mal posto; posteriormente comprovou-se a falsidade de tais ideias.

Dizemos que o problema (1.1) é bem-posto no sentido de Hadamard quando são satisfeitas as seguintes condições:

1. *Existência*: Para cada $g \in \mathcal{H}_2$ existe $f \in \mathcal{H}_1$ tal que $\mathcal{A}f = g$.
2. *Unicidade*: Para cada $g \in \mathcal{H}_2$ a equação (1.1) tem uma única solução.
3. *Estabilidade*: A solução f depende continuamente dos dados de entrada, isto é, o operador \mathcal{A}^{-1} é contínuo.

Caso alguma das condições não seja satisfeita o problema é dito mal posto. É importante destacar que a dependência contínua da solução com respeito aos dados é uma condição necessária mas não suficiente. Por isso, no caso de um problema linear bem-posto, a propagação do erro relativo na solução com respeito ao erro relativo nos dados é controlada pelo número de condição do operador $\kappa(\mathcal{A}) = \|\mathcal{A}\| \|\mathcal{A}^{-1}\|$.

Especificamente, sabe-se que se resolvemos o problema perturbado $\mathcal{A}(f + \delta f) = g + \delta g$ temos que

$$\frac{\|\delta f\|}{\|f\|} \leq \kappa(\mathcal{A}) \frac{\|\delta g\|}{\|g\|}$$

onde δg é a variação nos dados e δf é a variação na solução. Se $\kappa(\mathcal{A})$ está próximo de 1, diz-se que o problema é bem-condicionado

e a solução é estável com respeito a pequenas variações nos dados. Pelo contrário, se $\kappa(\mathcal{A})$ é grande, diz-se que o problema é mal condicionado e o problema é instável. Claramente, o conceito de ser bem ou mal condicionado é menos preciso que o conceito de ser bem ou mal posto.

A área de problemas inversos é um dos ramos da matemática aplicada com o maior crescimento na última década, devido ao grande número de aplicações extraídas de problemas de interesse contemporâneo na ciência e tecnologia. Exemplos são encontrados em problemas de restauração de imagens, eletrocardiograma, tomografia por impedância, exploração geofísica de depósitos de petróleo, identificação biométrica, entre outros [35, 48].

O exemplo típico de problema inverso mal posto corresponde a equações integrais de primeira espécie

$$(\mathcal{A}f)(x) := \int_a^b K(x, y)f(y)dy = g(x), \quad c \leq x \leq d, \quad (1.3)$$

onde a existência e a unicidade de soluções geralmente não é garantida. Por exemplo, se \mathcal{A} é operador integral definido pela função $K(x, y)$ contínua em $[c, d] \times [a, b]$ e $f \in L^2[a, b]$, então $\mathcal{A}f \in \mathcal{C}[c, d]$. Consequentemente, a equação (1.1) não tem solução em $L^2(a, b)$ se g não for contínua.

Por outro lado, a equação:

$$\int_{-\pi}^{\pi} x \operatorname{sen}(y) f(y) dy = g(x), \quad -\pi \leq x \leq \pi$$

tem infinitas soluções pois o núcleo do operador $\mathcal{N}(\mathcal{A})$ tem dimensão infinita. De fato, basta notar que a família infinita de funções definidas por $f_n(y) = \operatorname{sen}(ny)$, $n = 2, 3, \dots$, pertencem ao núcleo do operador. E, ainda que as situações acima não aconteçam, salvo o caso da função K ser da forma $K(x, y) = K_1(x)K_2(y)$ (também conhecido como o caso do núcleo degenerado), o problema sempre é instável, já que da análise funcional sabe-se que o operador inverso \mathcal{A}^{-1} é descontínuo.

Diversos problemas inversos são modelados pela equação integral (1.3) como por exemplo o Problema de Diferenciação e o Problema de Equação do Calor para tempo anterior, disponíveis em [35].

1.2 Método de Regularização

A teoria que trata dos métodos de resolução de problemas mal postos é conhecida como *teoria de regularização*, enquanto os métodos são chamados *métodos de regularização*. Intuitivamente, a teoria de regularização é a teoria das “aproximações contínuas” para a inversa descontínua. Portanto, a análise e solução de um problema mal posto é feita via solução de um problema associado que é bem-posto.

A ideia padrão de muitos métodos de regularização é construir uma família de operadores $R_\lambda : \mathcal{H}_2 \rightarrow \mathcal{H}_1$ limitados, que dependem de λ , que aproximem pontualmente o operador \mathcal{A}^{-1} (caso ele exista) ou operador pseudo-inversa \mathcal{A}^\dagger . Isto é,

$$\lim_{\lambda \rightarrow 0} R_\lambda \mathcal{A}(f) = f, \quad \forall f \in \mathcal{H}_1 \quad (1.4)$$

isto é, o operador $R_\lambda \mathcal{A}$ converge pontualmente para o operador Identidade [35].

O valor λ é chamado de *parâmetro de regularização* e os operadores de *operadores regularizantes*. Maiores detalhes sobre teoria de regularização podem ser encontradas em [14, 19].

De modo geral, estudam-se métodos de regularização considerando:

- a) Problemas mal postos como consequência de não existência ou ambiguidade na solução;
- b) Problemas mal postos como consequência de instabilidade.

Dentre essas possibilidades, a segunda é a mais delicada.

Dentre os métodos de regularização, um dos mais populares é devido a A. Tikhonov [53], introduzido em 1963. De forma

breve, o método consiste em tomar como soluções aproximadas para (1.1), as funções f_λ que minimizam o funcional

$$J_\lambda(f, g_\delta) = \|g_\delta - \mathcal{A}f\|^2 + \lambda^2 \Omega(f), \quad \lambda > 0 \quad (1.5)$$

para um valor adequado de λ , onde $\Omega(f)$ é um funcional contínuo chamado funcional estabilizador, satisfazendo condições apropriadas. Detalhes sobre este método podem ser obtidos em [19].

A forma mais comum do funcional estabilizado é $\Omega(f) = \|\mathcal{L}f\|_2^2$ onde \mathcal{L} é algum operador definido num subconjunto denso de \mathcal{H}_1 . Neste caso o minimizador do funcional J_λ é dado por

$$f_\lambda = (\mathcal{A}^* \mathcal{A} + \lambda^2 \mathcal{L}^* \mathcal{L})^{-1} \mathcal{A}^* g_\delta, \quad (1.6)$$

onde \mathcal{A}^* representa o operador adjunto de \mathcal{A} e o operador regularizante é

$$R_\lambda = (\mathcal{A}^* \mathcal{A} + \lambda^2 \mathcal{L}^* \mathcal{L})^{-1} \mathcal{A}^*. \quad (1.7)$$

Na formulação original, Tikhonov propôs

$$\Omega(f) = \int_a^b [v(x)f(x)^2 + w(x)f'(x)^2] dx \quad (1.8)$$

com v, w funções de peso positivas e \mathcal{H}_1 um espaço de Sobolev [1].

1.3 Problemas Discretos Mal Postos

Quando consideramos estudar uma equação integral de primeira espécie

$$\int_a^b K(x, y)f(y)dy = g(x), \quad c \leq x \leq d \quad (1.9)$$

e temos que resolvê-la numericamente, faz-se necessário o uso de algum método de discretização.

Geralmente a discretização resulta num problema de minimização

$$\operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 \quad (1.10)$$

em que a matriz $A \in \mathbb{R}^{m \times n}$ com $m \geq n$ e $\mathbf{b} \in \mathbb{R}^m$. A grande dificuldade encontrada agora é que, pelo fato da matriz se originar da discretização de um problema mal posto, o problema de minimização torna-se muito sensível a pequenas variações no vetor \mathbf{b} e sua solução não é de utilidade prática. Uma das ferramentas mais poderosas para análise e resolução do problema (1.10) é a decomposição em valores singulares (SVD) da matriz A .

Teorema 1.1 (SVD). *Para cada $A \in \mathbb{R}^{m \times n}$, existem matrizes ortogonais $U = [\mathbf{u}_1, \dots, \mathbf{u}_m] \in \mathbb{R}^{m \times m}$ e $V = [\mathbf{v}_1, \dots, \mathbf{v}_n] \in \mathbb{R}^{n \times n}$ tais que*

$$U^T A V = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_p) \in \mathbb{R}^{m \times n}, \quad p = \min\{m, n\} \quad (1.11)$$

com valores singulares σ_i , para cada $i \in \{1, 2, \dots, p\}$, ordenados de modo não-crescente $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$.

A demonstração do teorema pode ser encontrada em [18].

Neste caso, os valores singulares da matriz A decaem para zero sem que haja um salto notório neste decaimento e o número de condição da matriz $\kappa(A) = \frac{\sigma_1}{\sigma_n}$ é muito grande. Somando isto ao fato do vetor \mathbf{b} ser da forma

$$\mathbf{b} = \mathbf{b}_{\text{exato}} + \mathbf{e}, \quad (1.12)$$

com $\mathbf{b}_{\text{exato}}$ sendo o vetor sem perturbações desejado e desconhecido, e \mathbf{e} um vetor de incertezas, a conclusão é que a solução de mínimos quadrados, $\mathbf{x}_{LS} = A^\dagger \mathbf{b}$, não tem nenhuma relação com a solução do problema e não tem utilidade prática por estar completamente dominada pelos erros. De fato, usando a SVD segue imediatamente que

$$\mathbf{x}_{LS} = \sum_{i=1}^r \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i,$$

onde $r = \text{posto}(A)$. Sendo \mathbf{b} como em (1.12) temos

$$\mathbf{x}_{LS} = \sum_{i=1}^r \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i = \sum_{i=1}^r \frac{\mathbf{u}_i^T \mathbf{b}_{\text{exato}}}{\sigma_i} \mathbf{v}_i + \frac{\mathbf{u}_i^T \mathbf{e}}{\sigma_i} \mathbf{v}_i$$

e podemos notar que para valores singulares pequenos os coeficientes $\frac{\mathbf{u}_i^T \mathbf{e}}{\sigma_i}$ são grandes, fazendo com que a segunda parcela da relação acima seja dominante, tornando assim, a solução obtida inútil. Hansen [30] definiu esse tipo de problema como sendo *Problema Discreto Mal Posto*. Se o somatório for truncado em $s < r$, com s próximo de r , podemos amenizar o efeito do erro e no cálculo da solução \mathbf{x} , porém, se s for pequeno, deixamos de capturar informações importantes do problema. Disso segue que a escolha de s deve estabelecer um balanço apropriado entre a quantidade de informação do problema que é capturada e a quantidade de erro que é incluída na solução.

A escolha do índice s satisfazendo esse requerimento é conhecido como o método da SVD Truncada (Truncated SVD) [26]. A TSVD é um método de regularização cuja solução é escrita da forma

$$\mathbf{x}_s = \sum_{i=1}^s \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i.$$

em que s , chamado *índice de truncamento*, é o parâmetro de regularização a ser determinado.

1.4 Regularização de Tikhonov e GSVD

Para o problema discretizado a regularização de Tikhonov substitui o problema (1.10) por

$$\mathbf{x}_\lambda = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^n} \left\{ \|\mathbf{b} - A\mathbf{x}\|_2^2 + \lambda^2 \|\mathcal{L}(\mathbf{x} - \mathbf{x}_0)\|_2^2 \right\} \quad (1.13)$$

em que λ é o parâmetro de regularização e \mathcal{L} é uma versão discreta do operador \mathcal{L} em (1.6). O desafio é escolher um parâmetro tal que \mathbf{x}_λ aproxime satisfatoriamente a solução exata $\mathbf{x}_{\text{exato}}$. A matriz $L \in \mathbb{R}^{p \times n}$ é a matriz identidade ou uma aproximação discreta de algum operador diferencial definido, por exemplo, pela primeira ou segunda derivada. O vetor \mathbf{x}_0 é uma aproximação inicial para a solução caso esteja disponível, caso contrário define-se $\mathbf{x}_0 = \mathbf{0}$.

A matriz L na equação (1.13) determina o tipo de problema de Tikhonov:

$L = I$: Problema de Tikhonov na Forma Padrão.

$L \neq I$: Problema de Tikhonov na Forma Geral.

Mais adiante mostraremos que todo problema de regularização de Tikhonov na forma geral pode ser transformado em outro na forma padrão e que alguns problemas podem apresentar soluções mais úteis usando a forma geral.

Esquemas equivalentes à regularização de Tikhonov foram propostos por J. Riley em 1955 e D. L. Phillips em 1962, mas foi G. H. Golub o primeiro autor a propôr, em 1965, uma maneira apropriada de resolver o problema (1.13). A ideia é tratar (1.13) como um problema de mínimos quadrados

$$\mathbf{x}_\lambda = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^n} \left\| \begin{bmatrix} \mathbf{b} \\ \lambda L \mathbf{x}_0 \end{bmatrix} - \begin{bmatrix} A \\ \lambda L \end{bmatrix} \mathbf{x} \right\|_2^2 \quad (1.14)$$

cujas equações normais regularizadas são

$$(A^T A + \lambda^2 L^T L) \mathbf{x}_\lambda = A^T \mathbf{b} + \lambda^2 L^T L \mathbf{x}_0. \quad (1.15)$$

A solução para a equação (1.15) pode ser escrita, para o caso mais frequente $\mathbf{x}_0 = \mathbf{0}$, como

$$\mathbf{x}_\lambda = (A^T A + \lambda^2 L^T L)^{-1} A^T \mathbf{b} \quad (1.16)$$

sendo a unicidade obtida se $\mathcal{N}(A) \cap \mathcal{N}(L) = \{\mathbf{0}\}$.

Considerando regularização de Tikhonov na forma padrão, \mathbf{x}_λ é dado por

$$\mathbf{x}_\lambda = \sum_{i=1}^r f_i \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i \quad (1.17)$$

onde $r = \operatorname{posto}(A)$ e

$$f_i = \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} \cong \begin{cases} 1, & \sigma_i \gg \lambda \\ \frac{\sigma_i^2}{\lambda^2}, & \sigma_i \ll \lambda \end{cases} \quad (1.18)$$

são os chamados fatores de filtro para a regularização de Tikhonov. Os fatores f_i filtram as componentes do erro da solução. Assim, se em (1.17) λ for muito grande, a solução calculada pode não ter incorporado informações da solução do problema. Em contrapartida, se λ for muito pequeno, pouco ruído pode ter sido filtrado e a solução encontrada é inútil.

O resíduo associado pode ser escrito como

$$r_\lambda = \mathbf{b} - A\mathbf{x}_\lambda = \sum_{i=1}^r (1 - f_i) \mathbf{u}_i^T \mathbf{b} \mathbf{u}_i + \mathbf{b}_\perp \quad (1.19)$$

em que $\mathbf{b}_\perp = \mathbf{b} - \sum_{i=1}^r \mathbf{u}_i^T \mathbf{b} \mathbf{u}_i = \sum_{i=r+1}^m \mathbf{u}_i^T \mathbf{b} \mathbf{u}_i$ é a componente do vetor \mathbf{b} que não pertence ao espaço coluna da matriz A . Lembre-se que como o conjunto $\{\mathbf{u}_i\}$ forma uma base ortonormal para \mathbb{R}^m então $\mathbf{b} = \sum_{i=1}^m \mathbf{u}_i^T \mathbf{b} \mathbf{u}_i$.

Desta forma

$$\|\mathbf{x}_\lambda\|_2^2 = \sum_{i=1}^r \left(f_i \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \right)^2 \quad (1.20)$$

$$\|r_\lambda\|_2^2 = \sum_{i=1}^r ((1 - f_i) \mathbf{u}_i^T \mathbf{b})^2 + \|\mathbf{b}_\perp\|_2^2. \quad (1.21)$$

Neste trabalho estamos particularmente interessados na regularização de Tikhonov na forma geral. Com a utilização de uma matriz $L \neq I$ faz-se necessária a introdução da Decomposição em Valores Singulares Generalizados (GSVD).

Teorema 1.2 (GSVD). *Seja o par matricial (A, L) em que $A \in \mathbb{R}^{m \times n}$, $L \in \mathbb{R}^{p \times n}$ com $m \geq n \geq p$ com $\text{posto}(L) = p$. Então existem matrizes $U \in \mathbb{R}^{m \times n}$ e $V \in \mathbb{R}^{p \times p}$ com colunas ortonormais, e X uma matriz não singular tais que*

$$A = U \begin{bmatrix} \Sigma & 0 \\ 0 & I_{n-p} \end{bmatrix} X^{-1}, \quad L = V [M \quad 0] X^{-1} \quad (1.22)$$

com $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_p)$ e $M = \text{diag}(\mu_1, \dots, \mu_p)$. Os coeficientes σ_i e μ_i satisfazem

$$0 \leq \sigma_1 \leq \dots \leq \sigma_p \leq 1 \quad e \quad 1 \geq \mu_1 \geq \dots \geq \mu_p \geq 0. \quad (1.23)$$

Também, $\Sigma^2 + M^2 = I_p$ e os valores singulares generalizados de (A, L) são definidos como $\gamma_i := \frac{\sigma_i}{\mu_i}$.

Para a demonstração ver [18].

Observação 1.1. As matrizes U e V de GSVD de (A, L) , em geral, são diferentes das matrizes U e V da SVD de A .

Uma característica relevante dos valores singulares generalizados de um par matricial (A, L) é que a ordenação é oposta a dos valores singulares de uma matriz A , ou seja, enquanto os valores singulares estão ordenados de forma decrescente com o índice i , os valores singulares generalizados crescem a medida que i aumenta. Na Figura 1.2 temos a ilustração deste fato. Neste caso a matriz A é fornecida pela rotina *shaw* [24] para dimensão $n = 32$ e L sendo uma aproximação para o operador segunda derivada.

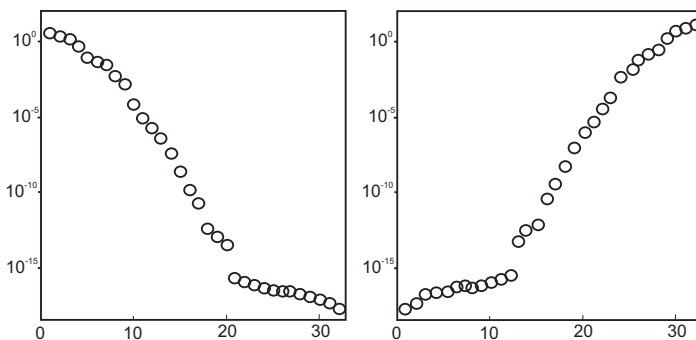


Figura 1.2: Os 32 primeiros valores singulares σ_i de uma matriz A e valores singulares generalizados γ_i do par matricial (A, L) .

Uma outra característica que a SVD e a GSVD têm em comum é o comportamento oscilatório dos vetores singulares à medida que os valores singulares (generalizados) se aproximam de zero. A Figura 1.3 ilustra bem este comportamento, para o problema *shaw* com $n = 32$.

Na prática, para problemas onde a GSVD de A, L pode ser calculada facilmente, a solução do método de regularização de

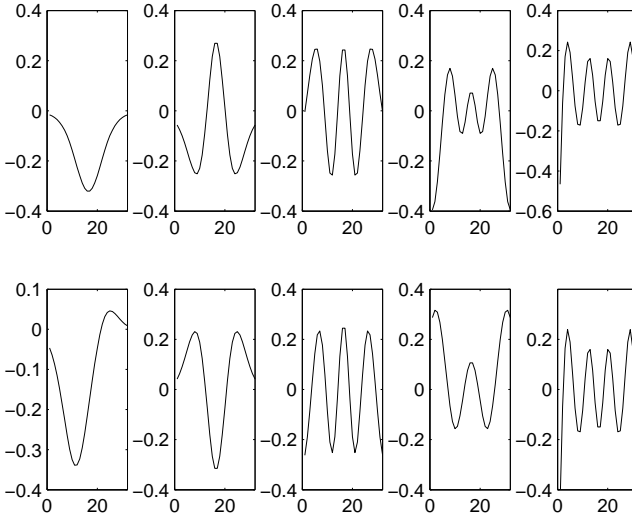


Figura 1.3: Vetores singulares \mathbf{u}_i , $i = 1, 3, 5, 7$ e 9 (acima) e vetores singulares generalizados \mathbf{u}_i , $i = 32, 30, 28, 26$ e 24 (abaixo).

Tikhonov dada na equação (1.16) pode ser escrita como

$$\mathbf{x}_\lambda = \sum_{i=1}^p f_i \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{z}_i + \sum_{i=p+1}^n (\mathbf{u}_i^T \mathbf{b}) \mathbf{z}_i \quad (1.24)$$

onde $f_i = \frac{\gamma_i^2}{\gamma_i^2 + \lambda^2}$ e \mathbf{z}_i são as colunas da matriz X dada pela GSVD.

Neste caso temos o resíduo associado dado por

$$r_\lambda = \mathbf{b} - A\mathbf{x}_\lambda = \sum_{i=1}^p (1 - f_i) \mathbf{u}_i^T \mathbf{b} \mathbf{u}_i + \sum_{i=n+1}^m \mathbf{u}_i^T \mathbf{b} \mathbf{u}_i \quad (1.25)$$

usando o fato que $A\mathbf{z}_i = \sigma_i \mathbf{u}_i$ para $i = 1, \dots, p$ e $A\mathbf{z}_i = \mathbf{u}_i$ para

$i = p + 1, \dots, n$. Assim

$$\|r_\lambda\|_2^2 = \sum_{i=1}^p ((1 - f_i) \mathbf{u}_i^T \mathbf{b})^2 + \sum_{i=n+1}^m (\mathbf{u}_i^T \mathbf{b})^2. \quad (1.26)$$

Note que $\|L\mathbf{x}_\lambda\|_2 = \|\mathbf{x}_\lambda\|_2$ quando $L = I$. Para o caso em que $L \neq I$, a norma $\|\mathbf{x}_\lambda\|_2$ é substituída pela seminorma $\|L\mathbf{x}_\lambda\|_2$. Usando a GSVD temos

$$\|L\mathbf{x}_\lambda\|_2^2 = \sum_{i=1}^p \left(f_i \frac{\mathbf{u}_i^T \mathbf{b}}{\gamma_i} \right)^2 \quad (1.27)$$

A equação (1.27) será usada em alguns métodos de regularização que serão apresentados mais adiante.

Assim, como a SVD pode ser utilizada como um método de regularização por meio da TSVD, o mesmo pode ser feito usando a GSVD. Esse método é conhecido como GSVD Truncada (TGSVD) e mais detalhes podem ser encontrados em [21]. O desafio é determinar o parâmetro s de truncamento.

No pacote *Regularization Tools* [24] existe a rotina *tikhonov.m* que calcula a solução \mathbf{x}_λ dado o parâmetro de regularização e a GSVD (ou SVD) das matrizes A e L . A rotina calcula a solução \mathbf{x}_λ como na equação (1.24). Como dificuldade do uso desta rotina podemos ressaltar o cálculo da GSVD, no caso geral, das matrizes A e L . Quando as dimensões destas matrizes são grandes, o esforço computacional empregado para este cálculo é muito grande.

Tanto SVD como GSVD são ferramentas teóricas importantes na análise de problemas inversos. No Apêndice A apresentamos uma retrospectiva desta decomposição matricial, bem como algumas propriedades.

1.5 Condição Discreta de Picard

Já comentamos que a matriz do problema (1.10) pode ter seus valores singulares indo para zero suavemente, ou seja, sem que haja um salto entre um e outro. Na Figura 1.2 vimos que os

primeiros vinte valores singulares têm um decaimento quase constante, já os últimos doze são de ordem menor do que 10^{-15} , o que significa que atingiram o limite de precisão do computador.

A taxa de decaimento dos valores singulares da matriz A ou do par matricial (A, L) é importante quando se faz a análise de problemas discretos. Para esclarecer este fato, vamos considerar o problema contínuo (1.1) em que \mathcal{A} é o operador integral definido em (1.3). Seja $K(x, y) = \sum_{i=1}^{\infty} \sigma_i^c u_i^c(x) v_i^c(y)$ a expansão em valores singulares do núcleo do operador $\mathcal{A}f$ e seja $g(x) = \sum_i^{\infty} \beta_i \mathbf{u}_i(x)$. É sabido que para a equação integral $\mathcal{A}f = g$ tenha uma solução de mínimos quadrados de quadrado integrável é necessário e suficiente que g satisfaça a seguinte condição

$$\sum_{i=1}^{\infty} \left(\frac{\langle u_i, g \rangle}{\sigma_i^c} \right)^2 < \infty \quad (1.28)$$

em que $\langle \cdot, \cdot \rangle$ denota o produto interno usual no espaço $L^2(a, b)$ [19]. Essa condição é chamada de Condição de Picard e maiores informações podem ser obtidas em [35]. A equação (1.28) implica que em certo ponto do somatório os coeficientes de Fourier $\langle u_i, g \rangle$ decaem mais rapidamente do que os valores singulares σ_i^c .

Obviamente, (1.28) é sempre satisfeita para o caso finito, no entanto, a razão entre o decaimento dos valores singulares de A e o decaimento dos coeficientes de Fourier de \mathbf{b} , $\mathbf{u}_i^T \mathbf{b}$, é importante para que a regularização de Tikhonov seja bem sucedida, pois determina o quão boa a solução regularizada \mathbf{x}_λ pode ser. O resultado central é que se os coeficientes de Fourier $\mathbf{u}_i^T \mathbf{b}$ decaem mais rápido para zero do que os valores singulares generalizados, então as soluções regularizadas \mathbf{x}_λ e \mathbf{x}_s têm aproximadamente as mesmas propriedades da solução exata $\mathbf{x}_{\text{exato}}$.

Existem duas importantes exceções para a exigência de decaimento dos coeficientes de Fourier. A primeira delas é que o menor valor singular generalizado do par matricial (A, L) pode ser numericamente nulo, ou seja, menor do que alguma tolerância ε , refletindo, assim, os erros na matriz A . Nesse caso, é natural considerar valores singulares generalizados menores do que $\varepsilon \|L^\dagger\|_2$ numericamente nulos devido à relação $\gamma_i = \frac{\sigma_i}{1 - \sigma_i} \approx \sigma_i$

para σ_i pequeno.

A segunda exceção deve-se ao caso em que alguns coeficientes $|\mathbf{u}_i^T \mathbf{b}|$ podem ser numericamente nulos com respeito a alguma tolerância δ refletindo os erros em \mathbf{b} . Isso nos leva à definição da condição discreta de Picard.

Condição Discreta de Picard (CDP): Seja $\mathbf{b}_{\text{exato}}$ o vetor de dados não perturbado. Então $\mathbf{b}_{\text{exato}}$ satisfaz a condição de Picard se, para todo valor singular generalizado, numericamente não nulo $\gamma_i > \varepsilon \|L^\dagger\|_2$ os coeficientes de Fourier decaem, em média, para zero mais rápido do que γ_i [22]. De modo intuitivo, isso significa que para a CDP ser válida, a maior parte dos índices i deve satisfazer a desigualdade $|\mathbf{u}_i^T \mathbf{b}| \leq \gamma_i$.

A Figura 1.4 traz os valores singulares generalizados e os coeficientes de Fourier, para inspeção visual da CDP. Para a análise da CDP usamos os seguintes símbolos:

- (o) : coeficientes $\frac{|u_i^T|}{\gamma_i}$;
- (×) : coeficientes de Fourier;
- (·) : valores singulares.

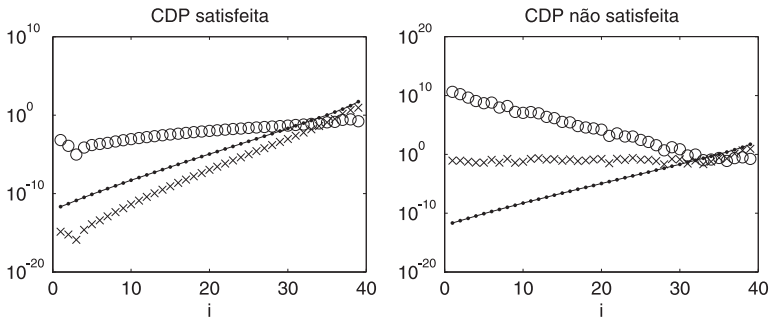


Figura 1.4: Condição discreta de Picard para o problema *Gravity* com $\mathbf{b}_{\text{exato}}$ (esquerda) e $\mathbf{b} = \mathbf{b}_{\text{exato}} + \mathbf{e}$ (direita).

Quando resolvemos problemas do mundo real, em que o vetor \mathbf{b} e, algumas vezes a matriz A , está dominado por erros,

raramente a condição discreta de Picard é satisfeita ainda que a solução exata do problema a satisfaça.

A equação integral de Fredholm de primeira espécie satisfaz a condição de Picard e serve como exemplo para essa situação. Devido à forte conexão entre a expansão em valores singulares do núcleo K e a SVD da matriz A , a condição discreta de Picard é também satisfeita. No entanto, em razão dos erros em alguns ou todos os dados, os coeficientes $\mathbf{u}_i^T \mathbf{b}$ geralmente não satisfazem a condição de Picard.

Com uma escolha apropriada do parâmetro de regularização λ é possível garantir que os coeficientes de Fourier para o problema regularizado, $f_i \mathbf{u}_i^T \mathbf{b}$, satisfaçam a CDP. Por outro lado, se o problema exato não satisfaz a condição discreta de Picard (ou a condição de Picard), então geralmente não é possível calcular uma boa solução por Tikhonov ou qualquer outro método relacionado.

1.6 Determinação do Parâmetro de Regularização

Como já citado anteriormente, a escolha do parâmetro de regularização para o método de Tikhonov, seja na forma padrão ou geral, não é uma tarefa fácil. Existem muitos métodos na literatura para escolher o parâmetro. Dentre eles podemos citar:

- Método da Validação Cruzada Generalizada (GVC) [17] e Weighted-GCV [9];
- Princípio da Discrepância [38, 39];
- Método da Curva L [23, 27];
- Método de Ponto Fixo [3, 4];

Cada um dos métodos acima tem suas particularidades, que devem ser consideradas no momento de decidir qual método utilizar para determinar λ . O apêndice B traz alguns dos métodos acima mais detalhadamente bem como referências bibliográficas.

Trataremos aqui somente o Método de Ponto Fixo [3] pois este método será utilizado nos métodos de projeções propostos neste trabalho.

1.6.1 Método de Ponto Fixo

O algoritmo de Ponto Fixo foi proposto por Bazán [3] baseado num trabalho anterior de Regińska [45]. O algoritmo utiliza informação da norma da solução regularizada e da norma do resíduo correspondente e não requer qualquer tipo de informação do tamanho de ruído nos dados.

Considerando novamente as equações (1.20) e (1.21) podemos definir

$$y(\lambda) = \|\mathbf{x}_\lambda\|_2^2 = \sum_{i=1}^r \left(\frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \right)^2$$

$$x(\lambda) = \|\mathbf{b} - A\mathbf{x}_\lambda\|_2^2 = \sum_{i=1}^r \left(\left(1 - \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} \right) \mathbf{u}_i^T \mathbf{b} \right)^2 + \|\mathbf{b}_\perp\|_2^2.$$

Derivando estas expressões em relação a λ obtemos

$$\frac{dy}{dx} = -\frac{1}{\lambda^2} \quad (1.29)$$

o que mostra que y é uma função de x monótona e decrescente.

Regińska provou que se $\lambda = \lambda^*$ for o parâmetro de regularização que maximiza a curvatura da curva L e, se a tangente à curva L em $(\log x(\lambda^*), \log y(\lambda^*))$ tem inclinação $-1/\mu$, então $\lambda = \lambda^*$ deve ser um minimizador de

$$\Psi_\mu(\lambda) = x(\lambda)y(\lambda)^\mu, \quad \mu > 0. \quad (1.30)$$

Derivando (1.30) em relação a λ temos

$$\Psi'_\mu(\lambda) = y(\lambda)^\mu y'(\lambda) \left[\mu \frac{x(\lambda)}{y(\lambda)} + \frac{x'(\lambda)}{y'(\lambda)} \right]. \quad (1.31)$$

Como $y(\lambda)^\mu y'(\lambda) \neq 0$ e $\frac{x'(\lambda)}{y'(\lambda)} = -\lambda^2$, a condição necessária para que $\Psi_\mu(\lambda)$ tenha um mínimo local em $\lambda = \lambda^* \neq 0$ é que

$\Psi'_\mu(\lambda^*) = 0$, que é equivalente a

$$\lambda^{*2} = \mu \frac{x(\lambda^*)}{y(\lambda^*)} \Leftrightarrow \lambda^* = \sqrt{\mu \frac{x(\lambda^*)}{y(\lambda^*)}}. \quad (1.32)$$

Conseqüentemente, se λ^* é um minimizador de $\Psi_\mu(\lambda)$ então λ^* deve ser um ponto fixo de $\varphi_\mu : \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$ definida por

$$\varphi_\mu(\lambda) = \sqrt{\mu} \frac{\|\mathbf{b} - A\mathbf{x}_\lambda\|_2}{\|\mathbf{x}_\lambda\|_2}. \quad (1.33)$$

O algoritmo de Ponto Fixo começa com um chute inicial λ_0 , $\mu = 1$ e prossegue com a seqüência, para $k = 0, 1, 2, \dots$

$$\lambda_{k+1} = \varphi_\mu(\lambda_k). \quad (1.34)$$

A Figura 1.5 ilustra a busca do ponto fixo de $\varphi_\mu(\lambda_k)$ para um problema teste.

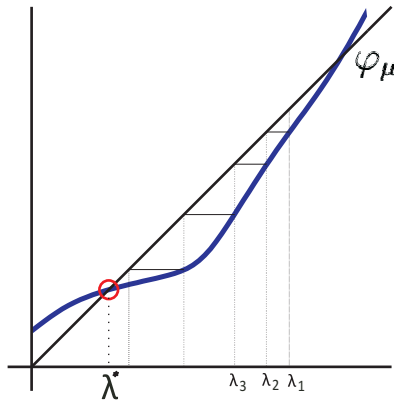


Figura 1.5: Iterações λ_k para o problema teste *shaw* [24].

A convergência da seqüência (1.34) decorre do fato que a função $\varphi_\mu(\lambda)$ é estritamente crescente, o que segue da propriedade (1.29), enquanto a existência de ponto fixo da função $\varphi_\mu(\lambda)$ depende do parâmetro μ (na maioria dos problemas $\mu = 1$ é suficiente para garantir tal existência). Para critérios de parada ou ajustes no μ o leitor é recomendado a ler [3].

Para contornar possíveis dificuldades provenientes da presença de vários pontos fixos, Bazán e Francisco introduziram o conceito de ponto fixo convexo que pode ser encontrado em [4].

O método FP pode também ser utilizado para determinar o parâmetro de regularização para um problema de Tikhonov na forma geral. Neste caso, utilizamos $y(\lambda) = \|L\mathbf{x}_\lambda\|_2^2$ e, portanto, a função φ_μ é dada por

$$\varphi_\mu(\lambda) = \sqrt{\mu} \frac{\|\mathbf{b} - A\mathbf{x}_\lambda\|_2}{\|L\mathbf{x}_\lambda\|_2}. \quad (1.35)$$

Para problemas pequenos, a função φ_μ pode ser construída utilizando a fatoração GSVD do par matricial (A, L) para obter expressões mais simples para $\|\mathbf{b} - A\mathbf{x}_\lambda\|_2$ e $\|L\mathbf{x}_\lambda\|_2$, como descrito na seção 1.4.

Encerramos o capítulo ilustrando o comportamento das soluções \mathbf{x}_λ para $L = I$ e $L \neq I$ construídas pelos métodos Ponto Fixo, Curva L e GCV. Nos três métodos usamos a fatoração SVD (GSVD). Neste caso, utilizamos o problema teste *Deriv2* (exemplo 2), que basicamente é o cálculo da segunda derivada de um operador [24]. Todos os detalhes deste problema são dados na seção 5.1.

Dado um natural n como entrada e o exemplo escolhido, os dados de saída são a matriz $A \in \mathbb{R}^{n \times n}$, o lado direito exato $\mathbf{b} \in \mathbb{R}^n$ e a solução exata $\mathbf{x}_{\text{exato}} \in \mathbb{R}^n$. Para este exemplo trabalhamos com um erro relativo de 0,1% no lado direito \mathbf{b} , sendo este gerado pelo comando *randn* do Matlab, e $n = 1000$.

Para $L = I$, os erros relativos na solução

$$E = \frac{\|\mathbf{x}_{\text{aprox}} - \mathbf{x}_{\text{exato}}\|_2}{\|\mathbf{x}_{\text{exato}}\|_2} \quad (1.36)$$

foram muito grandes, resultado retratado pela Figura 1.6. Buscando solução de melhor qualidade, usamos matrizes L diferentes da matriz identidade. As Figuras 1.7 e 1.8 mostram soluções en-

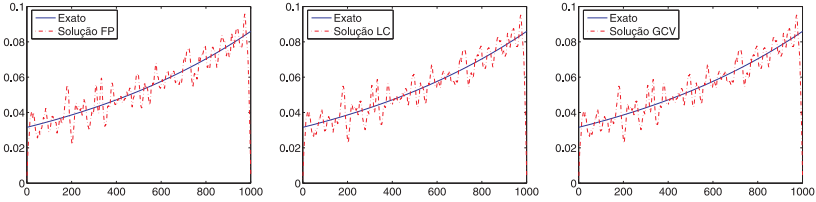


Figura 1.6: Solução do problema *Deriv2* de dimensão 1000 com 0,1% de ruído nos dados, utilizando regularização de Tikhonov na forma padrão, ou seja, $L = I$. Erros relativos: $E_{FP} = 14,56\%$, $E_{LC} = 14,39\%$ e $E_{GCV} = 14,39\%$.

contradas utilizando as seguintes matrizes, respectivamente:

$$L_1 = \begin{bmatrix} -1 & 1 & & & \\ & \ddots & \ddots & & \\ & & & -1 & 1 \\ & & & & \end{bmatrix} \in \mathbb{R}^{999 \times 1000} \quad (1.37)$$

sendo uma aproximação discreta para o operador de primeira derivada e

$$L_2 = \begin{bmatrix} 1 & -2 & 1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & & 1 & -2 & 1 \\ & & & & & \end{bmatrix} \in \mathbb{R}^{998 \times 1000} \quad (1.38)$$

para a segunda derivada. Neste caso, usamos novamente ruído de 0,1% de ruído no lado direito \mathbf{b} e dimensão 1000. Em geral, para n dado, $L_1 \in \mathbb{R}^{(n-1) \times n}$ e $L_2 \in \mathbb{R}^{(n-2) \times n}$.

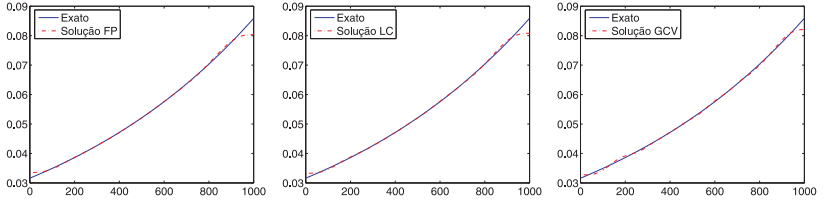


Figura 1.7: Solução do problema *Deriv2* de dimensão 1000 com 0,1% de ruído nos dados, utilizando regularização de Tikhonov na forma geral para $L = L_1$. Erros relativos: $E_{FP} = 1,64\%$, $E_{LC} = 1,43\%$ e $E_{GCV} = 1,05\%$.

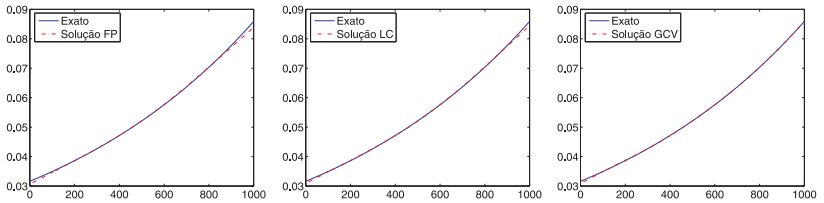


Figura 1.8: Solução do problema *Deriv2* de dimensão 1000 com 0,1% de ruído nos dados, utilizando regularização de Tikhonov na forma geral para $L = L_2$. Erros relativos: $E_{FP} = 0,86\%$, $E_{LC} = 0,57\%$ e $E_{GCV} = 0,28\%$.

Capítulo 2

Métodos Básicos para Problemas de Grande Porte

A seguir serão apresentados conceitos fundamentais para o desenvolvimento dos métodos de regularização introduzidos por este trabalho, dentre eles o método GKB-FP introduzido por Bazán e Borges [5], que serve de base para o método proposto GGKB-FP; e o processo de Bidiagonalização de Golub-Kahan (GKB) que induz o método proposto PROJ-FP, ambos apresentados no capítulo 4. Iniciamos tratando sobre os métodos de projeção, um dos conceitos que fundamentam os métodos propostos neste trabalho.

Dentre os métodos de regularização para problemas discretos mal postos, podemos destacar os métodos de penalidade, como por exemplo o método de Tikhonov, e os métodos de projeção. Também existem métodos chamados *híbridos*, que combinam um método de penalidade com um método de projeção. Nesta primeira seção discutiremos em detalhes os métodos de projeção.

2.1 Métodos de Projeção

Consideremos um problema discreto mal posto como em (1.10). Lembrando que $\mathbf{b} = \mathbf{b}_{\text{exato}} + \mathbf{e}$, sabemos que a solução calculada por métodos usuais é dominada pela influência do ruído \mathbf{e} .

A ideia básica dos métodos de projeção é aproximar

$$\mathbf{x}_{\text{exato}} = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^n} \{ \|\mathbf{b}_{\text{exato}} - A\mathbf{x}\|_2^2 \}.$$

por $\mathbf{x}^{(k)}$ em que

$$\mathbf{x}^{(k)} = \operatorname{argmin}_{\mathbf{x} \in \mathcal{V}_k} \{ \|A\mathbf{x} - \mathbf{b}\|_2^2 \} \quad (2.1)$$

onde $\{\mathcal{V}_k\}$ é uma sequência de subespaços de \mathbb{R}^n tais que

$$\mathcal{V}_1 \subset \mathcal{V}_2 \subset \dots \subset \mathcal{V}_k \subset \mathcal{V}_{k+1} \dots \subset \mathbb{R}^n, \quad \dim(\mathcal{V}_k) = k, \quad (2.2)$$

como na Figura 2.1.

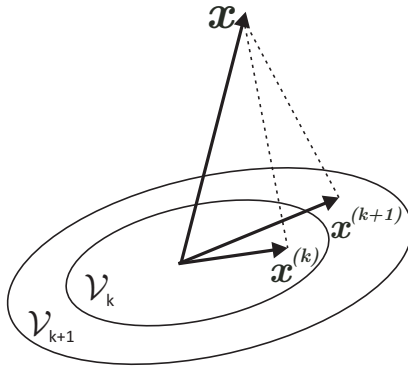


Figura 2.1: Solução aproximadas para $\mathbf{x}_{\text{exato}}$.

Tais métodos podem ser considerados métodos iterativos pois constroem uma sequência de soluções $\{\mathbf{x}^{(k)}\}$, $k = 1, 2, \dots$ onde $\mathbf{x}^{(k)} \in \mathcal{V}_k$.

Métodos de projeção são usualmente utilizados em problemas cuja dimensão inviabiliza o cálculo da SVD de A devido ao alto

custo computacional. Outra característica desses métodos é que em geral boas soluções são obtidas com poucas iterações, mas se o processo iterativo prossegue, as novas componentes passam a incorporar mais contribuições do ruído, o que pode desestabilizar a solução.

Sendo assim, nos métodos de projeção, devemos estabelecer um bom critério de parada, fazendo com que a solução calculada seja a melhor possível.

Como exemplos clássicos de métodos de projeção, podemos destacar:

- SVD Truncada (TSVD) e GSVD Truncada (TGSVD) [26];
- Método dos Gradientes Conjugados (CG) [8];
- Método Least Square QR (LSQR) [41];
- Método dos Resíduos Mínimos Generalizados (GMRES) [46].

O método LSQR será apresentado com detalhes neste capítulo, mas informações adicionais sobre este e os demais métodos citados anteriormente podem ser obtidas em [26, 41].

Podemos também aplicar a ideia de “projeção” para a regularização de Tikhonov, isto é, calcular $\mathbf{x}_\lambda^{(k)}$ solução aproximada e regularizada para a solução exata como

$$\mathbf{x}_\lambda^{(k)} = \operatorname{argmin}_{\mathbf{x} \in \mathcal{V}_k} \{ \|\mathbf{b} - A\mathbf{x}\|_2^2 + \lambda^2 \|L\mathbf{x}\|_2^2 \}. \quad (2.3)$$

Esta estratégia combina um método de projeção, excelente para problemas grandes, com um método de penalidade. Como veremos no próximo capítulo, quando \mathcal{V}_k é escolhido de forma conveniente, podemos obter um problema equivalente ao problema (2.3), cuja solução pode ser calculada de forma muito eficiente.

Métodos que seguem a ideia de obter aproximações $\mathbf{x}^{(k)} \in \mathcal{V}_k$ poderiam mais corretamente ser denominados *métodos de restrições*, pois calculam a solução do problema (1.10) restrita ao subespaço \mathcal{V}_k .

A denominação *projeção* seria mais corretamente utilizada se $\mathbf{x}^{(k)} = P_k \mathbf{x}_{\text{exato}}$, onde P representa algum projetor sobre \mathcal{V}_k .

O teorema 2.1 mostra que a solução do problemas regularizado “projetado” (2.3), $\mathbf{x}_\lambda^{(k)}$, é de fato uma projeção da solução do problema regularizado

$$\mathbf{x}_\lambda = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^n} \left\{ \|\mathbf{b} - A\mathbf{x}\|_2^2 + \lambda^2 \|L\mathbf{x}\|_2^2 \right\}. \quad (2.4)$$

Teorema 2.1. *Assuma que $\mathcal{N}(A) \cap \mathcal{N}(L) = \{\mathbf{0}\}$. Sejam \mathbf{x}_λ solução de (2.4) e $\mathbf{x}_\lambda^{(k)}$ solução da equação (2.3). Então $\mathbf{x}_\lambda^{(k)} = P\mathbf{x}_\lambda$ em que P é um projetor ortogonal adequado.*

Demonstração. Visto que $\mathcal{N}(A)$ e $\mathcal{N}(L)$ têm intersecção trivial, é imediato ver que a função $\langle \cdot, \cdot \rangle_\# : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ dada por

$$\langle \mathbf{u}, \mathbf{v} \rangle_\# = \langle A\mathbf{u}, A\mathbf{v} \rangle + \lambda^2 \langle L\mathbf{u}, L\mathbf{v} \rangle \quad (2.5)$$

onde $\langle \cdot, \cdot \rangle$ denota o produto interno usual em \mathbb{R}^n , define um produto interno em \mathbb{R}^n .

Agora, note que a solução regularizada \mathbf{x}_λ satisfaz

$$(A^T A + \lambda^2 L^T L) \mathbf{x}_\lambda = A^T \mathbf{b},$$

e que isto pode ser usado para provar que

$$\langle A\mathbf{x}_\lambda - \mathbf{b}, A\mathbf{x} \rangle + \lambda^2 \langle L\mathbf{x}_\lambda, L\mathbf{x} \rangle = 0, \quad \forall \mathbf{x} \in \mathbb{R}^n. \quad (2.6)$$

De fato, podemos reescrever o problema (2.4) como

$$\mathbf{x}_\lambda = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^n} \left\{ \left\| \begin{bmatrix} A \\ \lambda L \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\|_2^2 \right\}. \quad (2.7)$$

Assim, se denotarmos $\tilde{A} = [A^T \ \lambda L^T]^T$ e $\tilde{\mathbf{b}} = [\mathbf{b}^T \ \mathbf{0}^T]^T$ temos que $\mathbf{x}_\lambda = P_{\mathcal{R}(\tilde{A})} \tilde{\mathbf{b}}$ e, portanto,

$$\tilde{\mathbf{b}} - \tilde{A}\mathbf{x}_\lambda \perp \mathcal{R}(\tilde{A}).$$

Isto significa que $\tilde{\mathbf{b}} - \tilde{A}\mathbf{x}_\lambda$ é ortogonal a $\tilde{A}\mathbf{x}$ para qualquer $\mathbf{x} \in \mathbb{R}^n$. Logo

$$\left\langle \begin{bmatrix} A \\ \lambda L \end{bmatrix} \mathbf{x}_\lambda - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} A \\ \lambda L \end{bmatrix} \mathbf{x} \right\rangle = 0, \quad \forall \mathbf{x} \in \mathbb{R}^n,$$

$$\langle A\mathbf{x}_\lambda - \mathbf{b}, A\mathbf{x} \rangle + \lambda^2 \langle L\mathbf{x}_\lambda, L\mathbf{x} \rangle = 0, \quad \forall \mathbf{x} \in \mathbb{R}^n.$$

De forma semelhante, para o problema regularizado “projetado” (2.3), temos

$$\langle A\mathbf{x}_\lambda^{(k)} - \mathbf{b}, A\boldsymbol{\varphi} \rangle + \lambda^2 \langle L\mathbf{x}_\lambda^{(k)}, L\boldsymbol{\varphi} \rangle = 0, \quad \forall \boldsymbol{\varphi} \in \mathcal{V}^k. \quad (2.8)$$

Combinando as equações (2.6) e (2.8) obtemos

$$\langle \mathbf{x}_\lambda - \mathbf{x}_\lambda^{(k)}, \boldsymbol{\varphi} \rangle_\# = 0, \quad \forall \boldsymbol{\varphi} \in \mathcal{V}^k. \quad (2.9)$$

De fato, tome $\boldsymbol{\varphi} \in \mathcal{V}^k$. Escrevendo (2.6) para $\mathbf{x} = \boldsymbol{\varphi}$ e subtraindo a equação (2.8) da equação resultante obtemos

$$\begin{aligned} 0 &= \langle A\mathbf{x}_\lambda - \mathbf{b} - A\mathbf{x}_\lambda^{(k)} + \mathbf{b}, A\boldsymbol{\varphi} \rangle + \lambda^2 \langle L\mathbf{x}_\lambda - L\mathbf{x}_\lambda^{(k)}, L\boldsymbol{\varphi} \rangle \\ &= \langle A(\mathbf{x}_\lambda - \mathbf{x}_\lambda^{(k)}), A\boldsymbol{\varphi} \rangle + \lambda^2 \langle L(\mathbf{x}_\lambda - \mathbf{x}_\lambda^{(k)}), L\boldsymbol{\varphi} \rangle \\ &= \langle \mathbf{x}_\lambda - \mathbf{x}_\lambda^{(k)}, \boldsymbol{\varphi} \rangle_\#. \end{aligned}$$

A equação (2.9) significa que $\mathbf{x}_\lambda^{(k)} = \mathcal{P}_k \mathbf{x}_\lambda$, onde \mathcal{P}_k denota o projetor ortogonal sobre o subespaço \mathcal{V}_k , com respeito ao produto interno $\langle \cdot, \cdot \rangle_\#$. ■

2.2 Bidiagonalização de Golub-Kahan

A ideia da bidiagonalização de Golub-Kahan (GKB) é explorar a decomposição $A = UBV^T$ em que as matrizes $U \in \mathbb{R}^{m \times (n+1)}$ e $V \in \mathbb{R}^{n \times n}$ têm colunas ortonormais e a matriz $B \in \mathbb{R}^{(n+1) \times n}$ é bidiagonal inferior. Da ortogonalidade da matriz V é imediato que

$$AV = UB \quad \text{e} \quad A^T U = VB^T. \quad (2.10)$$

Para obter tal fatoração, aplica-se um processo iterativo que no k -ésimo passo produz as matrizes

$$\begin{aligned} U_{k+1} &= [\mathbf{u}_1, \dots, \mathbf{u}_{k+1}] \in \mathbb{R}^{m \times (k+1)}, & (2.11) \\ V_k &= [\mathbf{v}_1, \dots, \mathbf{v}_k] \in \mathbb{R}^{n \times k} \quad \text{e} \\ B_k &= \begin{bmatrix} \alpha_1 & & & & \\ \beta_2 & \alpha_2 & & & \\ & \beta_3 & \ddots & & \\ & & \ddots & \alpha_k & \\ & & & \beta_{k+1} & \end{bmatrix} \in \mathbb{R}^{(k+1) \times k} \end{aligned}$$

tais que

$$AV_k = U_{k+1}B_k. \quad (2.12)$$

Pelo fato da matriz B ser bidiagonal podemos equacionar as colunas na equação (2.10) e definindo $\beta_1 \mathbf{v}_0 \equiv 0$, $\alpha_{k+1} \mathbf{v}_{k+1} \equiv 0$ obtemos

$$A^T \mathbf{u}_j = \beta_j \mathbf{v}_{j-1} + \alpha_j \mathbf{v}_j \quad (2.13)$$

$$A \mathbf{v}_j = \alpha_j \mathbf{u}_j + \beta_{j+1} \mathbf{u}_{j+1}, \quad j = 1, \dots, k. \quad (2.14)$$

Começando com um vetor $\mathbf{u}_1 \in \mathbb{R}^m$ com $\|\mathbf{u}_1\|_2 = 1$ e $\alpha_1 = \|A\mathbf{u}_1\|_2$, podemos recursivamente gerar os vetores

$$\mathbf{v}_1, \mathbf{u}_2, \mathbf{v}_2, \dots, \mathbf{u}_k, \mathbf{v}_k, \mathbf{u}_{k+1}$$

e os elementos respectivos na matriz B_k .

No entanto, na prática, podem surgir dificuldades numéricas devido a aritmética finita utilizada pelo computador, fazendo com que os vetores \mathbf{u}_j e \mathbf{v}_j (colunas de U_k e V_k) percam ortogonalidade. Para evitar isso, a bidiagonalização GKB é feita utilizando reortogonalização.

A reortogonalização garante que cada novo vetor \mathbf{u}_i gerado seja ortogonal aos demais vetores \mathbf{u}_j para $j < i$. O mesmo ocorre com \mathbf{v}_j . Isso é obtido aplicando-se o processo de Ortogonalização de Gram-Schmidt aos conjuntos

$$\{\mathbf{u}_1, \dots, \mathbf{u}_j, \dots, \mathbf{u}_i\}$$

$$\{\mathbf{v}_1, \dots, \mathbf{v}_j, \dots, \mathbf{v}_i\}.$$

Para uma discussão do uso de reortogonalização em um algoritmo híbrido da bidiagonalização de Golub-Kahan para problemas mal postos ver [7].

Definição 2.1 (Subespaço de Krylov). *O subespaço de Krylov de ordem r gerado por $A \in \mathbb{R}^{n \times n}$ e $\mathbf{b} \in \mathbb{R}^n$ é o subespaço gerado pelas imagens de \mathbf{b} sob as r primeiras potências de A (começando com $A^0 = I$), isto é,*

$$\mathcal{K}_r(A, \mathbf{b}) = \text{span} \{\mathbf{b}, A\mathbf{b}, A^2\mathbf{b}, \dots, A^{r-1}\mathbf{b}\}. \quad (2.15)$$

Uma propriedade importante sobre os \mathbf{u}_j e \mathbf{v}_j é estabelecida na seguinte proposição.

Proposição 2.1. *Para \mathbf{u}_j e \mathbf{v}_j temos $\mathbf{u}_j \in \mathcal{K}_j(AA^T, u_1)$ e $\mathbf{v}_j \in \mathcal{K}_j(A^T A, A^T u_1)$.*

A demonstração da proposição anterior pode ser encontrada em [8], embora a verificação possa ser feita facilmente usando as equações (2.13) e (2.14) e de $\beta_1 \mathbf{v}_0 = 0$.

Se escolhermos como vetor inicial \mathbf{u}_1 o vetor $\frac{\mathbf{b}}{\|\mathbf{b}\|}$ teremos, pela proposição 2.1, que os vetores \mathbf{u}_j e \mathbf{v}_j para $j = 1, 2, \dots$ pertencerão aos espaços $\mathcal{K}_j(AA^T, \mathbf{b})$ e $\mathcal{K}_j(A^T A, A^T \mathbf{b})$, respectivamente. Além disso, $\beta_1 \mathbf{u}_1 = \mathbf{b}$ e $\alpha_1 \mathbf{v}_1 = A^T \mathbf{u}_1$, e usando as equações (2.12) e (2.14) obtemos no passo k da bidiagonalização matrizes U_{k+1} , V_k e B_k satisfazendo

$$\beta_1 U_{k+1} \mathbf{e}_1 = \mathbf{b}, \quad (2.16)$$

$$\begin{cases} AV_k = U_{k+1} B_k \\ A^T U_{k+1} = V_k B_k^T + \alpha_{k+1} v_{k+1} e_{k+1}^T \end{cases}. \quad (2.17)$$

onde \mathbf{e}_k denota o k -ésimo vetor canônico de dimensão apropriada.

2.3 Método LSQR

O método LSQR, apresentado por Paige e Saunders [41], produz uma sequência (finita) de soluções aproximadas $\mathbf{x}^{(k)}$, $k \geq 1$ para

o problema de mínimos quadrados

$$\min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{b} - A\mathbf{x}\|_2 \quad (2.18)$$

com $\mathbf{x}^{(k)} \in \mathcal{K}_k(A^T A, A^T \mathbf{b})$. Para tanto, note da Proposição 2.1 que $\mathcal{K}_k(A^T A, A^T \mathbf{b}) = \text{span}(V_k)$ e assim existe $\mathbf{y}^{(k)} \in \mathbb{R}^k$ tal que

$$\mathbf{x}^{(k)} = V_k \mathbf{y}^{(k)}. \quad (2.19)$$

Agora consideremos o resíduo $r^{(k)} = \mathbf{b} - A\mathbf{x}^{(k)}$. Utilizando a equação (2.12) e o fato que $\beta_1 \mathbf{u}_1 = \mathbf{b}$ temos

$$\begin{aligned} r^{(k)} &= \mathbf{b} - AV_k \mathbf{y}^{(k)} \\ &= \beta_1 \mathbf{u}_1 - U_{k+1} B_k \mathbf{y}^{(k)} \\ &= \beta_1 U_{k+1} \mathbf{e}_1 - U_{k+1} B_k \mathbf{y}^{(k)} \\ &= U_{k+1} \left(\beta_1 \mathbf{e}_1 - B_k \mathbf{y}^{(k)} \right) \end{aligned}$$

em que \mathbf{e}_1 é o primeiro vetor canônico de \mathbb{R}^{k+1} . Assim

$$\|r^{(k)}\| = \|\beta_1 \mathbf{e}_1 - B_k \mathbf{y}^{(k)}\|_2 \quad (2.20)$$

pois $U_{k+1}^T U_{k+1} = I_{k+1}$. Portanto

$$\mathbf{x}^{(k)} = \underset{\mathbf{x} \in \mathcal{K}_k(A^T A, A^T \mathbf{b})}{\text{argmin}} \left\{ \|\mathbf{b} - A\mathbf{x}\|_2^2 \right\} \quad (2.21)$$

equivale a

$$\mathbf{x}^{(k)} = V_k \mathbf{y}^{(k)}, \quad \mathbf{y}^{(k)} = \underset{\mathbf{y} \in \mathbb{R}^k}{\text{argmin}} \|\beta_1 \mathbf{e}_1 - B_k \mathbf{y}\|_2.$$

Um resultado importante para as soluções $\mathbf{x}^{(k)}$ é que enquanto sequência das normas da solução $\mathbf{x}^{(k)}$ cresce monotonamente com k , as normas do resíduo correspondente $r^{(k)} = \mathbf{b} - A\mathbf{x}^{(k)}$ decrescem monotonamente com k , isto é,

$$\|\mathbf{x}^{(k+1)}\|_2 \geq \|\mathbf{x}^{(k)}\|_2 \quad \text{e} \quad \|r^{(k+1)}\|_2 \leq \|r^{(k)}\|_2. \quad (2.22)$$

Maiores detalhes sobre o método LSQR, incluindo o algoritmo detalhado, podem ser encontrados em [41].

Considere novamente o problema de Tikhonov associado a equação (1.10), com $L = I$ e $\mathbf{x}_0 = \mathbf{0}$, escrito na forma de um problema de mínimos quadrados usual

$$\mathbf{x}_\lambda = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^n} \left\| \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} A \\ \lambda I_n \end{bmatrix} \mathbf{x} \right\|_2. \quad (2.23)$$

O método LSQR pode ser usado para resolver este problema, sendo λ um parâmetro de regularização fixado. Neste caso, a k -ésima solução aproximada é dada por $\mathbf{x}_\lambda^{(k)} = V_k \mathbf{y}_\lambda^{(k)}$, mas agora $\mathbf{y}_\lambda^{(k)}$ é a solução do problema “projetado regularizado”

$$\mathbf{y}_\lambda^{(k)} = \operatorname{argmin}_{\mathbf{y} \in \mathbb{R}^k} \left\| \begin{bmatrix} \beta_1 \mathbf{e}_1 \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} B_k \\ \lambda I_k \end{bmatrix} \mathbf{y} \right\|_2, \quad (2.24)$$

que é resolvido eficientemente usando a fatoração QR de $\begin{bmatrix} B_k \\ \lambda I_k \end{bmatrix}$,

$$Q_k \begin{bmatrix} B_k \\ \lambda I_k \end{bmatrix} = \begin{bmatrix} R_k \\ 0 \\ 0 \end{bmatrix} \quad \text{e} \quad Q_k \begin{bmatrix} \beta_1 \mathbf{e}_1 \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_k \\ \bar{\varphi}_{k+1} \\ \mathbf{c}_k \end{bmatrix}, \quad (2.25)$$

onde R_k é bidiagonal superior,

$$\mathbf{b}_k = \begin{bmatrix} \varphi_1 \\ \vdots \\ \varphi_k \end{bmatrix}, \quad \mathbf{c}_k = \begin{bmatrix} \psi_1 \\ \vdots \\ \psi_k \end{bmatrix} \quad (2.26)$$

e Q_k tem ordem $2k + 1$ e é um produto de duas seqüências de rotações planas. A solução $\mathbf{y}_\lambda^{(k)}$ pode então ser obtida usando um sistema triangular superior $R_k \mathbf{y} = \mathbf{b}_k$. Então $\mathbf{x}_\lambda^{(k)}$ é obtido substituindo a solução $\mathbf{y}_\lambda^{(k)}$ em $\mathbf{x}_\lambda^{(k)} = V_k \mathbf{y}_\lambda^{(k)}$. Alternativamente, como mostrado em [41], para λ fixo, a solução $\mathbf{x}_\lambda^{(k)}$ pode ser calculada através da atualização da fórmula

$$\mathbf{x}_\lambda^{(k)} = \mathbf{x}_\lambda^{(k-1)} + \varphi_k \mathbf{d}_k \quad (2.27)$$

onde \mathbf{d}_k é a última coluna de $D_k := V_k R_k^{-1}$, podendo ser calculada sem o armazenamento da matriz V_k .

2.4 Método GKB-FP

Este método combina o algoritmo de bidiagonalização de Golub-Kahan com regularização de Tikhonov em subespaços de Krylov, com o parâmetro de regularização para o problema projetado sendo escolhido pelo Método de Ponto Fixo de Bazán [3]. Sua eficiência para problemas discretos mal postos de grande porte é verificada pelos resultados numéricos apresentados em [5]. No entanto, o método GKB-FP foi desenvolvido para o caso em que o problema regularizado está na forma padrão, ou seja, $L = I$.

Mais detalhadamente, considere as funções

$$\Psi_\mu(\lambda) = \|\mathbf{b} - A\mathbf{x}_\lambda\|_2^2 \left(\|\mathbf{x}_\lambda\|_2^2 \right)^\mu \quad (2.28)$$

e

$$\varphi_\mu(\lambda) = \sqrt{\mu} \frac{\|\mathbf{b} - A\mathbf{x}_\lambda\|_2}{\|\mathbf{x}_\lambda\|_2} \quad (2.29)$$

em que \mathbf{x}_λ é como na equação (1.17).

Considere agora um sequência finita de funções $\varphi_\mu^{(k)} : \mathbb{R}^+ \rightarrow \mathbb{R}$, para $k = 1, \dots, n$ dada por

$$\varphi_\mu^{(k)}(\lambda) = \sqrt{\mu} \frac{\left\| \beta_1 \mathbf{e}_1 - B_k \mathbf{y}_\lambda^{(k)} \right\|_2}{\left\| \mathbf{y}_\lambda^{(k)} \right\|_2}, \quad \mu > 0 \quad (2.30)$$

e $\mathbf{y}_\lambda^{(k)}$ da equação (2.24).

Usando a SVD de B_k e derivando a equação (2.30) com respeito a λ obtemos que $\varphi_\mu^{(k)}(\lambda) > 0$ para cada $\lambda > 0$. Isto é, a função $\varphi_\mu^{(k)}(\lambda)$ é estritamente crescente, para cada k fixo.

Agora, iniciando com $\lambda_0^{(k)}$, consideramos a sequência

$$\lambda_{j+1}^{(k)} = \varphi_1^{(k)}(\lambda_j^{(k)}), \quad j \geq 0 \quad (2.31)$$

e assumamos que esta converge para um ponto fixo $\lambda^{(k)*}$ de $\varphi_1^{(k)}$. Quando isto é verdade e as componentes de \mathbf{b} que estão fora de $\mathcal{R}(A)$ não são nulas, é mostrado que a função φ_1 associada ao problema original, como na equação (2.29), tem um ponto fixo λ^* que minimiza Ψ_1 e que a sequência $\lambda^{(k)*}$ converge para λ^* . Também temos que λ^* é o maior ponto fixo convexo de φ_1 , que é o ponto fixo que produz o melhor parâmetro de regularização, conforme mostrado em [4].

Os detalhes sobre a origem das funções Ψ_1 e φ_1 podem ser encontrados em [3, 4], bem como a discussão sobre pontos fixos convexos e côncavos.

GKB-FP constrói uma aproximação para λ^* usando uma sequência finita não-crescente de pontos fixos $\lambda^{(k)*}$. Isto requer resolver o problema (2.24) para vários valores de λ para k fixado, mas crescente. GKB-FP usa a ideia do método LSQR para resolver cada um dos problemas projetados de maneira eficiente. Todos os detalhes podem ser encontrados em [5].

Em [5] é mostrado que se $\lambda^{(k)*} \rightarrow \lambda^*$ em um número finito de passos, então a solução regularizada $\mathbf{x}_{\lambda^*}^{(k)}$ permanece estável e portanto completamente insensível ao número de iterações.

2.4.1 Algoritmo

A Tabela 2.1 traz o algoritmo do método GKB-FP resumidamente. Os passos apresentados neste momento são de extrema importância para compreender um dos métodos propostos no capítulo 4.

Os parâmetros $q > 1$, $kmax$, ϵ_1 e ϵ_2 representam dimensão inicial do problema projetado, número máximo de iterações e parâmetros usados no critério de parada, respectivamente. Como padrão, usamos $\epsilon_1 = \epsilon_2 = 10^{-4}$, $kmax = n - 1$ e $q = 5$. Neste caso, os dados de saída k^* e λ^* são, respectivamente, a dimensão máxima dos espaços projetados e o parâmetro de regularização utilizado para calcular a solução $\mathbf{x}_{\lambda^*}^{(k)}$.

GKB-FP

Entrada: $A, \mathbf{b}, q > 1, kmax, \epsilon_1$ e ϵ_2 .

Saída: Solução regularizada $\mathbf{x}_{\lambda^*}^{(k)}$, k^* e λ^* .

1. Aplicar q passos GKB para A iniciando com $\mathbf{u}_1 = \frac{\mathbf{b}}{\|\mathbf{b}\|}$;
Formar a matriz B_q .
2. Definir $k = q$;
Calcular o ponto fixo $\lambda^{(k)*}$ de $\varphi_1^{(k)}$;
Definir $\lambda_0 = \lambda^{(k)*}$, $\lambda_{old} = \lambda_0$ e $k \leftarrow k + 1$.
3. Executar mais um passo GKB e calcular o ponto fixo $\lambda^{(k)*}$ de $\varphi_1^{(k)}$ tomando λ_0 como valor inicial;
Definir $\lambda_{old} = \lambda_0$, $\lambda_0 = \lambda^{(k)*}$.
4. **Se** (critério de parada satisfeito)
 $\lambda^* = \lambda_{old}$, $k^* = k$
Senão
 $k \leftarrow k + 1$ e ir para o passo 3.
Fim Se
5. Calcular a solução regularizada $\mathbf{x}_{\lambda^*}^{(k)}$.

Tabela 2.1: Algoritmo GKB-FP

2.4.2 Critério da Parada

É conhecido que o processo GKB captura a parte dominante do espectro singular da matriz A do problema original em um número relativamente pequeno de passos. Com isso, $\varphi_1^{(k)}(\lambda)$ deve aproximar $\varphi_1(\lambda)$ relativamente bem, pelo menos para λ em algum intervalo $[a_k, \sigma_1^{(k)}]$ com $\sigma_1^{(k)}$ sendo o maior valor singular de B_k . Neste caso, a_k é um valor maior do que o k -ésimo valor singular de B_k , $\sigma_k^{(k)}$, e perto de algum valor singular dominante de A já capturado no passo k .

Seja λ^* o ponto fixo procurado associado com o problema original. Então, se para certo k , $\lambda^* \leq a_k$, é razoável que a sequência de pontos fixos $\lambda^{(k)*}$ estabilize próximo de λ^* em poucos passos GKB. Isto é motivado pelo fato que $\varphi_1^{(k)}(\lambda)$ aproxima $\varphi_1(\lambda)$ por

cima e que a sequência de pontos fixos $\lambda^{(k)*}$ é não crescente, conforme provado em [5].

Baseado nisto, o processo iterativo para quando a mudança relativa de pontos fixos consecutivos é pequena, isto é,

$$|\lambda^{(k+1)*} - \lambda^{(k)*}| < \epsilon_1 |\lambda^{(k)*}| \quad (2.32)$$

onde ϵ_1 é um parâmetro de tolerância pequeno.

Uma desvantagem desse critério é que ele pode demorar a ser atingido quando a sequência dos $\lambda^{(k)*}$ decresce muito devagar, o que pode elevar o custo computacional. Para contornar esta dificuldade, o método GKB-FP usa também um segundo critério, definido por

$$|\lambda^{(k+1)*} - \lambda^{(k)*}| < \epsilon_2 |\lambda^{(0)*}|, \quad (2.33)$$

onde $\lambda^{(0)*}$ é o ponto fixo calculado no passo 2 do algoritmo e ϵ_2 é outro parâmetro de tolerância.

O algoritmo para quando um dos critérios for satisfeito. Mais detalhes sobre os critérios de parada estão expostos em [5].

Capítulo 3

Transformação para a Forma Padrão

Visto que já existem métodos eficientes para problemas de grande porte que exploram o método de Tikhonov na forma padrão – por exemplo, W-GCV e GKB-FP – uma alternativa para lidar com

$$\mathbf{x}_\lambda = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^n} \left\{ \|\mathbf{b} - A\mathbf{x}\|_2^2 + \lambda^2 \|L(\mathbf{x} - \mathbf{x}_0)\|_2^2 \right\} \quad (3.1)$$

é transformá-lo para um problema na forma padrão.

Neste capítulo mostramos que se \mathbf{x}_λ é decomposto em duas componentes

$$\mathbf{x}_\lambda = \mathbf{x}_1 + \mathbf{x}_2$$

em que $\mathbf{x}_2 \in \mathcal{N}(L)$, então o problema (3.1) pode ser transformado na forma padrão

$$\mathbf{y}_\lambda = \operatorname{argmin}_{\mathbf{y} \in \mathbb{R}^p} \left\{ \|\bar{\mathbf{b}} - \bar{A}\mathbf{y}\|_2^2 + \lambda^2 \|\mathbf{y} - \mathbf{y}_0\|_2^2 \right\}, \quad (3.2)$$

onde quantidades \bar{A} , $\bar{\mathbf{b}}$ e \mathbf{y}_0 dependem de A , \mathbf{b} , \mathbf{x}_0 e L . A partir da solução \mathbf{y}_λ , então, determinamos \mathbf{x}_λ solução regularizada de (3.1).

Para obter tal transformação, exploramos os conceitos de *projeções oblíquas* e *pseudo-inversa oblíqua*. Algumas das definições e resultados que servirão para demonstrar propriedades

importantes destas projeções estão expostas no apêndice A, seção A.1, como a definição de matriz pseudo-inversa, também conhecida como pseudo-inversa de Moore-Penrose, e suas propriedades.

3.1 Projeções Oblíquas

Qualquer matriz idempotente P , isto é, $P^2 = P$, é uma projeção sobre $\mathcal{R}(P)$ com relação a $\mathcal{R}(P^T)$. Se P é também simétrica então P é uma projeção ortogonal. Projetores que não são ortogonais são usualmente chamados de *projetores oblíquos* para enfatizar o fato que os subespaços envolvidos não são, necessariamente, ortogonais.

Considere dois subespaços \mathcal{X} e \mathcal{Y} de \mathbb{R}^n . Uma interpretação geométrica pode ser dada considerando a decomposição de um vetor arbitrário $s \in \mathbb{R}^n$ em três componentes

$$x \in \mathcal{X}, \quad y \in \mathcal{Y} \quad \text{e} \quad s \in (\mathcal{X} + \mathcal{Y})^\perp. \quad (3.3)$$

ilustrado pela Figura 3.1. O vetor x é a projeção oblíqua de z em \mathcal{X} com relação a \mathcal{Y} e denotamos

$$x = E_{\mathcal{X}, \mathcal{Y}} z.$$

Se $\mathcal{X} + \mathcal{Y} = \mathbb{R}^n$ então $s = \mathbf{0}$.

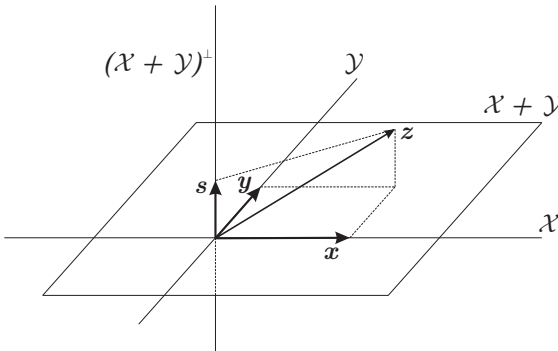


Figura 3.1: Decomposição de um vetor s arbitrário.

Definição 3.1. *Sejam \mathcal{X} e \mathcal{Y} subespaços de \mathbb{R}^n com intersecção trivial, isto é, $\mathcal{X} \cap \mathcal{Y} = \{\mathbf{0}\}$. Então a projeção oblíqua sobre \mathcal{X} com relação a \mathcal{Y} é o operador linear $E_{\mathcal{X},\mathcal{Y}}$ que satisfaz as três seguintes condições:*

- (a) $E_{\mathcal{X},\mathcal{Y}}\mathbf{x} = \mathbf{x}, \forall \mathbf{x} \in \mathcal{X};$
- (b) $E_{\mathcal{X},\mathcal{Y}}\mathbf{y} = \mathbf{0}, \forall \mathbf{y} \in \mathcal{Y};$
- (c) $E_{\mathcal{X},\mathcal{Y}}\mathbf{z} \in \mathcal{X}, \forall \mathbf{z} \in \mathbb{R}^n.$

Na Figura 3.1 temos $E_{\mathcal{X},\mathcal{Y}}\mathbf{s} = \mathbf{x}$ e $E_{\mathcal{Y},\mathcal{X}}\mathbf{s} = \mathbf{y}$.

Em busca da representação matricial do projetor $E_{\mathcal{X},\mathcal{Y}}$ consideremos matrizes $X \in \mathbb{R}^{n \times p}$ com $\text{posto}(X) = p$ e $Y \in \mathbb{R}^{n \times q}$ com $\text{posto}(Y) = q$ tais que $\mathcal{R}(X) = \mathcal{X}$ e $\mathcal{R}(Y) = \mathcal{Y}$. Como $\mathcal{X} \cap \mathcal{Y} = \{\mathbf{0}\}$ segue que $p+q \leq n$ onde $p = \dim(X)$ e $q = \dim(Y)$.

Assim, dado $\mathbf{z} \in \mathbb{R}^n$ podemos escrevê-lo como

$$\mathbf{z} = \widehat{\mathbf{z}} + \overline{\mathbf{z}} \quad (3.4)$$

com $\widehat{\mathbf{z}} \in \mathcal{X} + \mathcal{Y}$ e $\overline{\mathbf{z}} \in (\mathcal{X} + \mathcal{Y})^\perp$, o que significa que existem vetores $\boldsymbol{\alpha} \in \mathbb{R}^p$ e $\boldsymbol{\beta} \in \mathbb{R}^q$ tais que

$$\widehat{\mathbf{z}} = X\boldsymbol{\alpha} + Y\boldsymbol{\beta} = P_{\mathcal{R}(A)}\mathbf{z}$$

com $A = [X \ Y]$ de posto coluna completo e $\mathcal{X} + \mathcal{Y} = \mathcal{R}(A) = \mathcal{R}(X) + \mathcal{R}(Y)$.

Já que $P_{\mathcal{R}(A)} = AA^\dagger$, segue da Proposição A.2 que

$$\widehat{\mathbf{z}} = AA^\dagger\mathbf{z} \quad (3.5)$$

$$= [X \ Y] \begin{bmatrix} (P_{\mathcal{Y}^\perp}X)^\dagger \\ (P_{\mathcal{X}^\perp}Y)^\dagger \end{bmatrix} \mathbf{z} \quad (3.6)$$

$$= X(P_{\mathcal{Y}^\perp}X)^\dagger\mathbf{z} + Y(P_{\mathcal{X}^\perp}Y)^\dagger\mathbf{z}$$

o que sugere que

$$E_{\mathcal{X},\mathcal{Y}} = X(P_{\mathcal{Y}^\perp}X)^\dagger \quad \text{e} \quad E_{\mathcal{Y},\mathcal{X}} = Y(P_{\mathcal{X}^\perp}Y)^\dagger.$$

Note ainda que das propriedades das matrizes de projeção temos

$$\begin{aligned}
 X (P_{\mathcal{Y}^\perp} X)^\dagger &= X (X^T P_{\mathcal{Y}^\perp}^T P_{\mathcal{Y}^\perp} X)^\dagger (P_{\mathcal{Y}^\perp} X)^T \\
 &= X (X^T P_{\mathcal{Y}^\perp}^2 X)^\dagger (P_{\mathcal{Y}^\perp} X)^T \\
 &= X (X^T P_{\mathcal{Y}^\perp} X)^\dagger (P_{\mathcal{Y}^\perp} X)^T \\
 &= X \left((P_{\mathcal{Y}^\perp} X)^T X \right)^\dagger (P_{\mathcal{Y}^\perp} X)^T.
 \end{aligned}$$

Assim, denotando $Y_0 \equiv P_{\mathcal{Y}^\perp} X$ temos

$$XY_0^\dagger = X (Y_0^T X)^\dagger Y_0^T. \quad (3.7)$$

O teorema 3.1 mostra que $E_{\mathcal{X}, \mathcal{Y}} = XY_0^\dagger$.

Lema 3.1. *Sejam $X \in \mathbb{R}^{p \times n}$ e $Y \in \mathbb{R}^{n \times m}$. Então $\mathcal{R}(XY) \subset \mathcal{R}(X)$ e se Y tem linhas linearmente independentes, $\mathcal{R}(X) \subset \mathcal{R}(XY)$.*

Demonstração. Se $\mathbf{y} \in \mathcal{R}(XY) \subset \mathbb{R}^p$ então existe $\mathbf{x} \in \mathbb{R}^m$ tal que $XY\mathbf{x} = \mathbf{y}$, isto é, $X(Y\mathbf{x}) = \mathbf{y}$. Logo, $\mathbf{y} \in \mathcal{R}(X)$. Por outro lado, para $\mathbf{y} \in \mathcal{R}(X) \subset \mathbb{R}^p$ existe $\mathbf{x} \in \mathbb{R}^n$ tal que $X\mathbf{x} = \mathbf{y}$. Como Y tem n linhas linearmente independentes e $\mathcal{R}(Y) \subset \mathbb{R}^n$ temos $\mathcal{R}(Y) = \mathbb{R}^n$. Assim, existe $\mathbf{z} \in \mathbb{R}^m$ tal que $Y\mathbf{z} = \mathbf{x}$. Portanto, $\mathbf{y} = XY\mathbf{z}$ e $\mathbf{y} \in \mathcal{R}(XY)$. ■

Teorema 3.1. *Sejam \mathcal{X} e \mathcal{Y} subespaços de \mathbb{R}^n com $\mathcal{X} \cap \mathcal{Y} = \{\mathbf{0}\}$, e X e Y_0 como anteriormente. Então a representação matricial do projetor $E_{\mathcal{X}, \mathcal{Y}}$ é dada por*

$$E_{\mathcal{X}, \mathcal{Y}} = X (Y_0^T X)^\dagger Y_0^T. \quad (3.8)$$

Demonstração. Vamos verificar que a matriz dada em (3.8) é idempotente. Note que, usando a propriedade (ii) da Definição A.1 temos

$$\begin{aligned}
 E_{\mathcal{X}, \mathcal{Y}}^2 &= X (Y_0^T X)^\dagger Y_0^T X (Y_0^T X)^\dagger Y_0^T \\
 &= X \left[(Y_0^T X)^\dagger Y_0^T X (Y_0^T X)^\dagger \right] Y_0^T \\
 &= X (Y_0^T X)^\dagger Y_0^T \\
 &= E_{\mathcal{X}, \mathcal{Y}}.
 \end{aligned}$$

Para verificar a condição (a) da Definição 3.1 devemos mostrar que $E_{\mathcal{X},\mathcal{Y}}X = X$. Primeiramente, vamos mostrar que a matriz $P_{\mathcal{Y}^\perp}X \in \mathbb{R}^{n \times p}$ tem posto completo. De fato, note que

$$P_{\mathcal{Y}^\perp}X = [P_{\mathcal{Y}^\perp}\mathbf{x}_1 \ P_{\mathcal{Y}^\perp}\mathbf{x}_2 \ \dots \ P_{\mathcal{Y}^\perp}\mathbf{x}_p].$$

Para $\boldsymbol{\alpha} \in \mathbb{R}^p$ considere a combinação linear nula

$$\alpha_1 P_{\mathcal{Y}^\perp}\mathbf{x}_1 + \alpha_2 P_{\mathcal{Y}^\perp}\mathbf{x}_2 + \dots + \alpha_p P_{\mathcal{Y}^\perp}\mathbf{x}_p = \mathbf{0}.$$

Como $P_{\mathcal{Y}^\perp} = I - P_{\mathcal{Y}}$ segue que

$$P_{\mathcal{Y}}X\boldsymbol{\alpha} = X\boldsymbol{\alpha},$$

Assim $X\boldsymbol{\alpha} \in \mathcal{R}(P_{\mathcal{Y}})$, $X\boldsymbol{\alpha} \in \mathcal{R}(X)$ e $X\boldsymbol{\alpha} = \mathbf{0}$ já que \mathcal{X} e \mathcal{Y} tem intersecção trivial. Logo $\boldsymbol{\alpha} = \mathbf{0}$, o que prova que as colunas de $P_{\mathcal{Y}^\perp}X \in \mathbb{R}^{n \times p}$ são linearmente independentes. Ou seja $Y_0 = P_{\mathcal{Y}^\perp}X$ tem posto coluna completo e igual a p . Agora, como $P_{\mathcal{Y}^\perp}$ é um projetor ortogonal portanto idempotente e simétrico, podemos reescrever $E_{\mathcal{X},\mathcal{Y}}X$ como

$$\begin{aligned} E_{\mathcal{X},\mathcal{Y}}X &= X(X^T P_{\mathcal{Y}^\perp}^T X)^\dagger X^T P_{\mathcal{Y}^\perp}^T X \\ &= X(X^T P_{\mathcal{Y}^\perp}^T P_{\mathcal{Y}^\perp} X)^\dagger X^T P_{\mathcal{Y}^\perp}^T P_{\mathcal{Y}^\perp} X \\ &= X(Y_0^T Y_0)^\dagger Y_0^T Y_0 \end{aligned}$$

Como Y_0 tem posto completo, segue que $Y_0^T Y_0$ também tem posto p . De fato, se $\mathbf{y} \in \mathcal{N}(Y_0^T Y_0)$ então $\mathbf{0} = Y_0^T Y_0 \mathbf{y}$ e, assim

$$\mathbf{0} = \mathbf{y}^T Y_0^T Y_0 \mathbf{y} = \|Y_0 \mathbf{y}\|_2^2 \Rightarrow Y_0 \mathbf{y} = \mathbf{0}.$$

Mas, como Y_0 tem colunas linearmente independentes, segue que $\mathbf{y} = \mathbf{0}$. Logo, $\mathcal{N}(Y_0^T Y_0) = \{\mathbf{0}\}$. Isto implica que $(Y_0^T Y_0)^\dagger Y_0^T Y_0 = I_p$ e, portanto, $E_{\mathcal{X},\mathcal{Y}}X = X$.

Para (b), tome $\mathbf{y} \in \mathcal{Y}$ então

$$Y_0^T \mathbf{y} = (P_{\mathcal{Y}^\perp} X)^T \mathbf{y} = X^T P_{\mathcal{Y}^\perp}^T \mathbf{y} = X^T \mathbf{0} = \mathbf{0}$$

pois as colunas de $P_{\mathcal{Y}^\perp}$ são ortogonais a \mathcal{Y} .

Finalizando, da expressão para $E_{\mathcal{X},\mathcal{Y}}$ dada em (3.8) implica $\mathcal{R}(E_{\mathcal{X},\mathcal{Y}}) \subset \mathcal{R}(X) = \mathcal{X}$, o que demonstra (c) da Definição 3.1. Portanto, $E_{\mathcal{X},\mathcal{Y}} = X(Y_0^T X)^\dagger Y_0^T$, com $Y_0 = P_{\mathcal{Y}^\perp} X$ é de fato a representação matricial da projeção sobre \mathcal{X} com relação a \mathcal{Y} . ■

Note da prova do teorema anterior que $\mathcal{R}(Y_0) \subset \mathcal{Y}^\perp$. O seguinte corolário mostra que o projetor oblíquo $E_{\mathcal{X},\mathcal{Y}}$ pode ser construído de maneira análoga ao caso anterior mas usando uma matriz Y_0 tal que

$$\mathcal{R}(Y_0) = \mathcal{Y}^\perp. \quad (3.9)$$

Corolário 3.1. *Sejam \mathcal{X} e \mathcal{Y} subespaços de \mathbb{R}^n com $\mathcal{X} \cap \mathcal{Y} = \{\mathbf{0}\}$. Assuma que são dadas matrizes X e Y_0 tais que $\mathcal{R}(X) = \mathcal{X}$ e Y_0 uma matriz satisfazendo (3.9). Então a representação matricial do projetor $E_{\mathcal{X},\mathcal{Y}}$ é dada por*

$$E_{\mathcal{X},\mathcal{Y}} = X(Y_0^T X)^\dagger Y_0^T. \quad (3.10)$$

A demonstração deste corolário segue os mesmos passos da prova do teorema 3.1.

Esta forma de representar o projetor $E_{\mathcal{X},\mathcal{Y}}$ pode ser mais conveniente para a análise teórica.

Seguindo um processo análogo, é possível provar que

$$E_{\mathcal{Y},\mathcal{X}} = Y(P_{\mathcal{X}^\perp} Y)^\dagger.$$

Observação 3.1. *Note que $E_{\mathcal{X},\mathcal{Y}}$ não é simétrica e, da relação*

$$E_{\mathcal{X},\mathcal{Y}}^T = Y_0(X^T Y_0)^\dagger X^T, \quad (3.11)$$

segue que $E_{\mathcal{X},\mathcal{Y}}^T$ não é necessariamente um projetor oblíquo, dependendo das dimensões de \mathcal{X} e \mathcal{Y} .

Se $\mathcal{X} = \mathcal{R}(X)$ e $\mathcal{Y} = \mathcal{X}^\perp$, então denotamos $E_{\mathcal{X},\mathcal{X}^\perp} = P_{\mathcal{X}}$ o projetor ortogonal sobre \mathcal{X} com relação a \mathcal{X}^\perp ou simplesmente, sobre \mathcal{X} . A representação matricial $P_{\mathcal{X}} = XX^\dagger$ já é bastante conhecida e, obviamente, coincide com a forma apresentada no Teorema 3.1

$$P_{\mathcal{X}} = X(X^T X)^\dagger X^T = XX^\dagger. \quad (3.12)$$

A simetria de $P_{\mathcal{X}}$, necessária para que este seja um projetor ortogonal, segue das propriedades de Moore-Penrose sobre X^\dagger . Projetores ortogonais também obedecem a relação

$$P_{\mathcal{X}} + P_{\mathcal{X}^\perp} = I_n, \quad (3.13)$$

para $P_{\mathcal{X}^\perp} = E_{\mathcal{X}^\perp, \mathcal{X}}$.

Em geral, se $\mathcal{X} + \mathcal{Y} = \mathbb{R}^n$, o teorema a seguir mostra que $E_{\mathcal{X}, \mathcal{Y}}^T$ torna-se um projetor e que há uma relação semelhante à (3.13) para projetores oblíquos.

Teorema 3.2. *Se $P_{\mathcal{X}+\mathcal{Y}}$ denota a projeção ortogonal sobre $\mathcal{X}+\mathcal{Y}$ então*

$$E_{\mathcal{X}, \mathcal{Y}} + E_{\mathcal{Y}, \mathcal{X}} = P_{\mathcal{X}+\mathcal{Y}} \quad e \quad E_{\mathcal{X}, \mathcal{Y}} + E_{\mathcal{Y}, \mathcal{X}} + P_{(\mathcal{X}+\mathcal{Y})^\perp} = I. \quad (3.14)$$

Em particular, se $\mathcal{X} + \mathcal{Y} = \mathbb{R}^n$ então

$$E_{\mathcal{X}, \mathcal{Y}}^T = E_{\mathcal{Y}^\perp, \mathcal{X}^\perp} \quad e \quad E_{\mathcal{Y}, \mathcal{X}} = I_n - E_{\mathcal{X}, \mathcal{Y}}. \quad (3.15)$$

Demonstração. A equação (3.14) segue de (3.6) e do Teorema 3.1.

Agora considere $\mathcal{X} + \mathcal{Y} = \mathbb{R}^n$. Como $\mathcal{X} \cap \mathcal{Y} = \{\mathbf{0}\}$ segue que $\mathcal{X} = \mathcal{Y}^\perp$ e $\mathcal{Y} = \mathcal{X}^\perp$. Assim

$$E_{\mathcal{X}, \mathcal{Y}}^T = E_{\mathcal{X}, \mathcal{X}^\perp}^T = E_{\mathcal{X}, \mathcal{X}^\perp} = E_{\mathcal{Y}^\perp, \mathcal{X}^\perp}, \quad (3.16)$$

enquanto $E_{\mathcal{X}, \mathcal{Y}} + E_{\mathcal{Y}, \mathcal{X}} = I_n$ decorre do fato de $(\mathcal{X} + \mathcal{Y})^\perp = \{\mathbf{0}\}$ e de (3.14). ■

Definição 3.2. *Se $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ e $A \in \mathbb{R}^{m \times n}$ então definimos $\mathbf{x} \perp_A \mathbf{y}$ por*

$$A\mathbf{x} \perp A\mathbf{y} \Leftrightarrow \mathbf{x}^T A^T A\mathbf{y} = 0. \quad (3.17)$$

Definição 3.3. *Se \mathcal{X} é um subespaço de \mathbb{R}^n e $A \in \mathbb{R}^{m \times n}$ de posto r então definimos o complemento oblíquo de \mathcal{X} com relação a A como*

$$\mathcal{X}^{\perp A} = \{\mathbf{y} \in \mathbb{R}^n ; \mathbf{x} \perp_A \mathbf{y} \forall \mathbf{x} \in \mathcal{X}\}. \quad (3.18)$$

É simples verificar que $\mathcal{X}^{\perp A}$ é um subespaço de \mathbb{R}^n satisfazendo:

i) $\mathcal{N}(A) \subset \mathcal{X}^{\perp A}$;

ii) Se $\mathcal{N}(A) \cap \mathcal{X} = \{\mathbf{0}\}$ então $\mathbb{R}^n = \mathcal{X} + \mathcal{X}^{\perp A}$.

O item (i) é imediato da definição 3.3 pois se $A\mathbf{y} = \mathbf{0}$ é claro que $\mathbf{0} = (A\mathbf{y})^T A\mathbf{x} = \mathbf{y}^T A^T A\mathbf{x}$ para todo $\mathbf{x} \in \mathcal{X}$.

A propriedade (ii) justifica a denominação *complemento oblíquo* e pode ser verificada como segue.

Seja $X \in \mathbb{R}^{n \times p}$ com posto coluna completo tal que $\mathcal{R}(X) = \mathcal{X}$. Vamos mostrar primeiramente que $\dim \mathcal{R}(AX) = p$.

De fato, se $X = [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_p]$ considere a igualdade

$$\alpha_1 A\mathbf{x}_1 + \alpha_2 A\mathbf{x}_2 + \dots + \alpha_p A\mathbf{x}_p = \mathbf{0},$$

que pode ser reescrita como:

$$A(\alpha_1 \mathbf{x}_1 + \alpha_2 \mathbf{x}_2 + \dots + \alpha_p \mathbf{x}_p) = \mathbf{0}.$$

Assim, se $\boldsymbol{\alpha} = [\alpha_1 \ \alpha_2 \ \dots \ \alpha_p]^T$, temos $X\boldsymbol{\alpha} \in \mathcal{N}(A)$. Mas, $X\boldsymbol{\alpha} \in \mathcal{X}$.

Como $\mathcal{N}(A) \cap \mathcal{X} = \{\mathbf{0}\}$ segue que $X\boldsymbol{\alpha} = \mathbf{0}$. Do fato de X tem colunas linearmente independentes temos que $\boldsymbol{\alpha} = \mathbf{0}$.

Agora, como $\text{posto}(A) = r$ e $\mathcal{R}(AX) \subseteq \mathcal{R}(A)$, existem vetores $\mathbf{v}_{p+1}, \mathbf{v}_{p+2}, \dots, \mathbf{v}_r$ de \mathbb{R}^m linearmente independentes tais que, para cada $j = p+1, p+2, \dots, r$

$$\mathbf{v}_j \in \mathcal{R}(A), \quad \mathbf{v}_j \notin \mathcal{R}(AX) \quad \text{e} \quad \mathbf{v}_j \perp \mathcal{R}(AX). \quad (3.19)$$

Assim, podemos encontrar $\mathbf{y}_{p+1}, \mathbf{y}_{p+2}, \dots, \mathbf{y}_r$ de \mathbb{R}^n tais que $A\mathbf{y}_j = \mathbf{v}_j$, $j = p+1, p+2, \dots, r$. Em particular podemos tomar $\mathbf{y}_j = A^\dagger \mathbf{v}_j$ e, neste caso, $\mathbf{y}_j \in \mathcal{N}(A)^\perp$ pois

$$\mathbf{y}_j = A^\dagger \mathbf{v}_j = A^\dagger A A^\dagger \mathbf{v}_j = P_{\mathcal{R}(A^T)} A^\dagger \mathbf{v}_j$$

e assim $\mathbf{y}_j \in \mathcal{R}(A^T) = \mathcal{N}(A)^\perp$.

Da equação (3.19) temos

$$\mathbf{y}_j A^T A\mathbf{x} = \mathbf{0}, \quad j = p+1, p+2, \dots, r.$$

Isto é, $\mathbf{y}_j \in \mathcal{X}^{\perp A}$ e $\mathbf{y}_j \notin \mathcal{X} \forall j$, pois se o último ocorrer teríamos para algum $\mathbf{x} \in \mathbb{R}^p$

$$\mathbf{y}_j = X\mathbf{x} \Rightarrow \mathbf{v}_j = AX\mathbf{x} \Rightarrow \mathbf{v}_j \in \mathcal{R}(AX).$$

Vamos verificar que os vetores $\mathbf{y}_{p+1}, \mathbf{y}_{p+2}, \dots, \mathbf{y}_r$ formam um conjunto linearmente independente. De fato, multiplicando por A a igualdade

$$\alpha_{p+1}\mathbf{y}_{p+1} + \alpha_{p+2}\mathbf{y}_{p+2} + \dots + \alpha_r\mathbf{y}_r = \mathbf{0}$$

temos

$$\alpha_{p+1}A\mathbf{y}_{p+1} + \alpha_{p+2}A\mathbf{y}_{p+2} + \dots + \alpha_rA\mathbf{y}_r = \mathbf{0}.$$

Usando o fato que $A\mathbf{y}_j = \mathbf{v}_j$ para $j = p+1, \dots, r$, resulta

$$\alpha_{p+1}\mathbf{v}_{p+1} + \alpha_{p+2}\mathbf{v}_{p+2} + \dots + \alpha_r\mathbf{v}_r = \mathbf{0},$$

o que implica

$$\alpha_{p+1}, \alpha_{p+2}, \dots, \alpha_r = 0$$

pois os \mathbf{v}_j formam um conjunto linearmente independente.

Portanto, temos \mathcal{X} de dimensão p ,

$$\dim \mathcal{N}(A) = n - r \quad \text{e} \quad \mathcal{N}(A) \subset \mathcal{X}^{\perp A}$$

de (i) e para $j = p+1, p+2, \dots, r$

$$\mathbf{y}_j \in \mathcal{N}(A)^\perp \quad \text{e} \quad \mathbf{y}_j \in \mathcal{X}^{\perp A}.$$

Daí, $\dim \mathcal{X}^{\perp A} \geq n - r + (r - p) = n - p$. Como $\mathcal{N}(A) \cap \mathcal{X} = \{\mathbf{0}\}$ e $\mathbf{y}_j \notin \mathcal{X}$, segue que

$$\mathbb{R}^n = \mathcal{X} + \mathcal{X}^{\perp A}.$$

De agora em diante, assume-se que $\mathcal{N}(A) \cap \mathcal{X} = \{\mathbf{0}\}$.

Quanto aos projetores denotamos $E_{\mathcal{X}} = E_{\mathcal{X}, \mathcal{X}^{\perp A}}$.

Teorema 3.3. *Seja $A \in \mathbb{R}^{m \times n}$ e $\mathcal{X} = \mathcal{R}(X)$, com $X \in \mathbb{R}^{n \times p}$ e $p = \text{posto}(X)$. Então a representação matricial de $E_{\mathcal{X}}$ é*

$$E_{\mathcal{X}} = X (AX)^{\dagger} A, \quad (3.20)$$

e para todo $\mathbf{x} \in \mathbb{R}^n$ temos uma decomposição $\mathbf{x} = E_{\mathcal{X}}\mathbf{x} + (I - E_{\mathcal{X}})\mathbf{x}$ tal que

$$E_{\mathcal{X}}\mathbf{x} \perp_A (I - E_{\mathcal{X}})\mathbf{x}. \quad (3.21)$$

Demonstração. Primeiramente, vamos mostrar que $E_{\mathcal{X}}$ é um projetor. De fato,

$$E_{\mathcal{X}}^2 = X (AX)^{\dagger} AX (AX)^{\dagger} A = X (AX)^{\dagger} A = E_{\mathcal{X}}, \quad (3.22)$$

e assim, $E_{\mathcal{X}}$ é idempotente. Agora vamos verificar as condições da Definição 3.1. Para (a), note que AX é de posto completo, portanto

$$E_{\mathcal{X}}X = X (AX)^{\dagger} AX = X.$$

Para (b), considere $\mathbf{y} \in \mathcal{X}^{\perp_A}$. Então, $\mathbf{x}^T A^T A \mathbf{y} = 0$, $\forall \mathbf{x} \in \mathcal{X}$. Assim

$$E_{\mathcal{X}}\mathbf{y} = X (AX)^{\dagger} A \mathbf{y} = X (X^T A^T A X)^{\dagger} X^T A^T A \mathbf{y} = \mathbf{0}.$$

pois $X^T A^T A \mathbf{y} = \mathbf{0}$. O item (c) pode ser verificado do fato que $\mathcal{R}(E_{\mathcal{X}}) \subseteq \mathcal{R}(X) = \mathcal{X}$. Para verificar (3.21) vamos verificar que os vetores $E_{\mathcal{X}}\mathbf{x}$ e $(I - E_{\mathcal{X}})\mathbf{x}$ satisfazem a definição 3.2. Note que

$$\begin{aligned} E_{\mathcal{X}}^T A^T A (I - E_{\mathcal{X}}) &= A^T \left[(AX)^{\dagger} \right]^T X^T A^T A \left(I - X (AX)^{\dagger} A \right) \\ &= A^T \left[(AX) (AX)^{\dagger} \right]^T A \left(I - X (AX)^{\dagger} A \right) \\ &= A^T AX (AX)^{\dagger} A - A^T AX (AX)^{\dagger} AX (AX)^{\dagger} A \\ &= A^T AX (AX)^{\dagger} A - A^T AX (AX)^{\dagger} A \\ &= 0. \end{aligned}$$

Assim, tomando $\mathbf{x} \in \mathbb{R}^n$ temos $\mathbf{x}^T E_{\mathcal{X}}^T A^T A (I - E_{\mathcal{X}})\mathbf{x} = 0$. ■

Definição 3.4. Dada uma matriz $X \in \mathbb{R}^{p \times n}$ com $p \leq n$ e um subespaço $\mathcal{Y} \subseteq \mathbb{R}^n$ tal que $\mathcal{N}(X)$ e \mathcal{Y} tem intersecção trivial, definimos a pseudo-inversa oblíqua $X_{\mathcal{Y}}^{\dagger}$ como a matriz $n \times p$

$$X_{\mathcal{Y}}^{\dagger} = E_{\mathcal{Y}, \mathcal{N}(X)} X^{\dagger}. \quad (3.23)$$

Para o caso em que $\mathcal{Y} = \mathcal{N}(X)^{\perp} = \mathcal{R}(X^T)$ temos

$$E_{\mathcal{N}(X)^{\perp}, \mathcal{N}(X)} = P_{\mathcal{N}(X)^{\perp}} = X^{\dagger} X \quad (3.24)$$

e assim,

$$X_{\mathcal{Y}}^{\dagger} = X^{\dagger} X X^{\dagger} = X^{\dagger}.$$

Para algumas manipulações teóricas, é importante conhecer a representação matricial de $X_{\mathcal{Y}}^{\dagger}$ com relação as matrizes que geram os espaços envolvidos. Sendo assim, vamos obter tais expressões.

Para dois espaços \mathcal{X} e \mathcal{Y} com intersecção trivial, temos $E_{\mathcal{X}, \mathcal{Y}} = X (Y_0^T X)^{\dagger} Y_0^T$, em que $\mathcal{R}(X) = \mathcal{X}$. Se Y_0 é como em (3.9) a expressão se mantém. Para o caso em que $Y_0 = P_{\mathcal{Y}^{\perp}} X$ podemos escrever

$$E_{\mathcal{X}, \mathcal{Y}} = X (X^T P_{\mathcal{Y}^{\perp}}^T X)^{\dagger} X^T P_{\mathcal{Y}^{\perp}}^T. \quad (3.25)$$

Para $E_{\mathcal{Y}, \mathcal{N}(X)}$ com $\mathcal{R}(Y) = \mathcal{Y}$, tomando $X_0 = P_{\mathcal{N}(X)^{\perp}} Y$ temos

$$\begin{aligned} E_{\mathcal{Y}, \mathcal{N}(X)} &= Y (X_0^T Y)^{\dagger} X_0^T \\ &= Y (Y^T P_{\mathcal{N}(X)^{\perp}}^T Y)^{\dagger} Y^T P_{\mathcal{N}(X)^{\perp}}^T. \end{aligned} \quad (3.26)$$

De $P_{\mathcal{N}(X)^{\perp}} = X^{\dagger} X$, podemos reescrever (3.26) como

$$E_{\mathcal{Y}, \mathcal{N}(X)} = Y (Y^T X^{\dagger} X Y)^{\dagger} Y^T X^{\dagger} X. \quad (3.27)$$

e assim

$$\begin{aligned} X_{\mathcal{Y}}^{\dagger} &= E_{\mathcal{Y}, \mathcal{N}(X)} X^{\dagger} \\ &= Y (Y^T X^{\dagger} X Y)^{\dagger} Y^T X^{\dagger} X X^{\dagger} \\ &= Y (Y^T X^{\dagger} X Y)^{\dagger} Y^T X^{\dagger} \end{aligned} \quad (3.28)$$

onde usamos a condição (ii) da Definição A.1.

No entanto, tomando X_0 tal que $\mathcal{R}(X_0) = \mathcal{N}(X)^\perp$ e do fato de $\mathcal{N}(X)^\perp = \mathcal{R}(X^T)$, temos

$$E_{\mathcal{Y}, \mathcal{N}(X)} = Y(XY)^\dagger X \quad (3.29)$$

e assim

$$X_{\mathcal{Y}}^\dagger = Y(XY)^\dagger XX^\dagger. \quad (3.30)$$

Para o caso em que X tem posto completo, o que implica $XX^\dagger = I_p$, temos

$$X_{\mathcal{Y}}^\dagger = Y(XY)^\dagger. \quad (3.31)$$

Teorema 3.4. *Sejam X e \mathcal{Y} como na definição 3.4 e $\mathcal{R}(Y) = \mathcal{Y}$. Assuma $\text{posto}(X) = p$ e $Y \in \mathbb{R}^{n \times q}$ com $\text{posto}(Y) = q$. A pseudo-inversa oblíqua $X_{\mathcal{Y}}^\dagger$ satisfaz as seguintes relações*

(a) $XX_{\mathcal{Y}}^\dagger = P_{\mathcal{R}(XY)}$

(b) $X_{\mathcal{Y}}^\dagger X = E_{\mathcal{Y}, \mathcal{N}(X)}$

(c) $X^\dagger = P_{\mathcal{R}(X^T)} X_{\mathcal{Y}}^\dagger$

Demonstração. Vamos usar $E_{\mathcal{Y}, \mathcal{N}(X)}$ como na equação (3.29) e, conseqüentemente, $X_{\mathcal{Y}}^\dagger$ como na equação (3.30). Como X tem posto linha completo segue que $XX^\dagger = I_p$.

O item (a) segue de

$$XX_{\mathcal{Y}}^\dagger = XY(XY)^\dagger XX^\dagger = XY(XY)^\dagger = P_{\mathcal{R}(XY)}.$$

O item (b) segue de

$$X_{\mathcal{Y}}^\dagger X = Y(XY)^\dagger XX^\dagger X = Y(XY)^\dagger X = E_{\mathcal{Y}, \mathcal{N}(X)} \quad (3.32)$$

Para (c) usamos o item (a) pois

$$P_{\mathcal{R}(X^T)} X_{\mathcal{Y}}^\dagger = X^\dagger XX_{\mathcal{Y}}^\dagger = X^\dagger P_X = X^\dagger XX^\dagger = X^\dagger.$$

■

Em geral, como $\dim \mathcal{N}(X) = n - p$ e $\mathcal{R}(Y), \mathcal{N}(X) \subset \mathbb{R}^n$ então $q \leq p$ pois $q + (n - p) \leq n$. Para o caso em que $p = q$, que significa que $\mathcal{Y} + \mathcal{N}(X) = \mathbb{R}^n$, as condições (b) e (c) do teorema 3.4 continuam válidas e a propriedade (a) pode ser escrita com $XX_{\mathcal{Y}}^{\dagger} = P_{\mathcal{R}(X)}$. De fato, se $X \in \mathbb{R}^{p \times n}$ e $Y \in \mathbb{R}^{n \times p}$ então

$$\text{posto}(XY) = \text{posto}(Y) - \dim(\mathcal{R}(X) \cap \mathcal{R}(Y)) = \text{posto}(Y)$$

como verificado em [37]. Também, $\mathcal{R}(XY) \subset \mathcal{R}(X)$ e como X e Y tem posto p então $\dim \mathcal{R}(XY) = \dim \mathcal{R}(X)$. Logo $\mathcal{R}(XY) = \mathcal{R}(X)$.

Observação 3.2. Em [28] e [30], o autor apresenta o teorema 3.4 com a condição (a) sendo $XX_{\mathcal{Y}}^{\dagger} = P_{\mathcal{R}(X)}$ para quaisquer p e q . O caso $p = q$, como visto acima, é válido no entanto, para $q < p$, não é verdade que $XX_{\mathcal{Y}}^{\dagger} = P_{\mathcal{R}(X)}$. De fato, note que $XX^{\dagger} = XY(XY)^{\dagger}$. Como Y tem posto q menor que p , é imediato que

$$\text{posto}(XY(XY)^{\dagger}) \leq q.$$

Se $XX^{\dagger} = P_{\mathcal{R}(X)}$ então teríamos que $P_{\mathcal{R}(X)}$ tem posto, no máximo, q . Isso não pode ocorrer pois $\text{posto}(P_{\mathcal{R}(X)}) = \text{posto}(\mathcal{R}(X)) = p$.

Como será visto na seção 3.2, a transformação para a forma padrão usa somente o caso $p = q$.

Quando $X \in \mathbb{R}^{p \times n}$ com $p \geq n$ temos

$$X_{\mathcal{Y}}^{\dagger} = X^{\dagger} E_{\mathcal{X}, \mathcal{Y}} \quad (3.33)$$

para $\mathcal{X} = \mathcal{R}(X)$ e \mathcal{Y} com intersecção trivial. Neste caso é válido o teorema a seguir, similar ao teorema 3.4.

Teorema 3.5. Seja $X_{\mathcal{Y}}^{\dagger}$ como na equação (3.33) então

(a) $XX_{\mathcal{Y}}^{\dagger} = E_{\mathcal{X}, \mathcal{Y}}$

(b) $X^{\dagger} = X_{\mathcal{Y}}^{\dagger} P_{\mathcal{X}}$

(c) $X_{\mathcal{Y}}^{\dagger} X = P_{\mathcal{R}(X^T)}$

A demonstração do Teorema 3.5 é análoga a do Teorema 3.4.

Observação 3.3. Dadas matrizes $A \in \mathbb{R}^{m \times n}$ e $X \in \mathbb{R}^{p \times n}$, se $\mathcal{Y} = \mathcal{N}(X)^{\perp A}$ então a pseudo-inversa oblíqua $X_{\mathcal{N}(X)^{\perp A}}^{\dagger}$ é dada por

$$X_{\mathcal{N}(X)^{\perp A}}^{\dagger} = \left(I_n - \left(A(I_n - X^{\dagger}X) \right)^{\dagger} A \right) X^{\dagger} \quad (3.34)$$

De fato, se temos uma matriz W tal que $\mathcal{R}(W) = \mathcal{N}(X)$, então

$$\begin{aligned} X_{\mathcal{N}(X)^{\perp A}}^{\dagger} &= E_{\mathcal{N}(X)^{\perp A}, \mathcal{N}(X)} X^{\dagger} = \left(I_n - E_{\mathcal{N}(X), \mathcal{N}(X)^{\perp A}} \right) X^{\dagger} = \\ &= \left(I_n - E_{\mathcal{N}(X)} \right) X^{\dagger} = \left(I_n - W(AW)^{\dagger} A \right) X^{\dagger} = \\ &= \left(I_n - \left(AWW^{\dagger} \right)^{\dagger} A \right) X^{\dagger} \\ &= \left(I_n - \left(AP_{\mathcal{N}(X)} \right)^{\dagger} A \right) X^{\dagger} \\ &= \left(I_n - \left(A(I_n - X^{\dagger}X) \right)^{\dagger} A \right) X^{\dagger}. \end{aligned}$$

3.2 Transformação do Problema de Regularização de Tikhonov na Forma Geral para a Forma Padrão

Conhecendo nossas ferramentas de trabalho, derivamos a transformação sugerida no início deste capítulo e apresentamos métodos para o cálculo da matriz \bar{A} e dos vetores $\bar{\mathbf{b}}$ e \mathbf{y}_0 a partir dos dados originais A , L , \mathbf{b} e \mathbf{x}_0 .

Assuma que $\mathcal{N}(A) \cap \mathcal{N}(L) = \{\mathbf{0}\}$ e que $L \in \mathbb{R}^{p \times n}$ tem posto completo, isto é, $\text{posto}(L) = \min\{p, n\}$. A partir destas restrições trabalhamos com três casos para a matriz L , variando a relação entre p e n .

No caso mais simples tem-se $p = n$ e, tendo L posto completo, sabemos que L^{-1} existe. Neste caso, obtemos $\bar{A} = AL^{-1}$, $\bar{\mathbf{b}} = \mathbf{b}$ e $\mathbf{y}_0 = L\mathbf{x}_0$ fazendo $\mathbf{y} = L\mathbf{x}$ e usando a não-singularidade de L .

Então, a transformação reversa torna-se $\mathbf{x} = L^{-1}\mathbf{y}$. Porém, em aplicações, o mais comum é uma matriz L não quadrada.

Para $p > n$, temos que a matriz $L^T L$ é inversível e, neste caso, consideramos a pseudo inversa de Moore Penrose de L dada por

$$L^\dagger = (L^T L)^{-1} L^T, \quad (3.35)$$

ou seja, $L^\dagger L = I_n$ e L tem inversa à esquerda. Semelhante ao que foi feito anteriormente, tomamos $\mathbf{y} = L\mathbf{x}$ e, então, $\bar{\mathbf{A}} = AL^\dagger$, $\bar{\mathbf{b}} = \mathbf{b}$ e $\mathbf{y}_0 = L\mathbf{x}_0$ substituindo $\mathbf{x} = L^\dagger \mathbf{y}$ em $\|\bar{\mathbf{b}} - \bar{\mathbf{A}}\mathbf{x}\|^2$.

Uma abordagem matematicamente equivalente e que é mais eficiente é considerar a fatoração QR de L . Seja

$$L = QR = [Q_1 \quad Q_2] \begin{bmatrix} R_1 \\ O \end{bmatrix} = Q_1 R_1 \quad (3.36)$$

onde $Q_1 \in \mathbb{R}^{p \times n}$ tem colunas ortonormais e $R \in \mathbb{R}^{n \times n}$ é triangular superior e não singular. Então,

$$\|L(\mathbf{x} - \mathbf{x}_0)\|_2 = \|Q_1 R_1(\mathbf{x} - \mathbf{x}_0)\|_2 = \|R_1(\mathbf{x} - \mathbf{x}_0)\|_2. \quad (3.37)$$

Desta forma, voltamos ao caso anterior e a transformação neste caso resulta

$$\bar{\mathbf{A}} = AR_1^{-1}, \quad \bar{\mathbf{b}} = \mathbf{b} \quad \text{e} \quad \mathbf{y}_0 = R_1 \mathbf{x}_0 \quad (3.38)$$

e a transformação reversa é $\mathbf{x}_\lambda = R_1^{-1} \mathbf{y}_\lambda$. Em termos de Q e R temos $L^\dagger = R_1^{-1} Q_1^T$.

De agora em diante, vamos considerar $\mathbf{x}_0 = 0$, visto que sempre é possível fazer uma mudança de variável $\hat{\mathbf{x}} \equiv \mathbf{x} - \mathbf{x}_0$ no regularização de Tikhonov na forma geral 3.1, resolver

$$\hat{\mathbf{x}}_\lambda = \operatorname{argmin}_{\hat{\mathbf{x}} \in \mathbb{R}^n} \left\{ \|(\mathbf{b} + A\mathbf{x}_0) - A\hat{\mathbf{x}}\|_2^2 + \lambda^2 \|L\hat{\mathbf{x}}\|_2^2 \right\} \quad (3.39)$$

e, em seguida, obter $\mathbf{x}_\lambda = \hat{\mathbf{x}}_\lambda + \mathbf{x}_0$.

Por fim, consideremos L com mais colunas do que linhas, o que implica que $\mathcal{N}(L)$ é não trivial. Neste caso, a transformação para a forma padrão torna-se um pouco mais complexa do que nos casos anteriores, sendo esta a situação mais frequente.

Quando isto ocorre, precisamos introduzir uma decomposição da solução de tal forma que uma das componentes pertença a $\mathcal{N}(L)$. Isto é, dado $\mathbf{x} \in \mathbb{R}^n$, escrevemos

$$\mathbf{x} = \mathbf{x}_{\mathcal{M}} + \mathbf{x}_{\mathcal{N}} \quad (3.40)$$

com $\mathbf{x}_{\mathcal{N}} \in \mathcal{N}(L)$.

Usando (3.40), o problema de minimização do funcional de Tikhonov na forma geral torna-se

$$\min \left\{ \|\mathbf{b} - A\mathbf{x}_{\mathcal{M}} - A\mathbf{x}_{\mathcal{N}}\|_2^2 + \lambda^2 \|L\mathbf{x}_{\mathcal{M}}\|_2^2 \right\}, \quad (3.41)$$

Note que

$$\begin{aligned} & \|\mathbf{b} - A\mathbf{x}_{\mathcal{M}} - A\mathbf{x}_{\mathcal{N}}\|_2^2 = \\ & = \|\mathbf{b} - A\mathbf{x}_{\mathcal{M}}\|_2^2 + \|\mathbf{b} - A\mathbf{x}_{\mathcal{N}}\|_2^2 - \mathbf{b}^T \mathbf{b} + 2(A\mathbf{x}_{\mathcal{M}})^T A\mathbf{x}_{\mathcal{N}}. \end{aligned}$$

Se exigimos que $A\mathbf{x}_{\mathcal{M}}$ seja ortogonal a $A\mathbf{x}_{\mathcal{N}}$, isto é, $\mathbf{x}_{\mathcal{M}}^T A^T A\mathbf{x}_{\mathcal{N}} = 0$, então obtemos dois problemas de mínimos quadrados independentes relacionados a cada um dos componentes de \mathbf{x} dados em (3.40)

$$\min \left\{ \|\mathbf{b} - A\mathbf{x}_{\mathcal{M}}\|_2^2 + \lambda^2 \|L\mathbf{x}_{\mathcal{M}}\|_2^2 \right\} \quad \text{e} \quad \min \left\{ \|\mathbf{b} - A\mathbf{x}_{\mathcal{N}}\|_2^2 \right\}$$

Desta forma, a exigência que fazemos é $\mathbf{x}_{\mathcal{M}} \perp_A \mathbf{x}_{\mathcal{N}}$, o que sugere $\mathbb{R}^n = \mathcal{N}(L) + \mathcal{N}(L)^{\perp_A}$, ao invés da ortogonalidade usual. Note também que o termo $-\mathbf{b}^T \mathbf{b}$, por ser uma constante, não interfere na minimização de (3.41).

Esta decomposição do \mathbb{R}^n é o que nos conduz ao uso da pseudo-inversa oblíqua para a transformação da regularização de Tikhonov da forma geral para a forma padrão.

Conforme visto na seção anterior, os projetores oblíquos relacionados são

$$\begin{aligned} E_{\mathcal{N}} &= E_{\mathcal{N}(L), \mathcal{N}(L)^{\perp_A}} = E_{\mathcal{N}(L)} \\ E_{\mathcal{M}} &= E_{\mathcal{N}(L)^{\perp_A}, \mathcal{N}(L)} = I - E_{\mathcal{N}(L), \mathcal{N}(L)^{\perp_A}} = I - E_{\mathcal{N}} \end{aligned}$$

com a propriedade de que $E_{\mathcal{M}} + E_{\mathcal{N}} = I$, ou seja,

$$\mathbf{x} = \mathbf{x}_{\mathcal{M}} + \mathbf{x}_{\mathcal{N}} = E_{\mathcal{M}}\mathbf{x} + E_{\mathcal{N}}\mathbf{x}. \quad (3.42)$$

Nosso objetivo, agora, é resolver os seguintes subproblemas:

(i) Para $\mathbf{x}_M = E_M \mathbf{x}$, resolver um problema na forma padrão:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \{ \|\mathbf{b} - AE_M \mathbf{x}\|_2^2 + \lambda^2 \|LE_M \mathbf{x}\|_2^2 \} \quad (3.43)$$

(ii) Para $\mathbf{x}_N = E_N \mathbf{x}$, resolver um problema de mínimos quadrados usual:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \{ \|\mathbf{b} - AE_N \mathbf{x}\|_2^2 \} \quad (3.44)$$

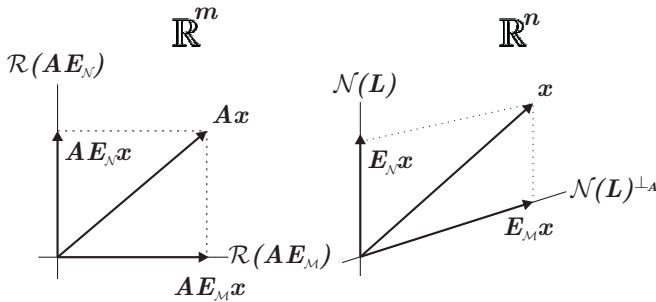


Figura 3.2: Ilustração da decomposição oblíqua $\mathbb{R}^n = \mathcal{N}(L) + \mathcal{N}(L)^{\perp A}$ para a transformação para a forma padrão.

Com a decomposição do \mathbb{R}^n foi possível separar o problema original (3.1) em dois problemas de mínimos quadrados independentes, através da separação ortogonal $\mathbb{R}^m = \mathcal{R}(AE_M) + \mathcal{R}(AE_N)$. A Figura 3.2 ilustra a decomposição proposta. Neste caso, decomparamos o \mathbb{R}^m como $\mathbb{R}^m = \mathcal{R}(AE_M) + \mathcal{R}(AE_N)$.

A solução do problema (i) nos fornece a componente da solução regularizada de (3.41) que está em $\mathcal{N}(L)^{\perp A}$, denotada por $\mathbf{x}_{\lambda, M}$. Desta forma,

$$\mathbf{x}_\lambda = \mathbf{x}_{\lambda, M} + \mathbf{x}_N \quad (3.45)$$

onde \mathbf{x}_N é a componente da solução regularizada que pertence ao espaço nulo de L . Vale ressaltar que encontrar \mathbf{x}_N é um problema que não depende do λ escolhido.

Para resolver o problema (ii) vamos considerar $W \in \mathbb{R}^{n \times (n-p)}$ tal que $\mathcal{R}(W) = \mathcal{N}(L)$. Assim, existe $\mathbf{z}_N \in \mathbb{R}^{n-p}$ tal que

$Wz_{\mathcal{N}} = \mathbf{x}_{\mathcal{N}}$. Isso produz o seguinte problema de mínimos quadrados

$$\min_{z \in \mathbb{R}^{n-p}} \| \mathbf{b} - AWz \|_2^2 \quad (3.46)$$

cuja solução de mínima norma é dada por $z_{\mathcal{N}} = (AW)^\dagger \mathbf{b}$ e, então, $\mathbf{x}_{\mathcal{N}} = W(AW)^\dagger \mathbf{b}$. Note que, do Teorema 3.4, segue que $W(AW)^\dagger = A_{\mathcal{N}(L)}^\dagger$.

Para o problema de mínimos quadrados (i), segue dos teoremas 3.1 e 3.4 que

$$E_{\mathcal{M}} = L_{\mathcal{N}(L)^\perp A}^\dagger L \quad \text{e} \quad E_{\mathcal{N}} = W(AW)^\dagger A. \quad (3.47)$$

Assim, temos uma representação matricial para a pseudo-inversa oblíqua de L , que é dada por

$$L_{\mathcal{N}(L)^\perp A}^\dagger = E_{\mathcal{M}} L^\dagger = \left(I - W(AW)^\dagger A \right) L^\dagger. \quad (3.48)$$

Usando diretamente a matriz L e visto que

$$\left(A(I_n - L^\dagger L) \right)^\dagger = \left(AP_{\mathcal{N}(L)} \right)^\dagger = \left(AWW^\dagger \right)^\dagger = W(AW)^\dagger$$

temos que

$$L_{\mathcal{N}(L)^\perp A}^\dagger \equiv \left(I_n - \left(A(I_n - L^\dagger L) \right)^\dagger A \right) L^\dagger \quad (3.49)$$

$$\mathbf{x}_{\mathcal{N}} \equiv \left(A(I_n - L^\dagger L) \right)^\dagger \mathbf{b}. \quad (3.50)$$

A pseudo-inversa oblíqua $L_{\mathcal{N}(L)^\perp A}^\dagger$ é também chamada de pseudo-inversa com peso- A . Por simplicidade, ela é denotada por L_A^\dagger . Note também que quando $L \in \mathbb{R}^{p \times n}$ com $\text{posto}(L) = n < p$ então $L^\dagger L = I_n$ e $L_A^\dagger = L^\dagger$, e reduz-se a L^{-1} quando L é quadrada e inversível.

Considere, agora, o problema (3.43). Então

$$AE_{\mathcal{M}} = \left(AL_A^\dagger \right) L \quad (3.51)$$

$$LE_{\mathcal{M}} = LL_A^\dagger L = P_{\mathcal{R}(L)} L = L. \quad (3.52)$$

Substituindo em (3.41) temos

$$\begin{aligned}
& \min \left\{ \|\mathbf{b} - A(E_{\mathcal{M}}\mathbf{x} + \mathbf{x}_{\mathcal{N}})\|_2^2 + \lambda^2 \|L(E_{\mathcal{M}}\mathbf{x} + \mathbf{x}_{\mathcal{N}})\|_2^2 \right\} \\
&= \min \left\{ \|\mathbf{b} - AL_A^\dagger L\mathbf{x} - A\mathbf{x}_{\mathcal{N}}\|_2^2 + \lambda^2 \|LE_{\mathcal{M}}\mathbf{x}\|_2^2 \right\} \\
&= \min \left\{ \|(\mathbf{b} - A\mathbf{x}_{\mathcal{N}}) - AL_A^\dagger L\mathbf{x}\|_2^2 + \lambda^2 \|L\mathbf{x}\|_2^2 \right\}.
\end{aligned}$$

Definindo $\mathbf{y} = L\mathbf{x}$ temos $\bar{A} \equiv AL_A^\dagger$ e $\bar{\mathbf{b}} = \mathbf{b} - A\mathbf{x}_{\mathcal{N}}$ e obtemos a regularização de Tikhonov na forma padrão

$$\mathbf{y}_\lambda = \operatorname{argmin}_{\mathbf{y} \in \mathbb{R}^p} \left\{ \|\bar{\mathbf{b}} - \bar{A}\mathbf{y}\|_2^2 + \lambda^2 \|\mathbf{y}\|_2^2 \right\} \quad (3.53)$$

cuja transformação reversa é $\mathbf{x}_{\lambda, \mathcal{M}} = L_A^\dagger \mathbf{y}_\lambda$ pois

$$L_A^\dagger \mathbf{y} = L_A^\dagger L\mathbf{x} = E_{\mathcal{M}}\mathbf{x} = \mathbf{x}_{\mathcal{M}}. \quad (3.54)$$

Portanto, a solução regularizada do problema de Tikhonov na forma geral é

$$\mathbf{x}_\lambda = L_A^\dagger \mathbf{y}_\lambda + \mathbf{x}_{\mathcal{N}} = \left(I - W(AW)^\dagger A \right) L^\dagger \mathbf{y}_\lambda + W(AW)^\dagger \mathbf{b}.$$

Vale ressaltar que esta transformação requer métodos numericamente estáveis para o cálculo da matriz \bar{A} , do lado direito $\bar{\mathbf{b}}$ e do vetor \mathbf{y}_0 a partir das quantidades A , L , \mathbf{b} e \mathbf{x}_0 .

3.3 Transformação e a GSVD

Juntamente com as equações (4.17) e (3.50), considere a GSVD de (A, L) particionada como

$$A = [U_1 \ U_2] \begin{bmatrix} \Sigma & 0 \\ 0 & I_{n-p} \end{bmatrix} [X_1 \ X_2]^{-1} \quad (3.55)$$

$$L = V[M, 0] [X_1 \ X_2]^{-1}. \quad (3.56)$$

onde U_1 e X_1 tem p colunas, e Σ e M são matrizes diagonais $p \times p$.

Então podemos obter a pseudo-inversa oblíqua L_A^\dagger e \mathbf{x}_N como

$$L_A^\dagger = X \begin{bmatrix} M^{-1} \\ 0 \end{bmatrix} V^T = X_1 M^{-1} V^T \quad \text{e} \quad \mathbf{x}_N = \sum_{i=p+1}^n \mathbf{u}_i^T \mathbf{b} \mathbf{x}_i$$

o que nos fornece outra expressão para a pseudo-inversa oblíqua. Assim, \overline{A} pode ser expressa como

$$AL_A^\dagger = U_1 \Sigma M^{-1} V^T = \sum_{i=1}^p \gamma_i \mathbf{u}_i \mathbf{v}_i^T, \quad (3.57)$$

onde U_p representa as p primeiras colunas de U , embora essa abordagem não seja muito útil para problemas de grande porte, tendo em vista a dificuldade em calcular a GSVD neste caso.

A expressão acima mostra que os valores singulares generalizados γ_i são os valores singulares da matriz \overline{A} . Outra vantagem da transformação para a forma padrão é que a relação $\mathbf{y} = L\mathbf{x}$ e $\mathbf{b} - A\mathbf{x} = \overline{\mathbf{b}} - \overline{A}\mathbf{y}$ leva imediatamente as igualdades $\|L\mathbf{x}\|_2 = \|\mathbf{y}\|_2$ e $\|\mathbf{b} - A\mathbf{x}\|_2 = \|\overline{\mathbf{b}} - \overline{A}\mathbf{y}\|_2$, visto que $LL_A^\dagger = I_p$.

Para problemas de grande porte a transformação para a forma padrão pode ser inviável do ponto de vista computacional uma vez que necessita da GSVD do par matricial (A, L) .

3.4 Matrizes Estruturadas

Já discutimos anteriormente as dificuldades em trabalhar com a matriz $\overline{A} = AL_A^\dagger$, principalmente quando trabalhamos com matrizes com dimensões grandes e quando L é uma matriz cheia, devido ao alto número de operações necessárias para realizar produtos matriz-vetor e matriz-matriz. Como, em muitos casos, a matriz L apresenta alguma estrutura especial, vamos falar brevemente sobre este tópico. Voltaremos a comentar sobre os aspectos da matriz L nos capítulos seguintes, ao apresentar os métodos de regularização aqui propostos e seus resultados.

Exemplos de estruturas bastante frequentes e que facilitam os cálculos numéricos são dados a seguir.

Definição 3.5 (Matriz de Toeplitz ou matriz de diagonais constantes). *É uma matriz em que cada diagonal paralela a diagonal principal, isto é, descendente da esquerda para a direita, tem valor constante. Uma matriz de Toeplitz é da forma*

$$\begin{bmatrix} a_0 & a_{-1} & a_{-2} & \cdots & a_{-(n-2)} & a_{-(n-1)} \\ a_1 & a_0 & a_{-1} & \ddots & & a_{-(n-2)} \\ a_2 & a_1 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & a_{-1} & a_{-2} \\ a_{n-2} & & \ddots & a_1 & a_0 & a_{-1} \\ a_{n-1} & a_{n-2} & \cdots & a_2 & a_1 & a_0 \end{bmatrix} \quad (3.58)$$

onde $a_i \in \mathbb{R}$.

Definição 3.6 (Matriz de Hankel). *É uma matriz em que cada diagonal paralela a diagonal secundária, isto é, descendente da direita para a esquerda, tem valor constante. A matriz de Hankel está intimamente relacionada com a matriz de Toeplitz e é da forma*

$$\begin{bmatrix} a_{-(n-1)} & a_{-(n-2)} & \cdots & a_{-2} & a_{-1} & a_0 \\ a_{-(n-2)} & & \ddots & a_{-1} & a_0 & a_1 \\ & & \ddots & \ddots & a_1 & a_2 \\ a_{-2} & a_{-1} & \ddots & \ddots & \ddots & \vdots \\ a_{-1} & a_0 & a_1 & \ddots & & a_{n-2} \\ a_0 & a_1 & a_2 & \cdots & a_{n-2} & a_{n-1} \end{bmatrix} \quad (3.59)$$

Definição 3.7 (Matriz de Hessenberg). *É uma matriz “quase” triangular, exceto por uma diagonal acima ou abaixo da diagonal principal. Uma matriz de Hessenberg pode ser inferior ou superior, sendo uma Hessenberg superior da seguinte forma*

$$\begin{bmatrix} a_0 & a_{-1} & a_{-2} & \cdots & \cdots & a_{-(n-1)} \\ a_1 & a_0 & a_{-1} & \ddots & & \vdots \\ 0 & a_1 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & a_{-1} & a_{-2} \\ \vdots & & \ddots & a_1 & a_0 & a_{-1} \\ 0 & \cdots & \cdots & 0 & a_1 & a_0 \end{bmatrix} \quad (3.60)$$

Definição 3.8 (Matriz banda). *É uma matriz onde os valores não nulos estão nas diagonais paralelas a diagonal principal e mais próximas dela. Isto é, uma matriz $A = (a_{i,j}) \in \mathbb{R}^{n \times n}$ é uma matriz banda se existem naturais $k_1, k_2 \in [0, n - 1]$ tais que*

$$a_{i,j} = 0 \quad \text{se} \quad j < i - k_1 \quad \text{ou} \quad j > i + k_2, \quad k_1, k_2 \geq 0.$$

Os valores k_1 e k_2 determinam a largura da banda acima e abaixo da diagonal principal, respectivamente. A largura de banda de uma matriz é $k_1 + k_2 + 1$, e pode-se definir como o número menor de diagonais adjacentes com valores não-nulos.

- *Uma matriz banda com $k_1 = k_2 = 0$ é uma matriz diagonal;*
- *Uma matriz banda com $k_1 = k_2 = 1$ é uma matriz tridiagonal, se $k_1 = k_2 = 2$ a matriz é dita pentadiagonal e assim sucessivamente;*
- *Uma matriz com $k_1 = 0$ e $k_2 = n - 1$, obtém-se a definição de uma matriz triangular inferior e, analogamente, se $k_1 = n - 1$ e $k_2 = 0$ obtém-se a definição de uma matriz triangular superior;*
- *Uma matriz banda com $k_1 = k_2 = p$ chama-se matriz p-banda e formalmente pode-se definir como*

$$a_{i,j} = 0 \quad \text{se} \quad |i - j| > p \quad ; \quad p \geq 0. \quad (3.61)$$

Uma matriz p-banda tem a forma

$$\begin{bmatrix} a_{11} & \cdots & a_{1(p+1)} & 0 & \cdots & 0 \\ \vdots & a_{22} & & a_{2(p+2)} & \ddots & \vdots \\ a_{(p+1)1} & & \ddots & & \ddots & 0 \\ 0 & a_{(p+2)2} & & \ddots & & a_{(n-p)n} \\ \vdots & \ddots & \ddots & & a_{(n-1)(n-1)} & \vdots \\ 0 & \cdots & 0 & a_{n(n-p)} & \cdots & a_{nn} \end{bmatrix}$$

Eventualmente, a matriz A pode também apresentar características especiais, como estrutura e esparsidade, o que deve ser levado em conta no momento de efetuar os produtos que envolvam a matriz ou sua transposta.

Definição 3.9 (Produto de Kronecker ou Produto Tensorial). Dadas matrizes $A = (a_{i,j})_{m \times n}$ e $B = (b_{i,j})_{p \times q}$ o produto de Kronecker de A por B é dado por

$$A \otimes B \equiv \begin{bmatrix} a_{1,1}B & a_{1,2}B & \dots & a_{1,n}B \\ a_{2,1}B & a_{2,2}B & & a_{2,n}B \\ \vdots & & \ddots & \vdots \\ a_{m,1}B & a_{m,2}B & \dots & a_{m,n}B \end{bmatrix} \in \mathbb{R}^{(mp) \times (nq)}$$

Propriedade 3.1. Se $A, \tilde{A} \in \mathbb{R}^{m \times n}$, $B, \tilde{B} \in \mathbb{R}^{p \times q}$, C, D matrizes reais de qualquer ordem e $\alpha \in \mathbb{R}$ então

- a) $(A \otimes B)^T = A^T \otimes B^T$;
- b) $(A + \tilde{A}) \otimes B = (A \otimes B) + (\tilde{A} \otimes B)$;
- c) $A \otimes (B + \tilde{B}) = (A \otimes B) + (A \otimes \tilde{B})$;
- d) $(\alpha A) \otimes B = A \otimes (\alpha B) = \alpha (A \otimes B)$;
- e) $(A \otimes B) \otimes C = A \otimes (B \otimes C)$;
- f) $(A \otimes B)(C \otimes D) = (AC \otimes BD)$.

Definição 3.10. Seja $V \in \mathbb{R}^{m \times n}$. Definimos $\text{vec}(V)$ como o vetor em \mathbb{R}^{mn} , obtido concatenando as colunas da matriz. Isto é,

$$\text{vec}(V)^T = [\mathbf{v}_1 \dots \mathbf{v}_n] \quad (3.62)$$

em que $\mathbf{v}_i \in \mathbb{R}^m$ é a i -ésima coluna de V .

Propriedade 3.2. Sejam $A \in \mathbb{R}^{m \times n}$, $X \in \mathbb{R}^{n \times p}$ e $B \in \mathbb{R}^{q \times p}$. Então

$$\text{vec}(AXB^T) = (B \otimes A) \text{vec}(X). \quad (3.63)$$

Neste trabalho, utilizamos as matrizes L_1 e L_{kron} dadas por

$$L_1 = \begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix} \in \mathbb{R}^{(n-1) \times n} \quad (3.64)$$

$$L_{\text{kron}} = \begin{bmatrix} I_n \otimes L_1 \\ L_1 \otimes I_n \end{bmatrix} \in \mathbb{R}^{2(n-1)n \times n^2}. \quad (3.65)$$

Em problemas envolvendo imagens, a matriz L_{kron} tem o papel de aplicar regularização tanto na vertical da imagens quanto na horizontal.

Para de efetuar o cálculo $L_{\text{kron}}\mathbf{x}$, $\mathbf{x} \in \mathbb{R}^{n^2}$, de forma eficiente, aproveitando a estrutura da matriz envolvida, contruímos rotinas auxiliares para desenvolver os produtos $(I_n \otimes L_1)\mathbf{x}$ e $(L_1 \otimes I_n)\mathbf{x}$. Isso foi bastante importante quando essas multiplicações matriz-vetor surgiam dentro de rotinas principais construídas para os problemas de regularização. Os produtos feitos de forma rápida, tornam a rotina principal mais eficiente. Os produtos são desenvolvidos como apresentados nas Tabelas 3.1 e 3.2.

Entrada: L_1 e $\mathbf{x} \in \mathbb{R}^{n^2}$.
Saída: Vetor $\mathbf{y} = (I_n \otimes L_1)\mathbf{x}$
1. Para i de 1 até n **faça**
 $\mathbf{y}((n-1)(i-1)+1 : (n-1)i, 1) = L_1 * \mathbf{x}(n(i-1)+1 : ni, 1)$;
Fim para

Tabela 3.1: Algoritmo *mkron1*.

Entrada: L_1 e $\mathbf{x} \in \mathbb{R}^{n^2}$.
Saída: Vetor $\mathbf{y} = (L_1 \otimes I_n)\mathbf{x}$
1. Para i de 1 até $(n-1)$ **faça**
 $\mathbf{z} = \text{zeros}(n, 1)$;
 Para j de 1 até n **faça**
 $\mathbf{a} = L_1(i, j) * \mathbf{x}(n(j-1)+1 : nj)$;
 $\mathbf{z} = \mathbf{z} + \mathbf{a}$;
 Fim Para
 $\mathbf{y}(n(i-1)+1 : ni) = \mathbf{z}$;
Fim Para

Tabela 3.2: Algoritmo *mkron2*.

Capítulo 4

Métodos de Projeção para Regularização com Informação a Priori

Nosso objetivo é estabelecer estratégias eficientes de se construir soluções estáveis para $\mathbf{x}_{\text{exato}} = A^\dagger \mathbf{b}_{\text{exato}}$ a partir do problema discreto mal posto

$$\mathbf{x} = \underset{\mathbf{x} \in \mathbb{R}^n}{\operatorname{argmin}} \|\mathbf{b} - A\mathbf{x}\|^2$$

baseadas nas ideias de métodos de projeção apresentadas anteriormente.

Os dois primeiros métodos propostos neste trabalho buscam resolver problemas de Tikhonov na forma geral, visto que a escolha $L = I$ nem sempre é satisfatória. Uma primeira abordagem sugere utilizar a transformação estudada no capítulo anterior, mas evitando o cálculo explícito da matriz \overline{A} , e resolver o problema transformado usando o método GKB-FP discutido anteriormente. Em seguida, apresentamos um algoritmo que utiliza o processo de bidiagonalização de Golub-Kahan e o método LSQR no problema da forma geral, seguindo as ideias apresentadas nas seções 2.2 e 2.3.

Lembrando que escolher λ pode ser uma tarefa difícil, su-

gerimos um método de regularização iterativa que não utiliza o parâmetro de regularização de Tikhonov. Essa proposta explora a transformação para a forma padrão e, portanto, a influência da matriz L . Por se tratar de um método iterativo, o desafio associado a este método está em escolher o critério de parada, obtendo o máximo de informação da solução desejada e impedindo a influência descontrolada do ruído.

4.1 GKB-FP para Tikhonov na Forma Geral

Motivados pelo bom desempenho do método GKB-FP ilustrado em [5], buscamos implementar este método para o problema de Tikhonov na forma geral, utilizando a transformação estudada na seção 3.2. Nos referimos a este método como GGKB-FP por trabalhar com o problema na forma geral.

Consideremos novamente as expressões obtidas para L_A^\dagger e $\mathbf{x}_\mathcal{N}$ anteriormente:

$$L_A^\dagger = \left(I_n - \left(A(I_n - L^\dagger L) \right)^\dagger A \right) L^\dagger \quad (4.1)$$

$$\mathbf{x}_\mathcal{N} = \left(A(I_n - L^\dagger L) \right)^\dagger \mathbf{b}. \quad (4.2)$$

Se conhecemos a matriz W tal que $\mathcal{R}(W) = \mathcal{N}(L)$ então podemos escrever

$$L_A^\dagger = \left(I - W(AW)^\dagger A \right) L^\dagger \quad (4.3)$$

$$\mathbf{x}_\mathcal{N} = W(AW)^\dagger \mathbf{b}. \quad (4.4)$$

Frequentemente, mesmo para L com p e n grandes, $\mathcal{N}(L)$ tem dimensão pequena, o que significa que W tem poucas colunas. Assim, o cálculo da componente da solução regularizada que pertence ao núcleo de L , dada por $\mathbf{x}_\mathcal{N}$, não apresenta grandes dificuldades. No entanto, para montar a matriz \bar{A} do problema transformado é necessário, primeiramente, calcular a matriz L_A^\dagger , em seguida, fazer o produto das matrizes A e L_A^\dagger . Pensando

em problemas de grande porte, tais operações são completamente inviáveis.

Nesta proposta, utilizamos a transformação para a forma padrão apenas de forma implícita, sem desenvolver o produto matriz-matriz AL_A^\dagger , e a incorporamos no algoritmo GKB-FP. É conhecido que o processo de bidiagonalização captura a parte dominante do espectro singular da matriz dos coeficientes em um número relativamente pequeno de passos. Desta forma, ao aplicar GKB-FP no problema transformado, calculamos uma aproximação para a SVD de \bar{A} , que contém as informações mais relevantes para o cálculo da solução \mathbf{y}_λ , mas que não requer o cálculo completo da GSVD de (A, L) .

Vamos considerar dois modos de proceder, de acordo com o fato da matriz $L \in \mathbb{R}^{p \times n}$, $p \leq n$, poder ser particionada como $L = [L_{11} \quad L_{12}]$, onde $L_{11} \in \mathbb{R}^{p \times p}$ é não singular. Chamamos este de *caso não singular*. Quando L_{11} for singular ou quando tal divisão não puder ser efetuada, dizemos que L é do *caso singular*.

4.1.1 Caso Não Singular

Para o cálculo de L_A^\dagger considere a seguinte partição de L , $T = (AW)^\dagger A$ e \mathbf{x} como

$$L = [L_{11} \quad L_{12}], \quad T = [T_{11} \quad T_{12}] \quad \text{e} \quad \mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \quad (4.5)$$

onde $L_{11} \in \mathbb{R}^{p \times p}$ é não singular, $T_{11} \in \mathbb{R}^{(n-p) \times p}$ e $\mathbf{x}_1 \in \mathbb{R}^p$.

Então, como verificado em [13], L_A^\dagger é dada por

$$L_A^\dagger = [I_n - WT] \begin{bmatrix} L_{11}^{-1} \\ 0 \end{bmatrix} = \left[\begin{pmatrix} I_p \\ 0 \end{pmatrix} - WT_{11} \right] L_{11}^{-1}. \quad (4.6)$$

A solução regularizada para o problema na forma geral será obtida seguindo os passos do método GKB-FP mas com os produtos matriz-vetor $\mathbf{z} = \bar{A}\mathbf{x}$ e $\bar{\mathbf{z}} = \bar{A}^T \mathbf{x}$ sendo desenvolvidos como segue, usando a equação (4.6):

Note que, como $\mathcal{N}(L)$ tem dimensão pequena, o cálculo da matriz T em (4.6) e do produto WT não apresenta qualquer

Para $\mathbf{z} = \overline{\mathbf{A}}\mathbf{x}$ faça:	Para $\overline{\mathbf{z}} = \overline{\mathbf{A}}^T \mathbf{x}$ faça:
$\widehat{\mathbf{x}} \leftarrow L_{11}^{-1} \mathbf{x}$	$\widehat{\mathbf{x}} \leftarrow \mathbf{A}^T \mathbf{x}$
$\overline{\mathbf{x}} \leftarrow \begin{pmatrix} \widehat{\mathbf{x}} \\ 0 \end{pmatrix} - WT_{11} \widehat{\mathbf{x}}$	$\overline{\mathbf{x}} \leftarrow \widehat{\mathbf{x}}_1 - T_{11}^T W^T \widehat{\mathbf{x}}$
$\mathbf{z} \leftarrow \mathbf{A} \overline{\mathbf{x}}$	$\overline{\mathbf{z}} \leftarrow L_{11}^{-T} \overline{\mathbf{x}}$

Tabela 4.1: Desenvolvimento dos produtos matriz-vetor envolvendo $\overline{\mathbf{A}}$ no caso não singular.

dificuldade para problemas de grande porte. No entanto, quando precisamos calcular $\widehat{\mathbf{x}} = L_{11}^{-1} \mathbf{x}$, resolvemos o sistema

$$L_{11} \widehat{\mathbf{x}} = \mathbf{x}, \quad (4.7)$$

podendo optar por métodos que sejam mais convenientes e mais econômicos de acordo com a estrutura da matriz L_{11} .

Por exemplo, se L é de banda com banda $n - p + 1$, então L_{11} é de banda e triangular superior, e resolver o sistema (4.7) torna-se particularmente simples. Em outros casos, por exemplo se L representa o Laplaciano, então uma fatoração LU da L_{11} pode ser necessária.

A transformação reversa

$$\mathbf{x}_\lambda = L_A^\dagger \mathbf{y}_\lambda + \mathbf{x}_{\lambda, \mathcal{N}} \quad (4.8)$$

também é realizada utilizando a equação (4.6), resolvendo sistemas lineares para determinar $L_{11}^{-1} \mathbf{y}_\lambda$.

4.1.2 Caso Singular

Se $p > n$ podemos fazer a fatoração QR da matriz L , mas note que não é preciso armazenar a matriz Q da fatoração, pois matrizes ortogonais são invariantes pela norma 2. Feito isso substituímos $\|L\mathbf{x}\|$ por $\|R\mathbf{x}\|$. Neste caso, podemos optar pela fatoração QR reduzida de L e obter $R \in \mathbb{R}^{r \times n}$ com $r = \text{posto}(L) \leq \min\{p, n\}$.

Desta forma, retornamos ao caso em que a matriz na semi-norma tem mais colunas do que linhas. Se R da fatoração reduzida é particionada como $R = [R_1 \ R_2]$ com $R_1 \in \mathbb{R}^{r \times r}$ não singular, aplicamos o método GGKB-FP como descrito no caso não singular.

O ponto então é determinar o que fazer quando $L \in \mathbb{R}^{p \times n}$ tem mais colunas do que linhas mas o bloco $L_{11} \in \mathbb{R}^{p \times p}$ é singular. Neste caso, optamos pelo uso direto da equação (4.3), isto é, os produtos matriz-vetor $\mathbf{z} = \overline{A}\mathbf{x}$ e $\overline{\mathbf{z}} = \overline{A}^T \mathbf{x}$ são desenvolvidos utilizando $\overline{A} = A \left(I - W (AW)^\dagger A \right) L^\dagger$.

Para isso é necessário o cálculo da matriz pseudo-inversa de L , o que requer certo esforço computacional tendo em vista as dimensões da matriz L . Assim, para efetuar os produtos $L^\dagger \mathbf{x}$ resolvemos o seguinte problema de minimização

$$\min \|L\mathbf{y} - \mathbf{x}\|_2 \quad (4.9)$$

onde $L^\dagger \mathbf{x}$ é a solução de mínima norma deste problema. A verificação disso segue do fato que $L(L^\dagger \mathbf{x}) = (LL^\dagger) \mathbf{x} = P_{\mathcal{R}(L)} \mathbf{x}$, ou seja, o vetor em $\mathcal{R}(L)$ mais próximo de \mathbf{x} .

Para o produto $(L^\dagger)^T \mathbf{x}$, usamos a equivalência $(L^\dagger)^T = (L^T)^\dagger$. De fato, seja a SVD de L dada por $L = U\Sigma V^T$ com

$$\Sigma = \begin{bmatrix} \Sigma_r & O \\ O & O \end{bmatrix} \quad \text{para } \Sigma_r = \text{diag}(\sigma_1, \dots, \sigma_r) \quad (4.10)$$

em que r é o posto da matriz L . Assim, $L = U_r \Sigma_r V_r^T$ onde U_r e V_r são as r primeiras colunas de U e V . Essa decomposição é chamada de SVD reduzida de L . Consequentemente, a SVD de L^T é $L^T = (U_r \Sigma_r V_r^T)^T = V_r \Sigma_r^T U_r^T$. Então, $L^\dagger = V_r \Sigma_r^{-1} U_r^T$. Assim

$$(L^\dagger)^T = (V_r \Sigma_r^{-1} U_r^T)^T = U_r (\Sigma_r^{-1})^T V_r^T = (\Sigma_r^T)^{-1} V_r^T = (L^T)^\dagger.$$

Analogamente, o cálculo de $(L^\dagger)^T \mathbf{x}$ é feito resolvendo

$$\min \|L^T \mathbf{y} - \mathbf{x}\|_2. \quad (4.11)$$

Uma dificuldade a ser enfrentada é que para cada passo do processo de bidiagonalização de Golub-Kahan, são necessários dois produtos matriz-vetor envolvendo L^\dagger ou $(L^\dagger)^T$. Assim, é uma preocupação constante o modo como esses problemas de minimização serão resolvidos pois por um lado precisamos de métodos que tenham baixo custo computacional para que os passos GKB sejam efetuados de forma rápida mas também precisamos de boas soluções, no sentido de precisão, pois por ser um processo de recorrência, erros de arredondamento podem comprometer a solução final.

O cálculo de $\mathbf{z} = \overline{A}\mathbf{x}$ e $\overline{\mathbf{z}} = \overline{A}^T \mathbf{x}$ é feito como segue:

Para $\mathbf{z} = \overline{A}\mathbf{x}$ faça:	Para $\overline{\mathbf{z}} = \overline{A}^T \mathbf{x}$ faça:
$\mathbf{x}_1 \leftarrow L^\dagger \mathbf{x}$	$\mathbf{x}_1 \leftarrow A^T \mathbf{x}$
$\mathbf{x}_2 \leftarrow W(T\mathbf{x}_1)$	$\mathbf{x}_2 \leftarrow T^T(W^T \mathbf{x}_1)$
$\mathbf{x}_4 \leftarrow \mathbf{x}_1 - \mathbf{x}_2$	$\mathbf{x}_4 \leftarrow \mathbf{x}_1 - \mathbf{x}_2$
$\mathbf{z} \leftarrow A\mathbf{x}_4$	$\mathbf{z} \leftarrow (L^\dagger)^T \mathbf{x}_4$

Tabela 4.2: Desenvolvimento dos produtos matriz-vetor envolvendo \overline{A} no caso singular.

4.1.3 Critério de Parada

O critério de parada para o método GGKB-FP é o mesmo apresentado para o método GKB-FP, apresentado na seção 2.4. Esta escolha foi feita levando-se em conta que este método é a implementação do método de Bazán e Borges para regularização na forma geral.

4.1.4 Algoritmo

Após discutir as formas de calcular L_A^\dagger , apresentamos o algoritmo desta proposta. A Tabela 4.3 traz o algoritmo do método GKB-FP para problemas na forma geral, denominado por GGKB-FP.

Os parâmetros $q > 1$, $kmax$, ϵ_1 e ϵ_2 são como no método GKB-FP, sendo o padrão, usamos $\epsilon_1 = \epsilon_2 = 10^{-4}$, $kmax =$

GGKB-FP

Entrada: $A, L, W, \mathbf{b}, q > 1, kmax, \epsilon_1$ e ϵ_2 .

Saída: Solução regularizada $\mathbf{x}_{\lambda^*}^{(k)}$, k^* e λ^* .

1. Calcular T e $\bar{\mathbf{b}}$.
2. Aplicar os passos do algoritmo GKB-FP, desenvolvendo $\bar{A}\mathbf{x}$ e $\bar{A}^T \mathbf{x}$ como na Tabela 4.1 (ou Tabela 4.2) .
3. Obter $\mathbf{y}_{\lambda^*}^k$, solução de GKB-FP.
4. Aplicar a transformação reversa, obtendo $\mathbf{x}_{\lambda^*}^k$.

Tabela 4.3: Algoritmo GGKB-FP.

$n - 1$ e $q = 5$. Este valor para o parâmetro q funcionou na maior parte dos problemas testados. Os dados de saída k^* e λ^* correspondem a dimensão máxima dos espaços projetados e ao parâmetro de regularização utilizado para o cálculo da solução $\mathbf{x}_{\lambda^*}^{(k)}$, respectivamente, como no método GKB-FP.

4.2 PROJ-FP

Uma outra abordagem proposta para o problema

$$\mathbf{x}_\lambda = \operatorname{argmin} \left\{ \|\mathbf{b} - A\mathbf{x}\|_2^2 + \lambda^2 \|L\mathbf{x}\|_2^2 \right\}$$

também baseia-se no processo de bidiagonalização de Golub-Kahan e no método LSQR. De maneira resumida, trabalharemos com um problema projetado regularizado, de dimensão consideravelmente menor do que a do problema original, que por sua vez pode ser resolvido por métodos já conhecidos, como a Curva L, GCV e Ponto Fixo (FP).

Seja $A \in \mathbb{R}^{m \times n}$ e $\mathbf{b} \in \mathbb{R}^m$. Considere as matrizes $U_{k+1} \in \mathbb{R}^{m \times (k+1)}$, $V_k \in \mathbb{R}^{n \times k}$ e B_k geradas pelo k -ésimo passo GKB para a matriz A iniciando com $\mathbf{u}_1 = \frac{\mathbf{b}}{\|\mathbf{b}\|}$. Na seção 2.2 vimos que

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

O novo método, designado por PROJ-FP, consiste em obter soluções aproximadas $\mathbf{x}_\lambda^{(k)}$ para \mathbf{x}_λ a partir das soluções $\mathbf{y}_\lambda^{(k)}$ dadas por

$$\mathbf{y}_\lambda^{(k)} = \operatorname{argmin}_{\mathbf{y} \in \mathbb{R}^k} \left\{ \|\beta_1 \mathbf{e}_1 - B_k \mathbf{y}\|^2 + \lambda^2 \|L_k \mathbf{y}\| \right\} \quad (4.12)$$

onde $L_k = LV_k \in \mathbb{R}^{p \times k}$. Neste caso, $\mathbf{x}_\lambda^{(k)} = V_k \mathbf{y}_\lambda^{(k)}$.

O parâmetro de regularização λ para o problema (4.12) é obtido pelo Método de Ponto fixo apresentado na seção 1.6.1, para um problema pequeno na forma geral, e a solução $\mathbf{y}_\lambda^{(k)}$ calculada pelo método de Tikhonov como descrito na seção 1.4.

Como em geral $k \ll p$, podemos pensar em uma fatoração QR para a matriz L_k , já que esta tem mais linhas do que colunas. Seja

$$L_k = Q_k R_k, \quad Q_k \in \mathbb{R}^{p \times k} \quad \text{e} \quad R_k \in \mathbb{R}^{k \times k} \quad (4.13)$$

a fatoração QR reduzida de L_k . Como Q_k tem colunas ortonormais segue que

$$\|L_k \mathbf{y}\|_2^2 = \|Q_k R_k \mathbf{y}\|_2^2 = \|R_k \mathbf{y}\|_2^2 \quad (4.14)$$

Desta forma, substituímos o problema (4.12) por

$$\mathbf{y}_\lambda^{(k)} = \operatorname{argmin}_{\mathbf{y} \in \mathbb{R}^k} \left\{ \|\beta_1 \mathbf{e}_1 - B_k \mathbf{y}\|^2 + \lambda^2 \|R_k \mathbf{y}\| \right\} \quad (4.15)$$

e resolvemos da mesma maneira que a apresentada para (4.12).

Em linhas gerais o algoritmo tem a seguinte estrutura:

1. Aplicar q passos GKB para A iniciando com $\mathbf{u}_1 = \frac{\mathbf{b}}{\|\mathbf{b}\|}$ e formar a matriz B_q .
2. Definir $k = q$, determinar a fatoração QR de L_k , calcular o parâmetro $\lambda^{(k)}$ para o problema projetado (4.15) de dimensão k pelo método FP e definir $\lambda_0 = \lambda^{(k)}$, $\lambda_{old} = \lambda_0$ e $k \leftarrow k + 1$.

3. Executar mais um passo GKB, determinar a fatoração QR de L_k e calcular o novo parâmetro $\lambda^{(k)}$ começando com λ_0 como valor inicial. Definir $\lambda_{old} = \lambda_0$ e $\lambda_0 = \lambda^{(k)}$.
4. Se o critério de parada for satisfeito, $\lambda^* = \lambda_{old}$. Caso contrário, definir $k \leftarrow k + 1$ e ir para o passo 3.
5. Calcular a solução do problema projetado $\mathbf{y}_{\lambda^*}^{(k)}$.
6. Calcular a solução regularizada $\mathbf{x}_{\lambda^*}^{(k)} \leftarrow V_k \mathbf{y}_{\lambda^*}^{(k)}$.

No passo 3, note que em cada novo passo de GKB é necessário calcular a fatoração QR da matriz L_k . Uma observação importante é que a fatoração QR no passo 3 pode ser obtida atualizando a fatoração feita no passo 2. O lema 4.1 apresenta como obter a fatoração QR de L_{k+1} a partir da QR de L_k .

Consideremos inicialmente uma matriz arbitrária $P \in \mathbb{R}^{m \times n}$. Então sua fatoração QR é dada pelo algoritmo apresentado na Tabela 4.4, usando o método de Gram-Schmidt [18]. Neste caso, \mathbf{P}_j e \mathbf{Q}_j são as j -ésimas colunas de P e Q , e $R_{i,j}$ é o elemento da matriz R que está na i -ésima linha e na j -ésima coluna.

<p>Entrada: P Saída: Q e R.</p> <ol style="list-style-type: none"> 1. $[m, n] = \text{dimensão}(P)$. 2. Para j de 1 até n $\mathbf{Q}_j = \mathbf{P}_j$; <ol style="list-style-type: none"> 2.1. Para i de 1 até $j - 1$ $R_{i,j} = \mathbf{Q}_i^T \mathbf{P}_j$ e $\mathbf{Q}_j \leftarrow \mathbf{Q}_j - R_{i,j} \mathbf{Q}_i$. Fim Para $R_{j,j} = \ \mathbf{Q}_j\ ^2$ e $\mathbf{Q}_j \leftarrow \frac{\mathbf{Q}_j}{R_{j,j}}$ Fim Para

Tabela 4.4: Algoritmo da fatoração QR via Gram-Schmidt.

Lema 4.1. *Seja $L_k = LV_k \in \mathbb{R}^{p \times k}$ e sua fatoração QR dada por $L_k = Q_k R_k$, com $Q_k \in \mathbb{R}^{p \times k}$ e $R_k \in \mathbb{R}^{k \times k}$. Então a fatoração*

QR de L_{k+1} é dada por

$$L_{k+1} = [Q_k \quad \mathbf{q}_{k+1}] \begin{bmatrix} R_k & \mathbf{r}_{k+1}^* \\ \mathbf{0}^T & r_{k+1,k+1} \end{bmatrix}$$

em que $\mathbf{0} \in \mathbb{R}^k$, $\mathbf{r}_{k+1}^* \in \mathbb{R}^k$ e $r_{k+1,k+1} \in \mathbb{R}$.

Demonstração. Vamos considerar a fatoração QR obtida usando o algoritmo da Tabela 4.4. Note que

$$L_{k+1} = L [V_k \quad \mathbf{v}_{k+1}] = [LV_k \quad L\mathbf{v}_{k+1}] = [L_k \quad L\mathbf{v}_{k+1}]. \quad (4.16)$$

Considere a fatoração QR de L_k como

$$L_k = Q_k R_k = [\mathbf{q}_1 \quad \mathbf{q}_2 \quad \dots \quad \mathbf{q}_k] [\mathbf{r}_1 \quad \mathbf{r}_2 \quad \dots \quad \mathbf{r}_k] \quad (4.17)$$

em que $\mathbf{q}_j \in \mathbb{R}^p$ e $\mathbf{r}_j = [r_{1,j} \quad r_{2,j} \quad \dots \quad r_{k,j}]^T \in \mathbb{R}^k$, $j = 1, 2, \dots, k$.

Tome j^* um número natural entre 1 e k . No passo 2 do algoritmo, para $j = j^*$, são gerados os vetores \mathbf{q}_{j^*} e \mathbf{r}_{j^*} , como na equação (4.17). Para isto, são usados apenas a coluna j^* de L_k e os vetores \mathbf{q}_j e \mathbf{r}_j para $j < j^*$.

Como L_{k+1} difere de L_k apenas pela última coluna, as k primeiras execuções do passo 2 para o cálculo da fatoração QR de L_{k+1} são iguais às k execuções do passo 2 para o cálculo da fatoração QR de L_k .

Assim, $Q_{k+1} = [Q_k \quad \mathbf{q}_{k+1}]$ em que o vetor \mathbf{q}_{k+1} é obtido no passo 2 para $j = k + 1$.

Quanto à matriz R_{k+1} , o primeiro bloco $k \times k$ é a matriz R_k da fatoração QR de L_k , que é triangular superior. O vetor \mathbf{r}_{k+1}^* é obtido no passo 2.1 para $j = k + 1$ e i de 1 até k e o número $r_{k+1,k+1}$ é calculado no final do passo 2, também para $j = k + 1$.

■

A atualização da fatoração QR para a matriz L_{k+1} pode também ser realizada por meio de transformações de Householder mas neste caso, o custo computacional é maior.

4.2.1 Algoritmo

Na Tabela 4.5 apresentamos o algoritmo para o método PROJ-FP. Note que no passo 2, resolvemos um problema de Tikhonov

na forma geral, porém restrito, usando o Método de Ponto Fixo apresentado no primeiro capítulo. Também, a solução calculada no passo 5 é como em (1.24) e requer o cálculo da GSVD de (B_k, L_k) . Estes métodos são utilizados de forma bastante eficiente, visto que o problema restrito é um problema pequeno (dimensão k).

PROJ-FP

Entrada: $A, L, \mathbf{b}, q > 1, kmax, \epsilon_1$ e ϵ_2 .

Saída: Solução regularizada $\mathbf{x}_{\lambda^*}^{(k)}$, k^* e λ^* .

1. Aplicar q passos GKB para A iniciando com $\mathbf{u}_1 = \frac{\mathbf{b}}{\|\mathbf{b}\|}$;
Formar a matriz B_q .
2. Definir $k = q$;
Determinar a fatoração QR de L_k ;
Calcular o parâmetro $\lambda^{(k)}$ para o problema projetado (4.15) de dimensão k pelo método FP;
Definir $\lambda_0 = \lambda^{(k)}$, $\lambda_{old} = \lambda_0$ e $k \leftarrow k + 1$.
3. Executar mais um passo GKB;
Obter a fatoração QR de L_k a partir da atualização da QR de L_{k-1} ;
Calcular o novo parâmetro $\lambda^{(k)}$ começando com λ_0 como valor inicial.
Definir $\lambda_{old} = \lambda_0$ e $\lambda_0 = \lambda^{(k)}$.
4. **Se** (critério de parada satisfeito)
 $\lambda^* = \lambda_{old}$, $k^* = k$.
Senão
Definir $k \leftarrow k + 1$ e ir para o passo 3.
Fim Se
5. Calcular a solução do problema projetado $\mathbf{y}_{\lambda^*}^{(k)}$ usando a equação (1.24).
6. Calcular a solução regularizada $\mathbf{x}_{\lambda^*}^{(k)} \leftarrow V_k \mathbf{y}_{\lambda^*}^{(k)}$.

Tabela 4.5: Algoritmo PROJ-FP.

Novamente, as saídas k^* e λ^* correspondem à dimensão máxima dos espaços projetados e ao parâmetro de regularização utilizado

para o cálculo da solução $\mathbf{x}_{\lambda^*}^{(k)}$, respectivamente.

4.2.2 Critério de Parada

O critério de parada para o método é o mesmo utilizado em GKB-FP, isto é, são utilizados dois critérios

- a) $|\lambda^{(k+1)*} - \lambda^{(k)*}| < \epsilon_1 |\lambda^{(k)*}|$, onde ϵ_1 é uma tolerância pequena.
- b) $|\lambda^{(k+1)*} - \lambda^{(k)*}| < \epsilon_2 |\lambda^{(0)*}|$, onde $\lambda^{(0)*}$ é primeiro ponto fixo calculado e ϵ_2 é pequeno.

Como a sequência de pontos fixos calculada pelo método atende as mesma propriedades dos pontos fixos obtida pelo método GKB-FP, usar o mesmo critério de parada é uma escolha razoável.

4.3 G-LSQR: Regularização Iterativa

A proposta deste método é evitar a Regularização de Tikhonov, já que esta traz consigo a tarefa de escolher um parâmetro de regularização. Para tal fim, note que no problema de Tikhonov na forma geral

$$\mathbf{x}_\lambda = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^n} \{ \|\mathbf{b} - A\mathbf{x}\|_2^2 + \lambda^2 \|L\mathbf{x}\|_2^2 \},$$

a matriz L desempenha um papel importante, pois incorpora à solução aproximada informações sobre a solução exata e que para o problema transformado como na equação (3.53) $\lambda = 0$ produz

$$\bar{\mathbf{y}} = \operatorname{argmin}_{\mathbf{y} \in \mathbb{R}^p} \|\bar{\mathbf{b}} - \bar{A}\mathbf{y}\|_2^2. \quad (4.18)$$

onde a influência da matriz L já está incorporada na matriz \bar{A} . Isto sugere que calculando aproximações estáveis para (4.18), podemos também calcular soluções estáveis para $\mathbf{x}_{\text{exato}}$ usando a transformação reversa dada em (3.54).

A proposta do método G-LSQR é obter soluções aproximadas $\bar{\mathbf{y}}^{(k)}$ para $\bar{\mathbf{y}}$ através do método LSQR, utilizando a equivalência:

$$\begin{aligned}
\bar{\mathbf{y}}^{(k)} &= \operatorname{argmin}_{\mathbf{y} \in \mathcal{K}_k(\bar{\mathbf{A}}^T \bar{\mathbf{A}}, \bar{\mathbf{A}}^T \bar{\mathbf{b}})} \|\bar{\mathbf{b}} - \bar{\mathbf{A}}\mathbf{y}\|_2^2 \\
&\quad \Updownarrow \\
\bar{\mathbf{y}}^{(k)} &= \bar{\mathbf{V}}_k \mathbf{z}^{(k)}, \quad \mathbf{z}^{(k)} = \operatorname{argmin}_{\mathbf{z} \in \mathbb{R}^k} \|\bar{\beta}_1 \mathbf{e}_1 - \bar{\mathbf{B}}_k \mathbf{z}\|_2
\end{aligned} \tag{4.19}$$

e, após isso, calcular aproximações $\mathbf{x}^{(k)}$ para $\mathbf{x}_{\text{exato}}$ como

$$\mathbf{x}^{(k)} = L_A^\dagger \bar{\mathbf{y}}^{(k)} + \mathbf{x}_{\mathcal{N}}. \tag{4.20}$$

Desta forma estaremos considerando a influência da matriz L na solução e ao mesmo tempo tornando o método atrativo para problemas de grande porte. Só consideramos o caso em que L tem mais colunas do que linhas.

4.3.1 Critério de Parada

Considerando o problema exato

$$\mathbf{x}_{\text{exato}} = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{b}_{\text{exato}} - \mathbf{A}\mathbf{x}\|_2$$

vamos inicialmente estudar o comportamento da função

$$E(k) = \frac{\|\mathbf{x}_{\text{exato}} - \mathbf{x}^{(k)}\|}{\|\mathbf{x}_{\text{exato}}\|} \tag{4.21}$$

que descreve o erro relativo associado às soluções aproximadas $\mathbf{x}^{(k)}$ com relação a solução exata $\mathbf{x}_{\text{exato}}$.

Uma característica de $E(k)$ é que a medida que as iterações são realizadas, o erro relativo tende a diminuir pois \mathcal{V}_k se aproxima de \mathbb{R}^n e $\mathbf{x}^{(k)}$ captura informação da solução exata. Sendo assim, a função $E(k)$ é inicialmente decrescente. No entanto, a medida que o número de iterações aumenta, mais ruído é incorporado em $\mathbf{x}^{(k)}$, o que leva a um crescimento na função $E(k)$.

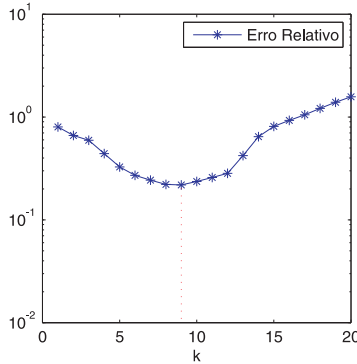


Figura 4.1: Função $E(k)$.

Desta forma, o gráfico da função $E(k)$ é usualmente como na Figura 4.1. Neste caso, é intuitivo parar as iterações no valor de k que minimiza $E(k)$, isto é, a iteração de parada deve ser dada por

$$k^* = \operatorname{argmin} E(k), \quad k \in \mathbb{N}^*. \quad (4.22)$$

Para problemas reais, no entanto, a solução $\mathbf{x}_{\text{exato}}$ não é disponível e o minimizador k^* não pode ser determinado. Porém, como veremos a seguir, a ideia de minimizar $E(k)$ pode ser útil para desenvolver um critério de parada para G-LSQR.

De fato, vimos no capítulo 2, que para soluções $\mathbf{x}^{(k)}$ calculadas pelo método LSQR vale:

$$\|\mathbf{x}^{(k+1)}\|_2 \geq \|\mathbf{x}^{(k)}\|_2, \quad \|\mathbf{r}^{(k+1)}\| \leq \|\mathbf{r}^{(k)}\| \quad (4.23)$$

onde $\mathbf{r}^{(k)} = \mathbf{b} - A\mathbf{x}^{(k)}$. Sendo assim, a função

$$\Psi(k) = \|\mathbf{x}^{(k)}\|_2 \|\mathbf{r}^{(k)}\|_2 \quad (4.24)$$

tem comportamento semelhante ao da Figura 4.2, por ser o produto de uma quantidade crescente e uma decrescente. A velocidade de crescimento/decrescimento de $\Psi(k)$ vai depender do comportamento das normas.

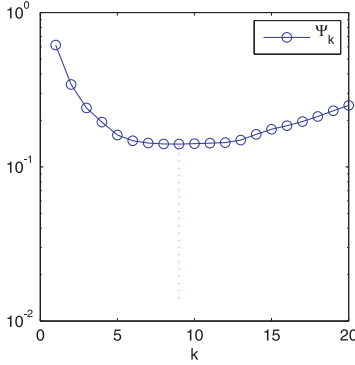


Figura 4.2: Comportamento da função $\Psi(k)$.

Com base nos comportamentos de $\Psi(k)$ e $E(k)$, a proposta é usar como critério de parada o k^* satisfazendo

$$k^* = \operatorname{argmin} \Psi(k), \quad k \in \mathbb{N}^*. \quad (4.25)$$

Note que para o problema transformado podem ser feitas as mesmas considerações a respeito das funções

$$\overline{E}(k) = \frac{\|\overline{\mathbf{y}}_{\text{exato}} - \overline{\mathbf{y}}^{(k)}\|_2}{\|\overline{\mathbf{y}}_{\text{exato}}\|_2} \quad (4.26)$$

em que $\overline{\mathbf{y}}_{\text{exato}}$ é a solução exata de (4.18), considerando o lado direito exato. Neste caso, para $\overline{\mathbf{r}}^{(k)} = \overline{\mathbf{b}} - \overline{\mathbf{A}}\overline{\mathbf{y}}^{(k)}$ temos

$$\overline{\Psi}(k) = \|\overline{\mathbf{y}}^{(k)}\|_2 \|\overline{\mathbf{r}}^{(k)}\|_2. \quad (4.27)$$

Pensando em resolver o problema (4.18), tomamos como iteração de parada para G-LSQR, a iteração k^* tal que

$$k^* = \operatorname{argmin} \overline{\Psi}(k), \quad k \in \mathbb{N}^*. \quad (4.28)$$

Para evitar que o método faça iterações desnecessárias, no caso da função Ψ ter um mínimo localizado em uma região plana, adotamos como critério parar o proceso quando

$$|\Psi(k) - \Psi(k-1)| < \epsilon_1 |\Psi(1)| \quad (4.29)$$

onde ϵ_1 é uma tolerância pequena. Sendo assim, o processo iterativo para quando (4.28) ou (4.29) for satisfeita.

4.3.2 Algoritmo

Com o critério de parada determinado, apresentamos a algoritmo de G-LSQR na Tabela 4.6. Neste caso, $kmax$ é o número de iterações máximo e $\epsilon_1 = 10^{-3}$ a tolerância usual. O parâmetro q_1 determina a dimensão do primeiro espaço projetado e o primeiro ponto onde a função $\overline{\Psi}$ será avaliada. O dado de saída k^* é a dimensão máxima do espaço projetado.

G-LSQR

Entrada: $A, L, W, \mathbf{b}, kmax, q_1 \geq 1, q_2 \geq 0$ e ϵ_1 .

Saída: Solução regularizada $\mathbf{x}^{(k)}, k^*$.

1. Calcular $T = (AW)^\dagger A, \overline{\mathbf{b}}$ e \mathbf{x}_N e definir $k = q_1$.
2. Aplicar k passos GKB para \overline{A} desenvolvendo os produtos matriz-vetor como na Tabela 4.1.
3. Calcular a solução $\overline{\mathbf{y}}^{(k)}$ do problema transformado (4.19).
4. Definir $k = k + 1$;
Executar mais um passo GKB como no passo 2;
Calcular a solução $\overline{\mathbf{y}}^{(k)}$ do problema transformado (4.19).
5. **Se** (critério de parada satisfeito)
 $k^* = k - 1$;
Calcular a solução $\mathbf{x}^{(k^*)}$ como em (4.20).

FIM

Senão

Ir para o passo 4.

Fim Se

Tabela 4.6: Algoritmo G-LSQR.

Capítulo 5

Resultados Numéricos

Apresentamos os resultados obtidos para os métodos propostos no capítulo anterior, comparando-os entre si e com outros métodos já conhecidos. Ilustraremos nossas propostas por meio de problemas teste da *toolbox* RegularizationTools e problemas relacionados ao tratamento de imagens.

O objetivo aqui é resolver

$$\min \left\{ \|\mathbf{b} - A\mathbf{x}\|_2^2 + \lambda^2 \|L\mathbf{x} - L\mathbf{x}_0\|_2^2 \right\}$$

em que o parâmetro de regularização também deve ser determinado. Nos testes consideramos $\mathbf{x}_0 = \mathbf{0}$ pois usualmente não dispomos de uma aproximação inicial; e o lado direito \mathbf{b} como um dado inexato, isto é, $\mathbf{b} = \mathbf{b}_{\text{exato}} + \mathbf{e}$, onde \mathbf{e} é um vetor de perturbações. Na prática, isso ocorre frequentemente pois, em geral, \mathbf{b} provém da discretização de uma função contínua obtida experimentalmente e, portanto, sujeita a erros de medição.

Em todos os problemas, o vetor de perturbações \mathbf{e} , gerado pela rotina *randn* do MATLAB, contém números aleatórios distribuídos normalmente com média zero e desvio padrão um e

$$\mathbf{b} = \mathbf{b}_{\text{exato}} + \|\mathbf{b}_{\text{exato}}\|_2 0.01 NL (\mathbf{e} / \|\mathbf{e}\|_2);$$

onde NL é o nível de ruído (*Noise Level*) desejado. Por exemplo, $NL = 1$ resulta em 1% de ruído. Todos os testes apresentados neste trabalho foram executados usando MATLAB versão 7.0.1.

5.1 Problemas Teste

A *toolbox* RegularizationTools de P.C. Hansen [24, 25] é um conjunto de rotinas para o MATLAB destinadas ao tratamento de problemas discretos mal postos (criação, análise e resolução), que inclui vários problemas teste com uma rotina para cada um deles, conforme a Tabela 5.1. Os problemas utilizados em sua grande maioria, provêm da discretização de equações integrais de Fredholm de primeira espécie.

Rotina	Descrição
baart	Equação integral de Fredholm de primeira espécie.
blur	Restauração de imagem.
deriv2	Cálculo da segunda derivada.
foxford	Problema extremamente mal posto.
gravity	Problema gravitacional unidimensional.
heat	Equação do calor inversa.
i_laplace	Inversão da transformada de Laplace.
parallax	Problema de observação estelar com dados reais.
phillips	Equação estudada por Phillips.
shaw	Modelo de restauração de imagem unidimensional.
spikes	Problema teste com solução tipo “picos”.
tomo	Tomografia bidimensional com matriz esparsa.
ursell	Problema teste sem solução.
wing	Problema teste com solução descontínua.

Tabela 5.1: Problemas teste do *toolbox* RegularizationTools.

De modo geral, as rotinas funcionam da mesma maneira: o usuário fornece o tamanho do sistema desejado, recebendo como variáveis de saída a matriz A , o vetor $\mathbf{b}_{\text{exato}}$ e a solução exata do problema no vetor \mathbf{x} . Nem todos os problemas têm solução e algumas rotinas possuem mais de um parâmetro de entrada. Para mais detalhes a respeito das rotinas basta usar o comando `help` do MATLAB.

A seguir apresentaremos uma sucinta descrição de cada um dos problemas usados para testar os métodos do capítulo 3, incluindo uma análise da condição discreta de Picard. Analisamos

a condição discreta de Picard para o problema na forma geral, isto é, os valores singulares generalizados γ_i do par (A, L) , onde A é fornecido para cada um dos problemas teste, com L igual a L_1 e L_2 dadas como:

$$L_1 = \begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix} \in \mathbb{R}^{(n-1) \times n} \quad (5.1)$$

uma aproximação discreta para o operador de primeira derivada e

$$L_2 = \begin{bmatrix} 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \end{bmatrix} \in \mathbb{R}^{(n-2) \times n} \quad (5.2)$$

para a segunda derivada. Outros exemplos de matrizes L podem ser encontradas em [30] com detalhes.

A dimensão escolhida para os testes foi $n = 1024$ e os problemas foram executados 50 vezes com diferentes vetores de perturbação e , para ilustrar o desempenho dos métodos. Os resultados obtidos com GGKB-FP, PROJ-FP e G-LSQR são também comparados com as soluções calculadas pelos métodos da Curva L e GCV, que usam a decomposição GSVD.

Os parâmetros iniciais para os métodos do capítulo 3 foram definidos como apresentado na Tabela 5.2. Se o problema de dimensão q não apresentar ponto-fixo, verificamos os problemas de dimensão imediatamente maior até $q = 20$. Se a função $\Psi_\mu(\lambda)$ ainda assim não apresentar ponto-fixo, o parâmetro μ é ajustado para que a nova função Ψ_μ apresente ponto-fixo. Esse ajuste é discutido em [4], no entanto, manter $\mu = 1$ e variar q entre 5 e 20, foi suficiente para garantir a existência de pelo menos um ponto-fixo, na maioria dos testes realizados.

Os símbolos utilizados na descrição dos resultados são dados na Tabela 5.3.

Os problemas a seguir foram resolvidos utilizando-se níveis de ruído de 0, 1%, 1% e 2, 5% no lado direito.

	GGKB-FP	PROJ-FP	G-LSQR
q	5	5	-
μ (FP)	1	1	-
ϵ_1	10^{-4}	10^{-4}	10^{-4}
ϵ_2	10^{-4}	10^{-4}	-
λ_0 (FP)	10^{-4}	10^{-4}	-

Tabela 5.2: Parâmetros iniciais para os problemas testes.

Símbolo	Descrição
$\bar{\lambda}$	Média dos parâmetros encontrados em todas as execuções.
$\lambda_{\min,\max}$	Parâmetro mínimo (resp. máximo).
λ_{std}	Desvio padrão dos parâmetros encontrados.
\bar{E}	Média dos erros relativos encontrados.
$E_{\min,\max}$	Erro relativo mínimo (resp. máximo)
E_{std}	Desvio padrão dos erros relativos encontrados.
$k_{\min,\max}$	Dimensão mínima (resp. máxima) do subespaço projetado.
t_{GSVD}	Tempo para o cálculo da GSVD de A, L .
\bar{t}	Média dos tempos de execução dos programas (em segundos).
$t_{\min,\max}$	Tempo mínimo (resp. máximo) de execução (em segundos).
t_{std}	Desvio padrão dos tempos.
\bar{t}_{total}	Soma do tempo médio t_{GSVD} .

Tabela 5.3: Símbolos para descrição dos resultados numéricos.

5.1.1 Gravity

A modelagem de um problema gravitacional unidimensional resulta numa equação integral de Fredholm de primeira espécie com núcleo K

$$K(x, y) = d(d^2 + (x - y)^2)^{-\frac{3}{2}} \quad (5.3)$$

e a solução f é dada por

$$f(y) = \text{sen}(\pi y) + \frac{1}{2} \text{sen}(2\pi y) \quad (5.4)$$

em que a constante d representa a profundidade na qual o centro de gravidade está localizado. Quanto maior for a profundidade, mais rápido os valores singulares caem para zero. Além da função f definida acima, podemos escolher f como sendo uma função linear por partes ou ainda uma função constante por partes. Na Figura 5.1 temos a solução exata para f , dada em (5.4), representada por uma curva suave. Na Figura 5.2 temos os coeficientes para a inspeção visual da condição discreta de Picard.

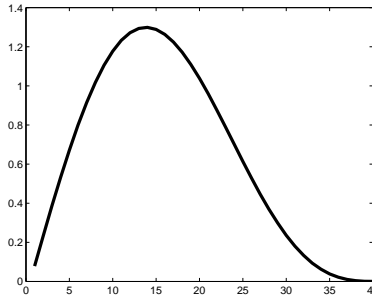


Figura 5.1: *Gravity*: Solução exata.

As tabelas 5.4 a 5.9 trazem os resultados obtidos usando o exemplo 1, com diferentes níveis de ruído e usando as matrizes L_1 e L_2 .

Para as Tabelas 5.4 e 5.5, os erros foram aceitáveis, mantendo-se em torno de 2% para L_1 e menores que 1% para L_2 , isso mostra como a escolha da matriz L é importante para a obtenção de resultados satisfatórios. Os erros dos métodos propostos no capítulo 3 também são próximos aos resultados obtidos pelos métodos da Curva L e GCV. O desvio padrão para os métodos GGKB-FP, PROJ-FP e G-LSQR também são pequenos, o que indica que a maior parte dos erros obtidos nas 50 execuções está próximo da média. Isso mostra que os métodos apresentam boa confiabilidade.

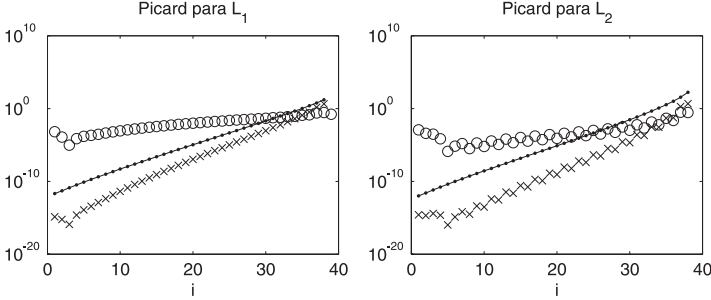


Figura 5.2: *Gravity*: Avaliação da CDP para (A, L_1) e (A, L_2) . Coeficientes $|\mathbf{u}_i^T \mathbf{b}|/\gamma_i$: ○, Coeficientes de Fourier: ×, Valores singulares: ·.

O erro máximo para o método GCV foi bastante grande. Isso ocorre em alguns casos para este método, tanto para erros de 0,1% como para erros maiores. No entanto, os erros mínimos são, em geral, próximos aos erros médios dos demais métodos. Para maiores detalhes sobre o porquê deste fato, consulte o apêndice B e as referências nele apresentadas.

Da mesma forma, para as tabelas 5.6 e 5.7, obtemos erros médios bons e com desvio padrão pequeno. Mas enquanto, a escolha $L = L_1$ produziu erros entre 3% e 5%, com $L = L_2$ os erros médios novamente ficaram menores que 2%, como no caso para ruído de 0,1% no lado direito \mathbf{b} .

Por fim, para 2,5% de ruído, os erros médios ficaram em torno de 5% a 8% para L_1 e entre 3% e 6% para L_2 sendo que o método G-LSQR apresentou erros médios menores ou iguais ao método da Curva L.

Para quaisquer escolhas de L ou intensidade de ruído, as dimensões dos subespaços projetados não ultrapassaram 29, o que equivale a 2,8% da dimensão total $n = 1024$. Em particular, para o método G-LSQR, a maior dimensão foi 9.

Para este problema, com todos os níveis de ruído, o método G-LSQR teve um desempenho superior aos demais métodos e a matriz L_2 produziu melhores resultados. No entanto, os métodos GGKB-FP e PROJ-FP também apresentaram resultados razoáveis.

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	1,6078	1,6020	-	1,2650	0,4396
λ_{\min}	1,5898	1,5801	-	1,0280	0,0068
λ_{\max}	1,6172	1,6144	-	1,4395	0,7668
λ_{std}	0,0054	0,0070	-	0,0963	0,1972
\bar{E}	0,0220	0,0203	0,0220	0,0207	0,0477
E_{\min}	0,0198	0,0140	0,0198	0,0188	0,0143
E_{\max}	0,0245	0,0255	0,0245	0,0242	0,8411
E_{std}	0,0012	0,0031	0,0010	0,0012	0,1244
k_{\min}	9	9	8	-	-
k_{\max}	10	17	9	-	-
t_{GSVD}	-	-	-	140,545	140,545
\bar{t}	0,1739	0,2240	0,1608	0,2166	0,1548
t_{\min}	0,1520	0,1580	0,1430	0,1950	0,1380
t_{\max}	0,2300	0,3600	0,2110	0,5960	0,2280
t_{std}	0,0181	0,0442	0,0119	0,0559	0,0159
\bar{t}_{total}	0,1739	0,2240	0,1608	140,762	140,700

Tabela 5.4: Resultados obtidos para o problema *Gravity* após 50 execuções com 0,1% de ruído nos dados e usando a matriz L_1 .

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	3,1629e+2	2,9609e+2	-	2,6186e+2	1,1516e+2
λ_{\min}	3,1316e+2	2,6177e+2	-	2,0876e+2	0,1488
λ_{\max}	3,1739e+2	3,1702e+2	-	2,9225e+2	2,3777e+2
λ_{std}	0,8999	20,1494	-	18,5126	62,5796
\bar{E}	0,0037	0,0066	0,0035	0,0034	0,0315
E_{\min}	0,0027	0,0029	0,0019	0,0024	0,0018
E_{\max}	0,0057	0,0159	0,0056	0,0055	0,8455
E_{std}	0,0007	0,0035	0,0012	0,0007	0,1274
k_{\min}	7	10	5	-	-
k_{\max}	7	28	6	-	-
t_{GSVD}	-	-	-	127,281	127,281
\bar{t}	0,1387	0,3145	0,1241	0,2246	0,1647
t_{\min}	0,1240	0,1740	0,1010	0,1990	0,1430
t_{\max}	0,3490	0,5540	0,1630	0,6930	0,4970
t_{std}	0,0341	0,1084	0,0138	0,0694	0,0549
\bar{t}_{total}	0,1387	0,3145	0,1241	127,506	127,446

Tabela 5.5: Resultados obtidos para o problema *Gravity* após 50 execuções com 0,1% de ruído nos dados e usando a matriz L_2 .

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	1,7377e+1	1,7327e+1	-	1,2241e+1	2,6479
λ_{\min}	1,7054e+1	1,6983e+1	-	1,0799e+1	0,0068
λ_{\max}	1,7523e+1	1,7491e+1	-	1,3679e+1	5,0924
λ_{std}	0,0891	0,0980	-	0,6592	1,4117
\bar{E}	0,0509	0,0500	0,0378	0,0443	0,3164
E_{\min}	0,0462	0,0439	0,0331	0,0402	0,0237
E_{\max}	0,0548	0,0558	0,0445	0,0483	8,4034
E_{std}	0,0019	0,0032	0,0028	0,0019	1,2676
k_{\min}	7	8	6	-	-
k_{\max}	8	17	6	-	-
t_{GSVD}	-	-	-	140,545	140,545
\bar{t}	0,1448	0,2024	0,1318	0,2320	0,1583
t_{\min}	0,1230	0,1400	0,1150	0,1970	0,1390
t_{\max}	0,2540	0,4610	0,1710	0,6570	0,2480
t_{std}	0,0206	0,0583	0,0114	0,0757	0,0203
t_{total}	0,1448	0,2024	0,1318	140,777	140,703

Tabela 5.6: Resultados obtidos para o problema *Gravity* após 50 execuções com 1% de ruído nos dados e usando a matriz L_1 .

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	3,3539e+3	3,2376e+3	-	2,4193e+3	0,5311e+3
λ_{\min}	3,3132e+3	2,9586e+3	-	2,0025e+3	0,0001e+3
λ_{\max}	3,3833e+3	3,3762e+3	-	2,6750e+3	1,0432e+3
λ_{std}	17,934	143,4748	-	128,2891	300,7406
\bar{E}	0,0216	0,0273	0,0080	0,0169	0,2649
E_{\min}	0,0165	0,0195	0,0056	0,0124	0,0049
E_{\max}	0,0241	0,0457	0,0157	0,0194	8,4548
E_{std}	0,0014	0,0078	0,0021	0,0013	1,2746
k_{\min}	8	8	4	-	-
k_{\max}	8	29	4	-	-
t_{GSVD}	-	-	-	127,281	127,281
\bar{t}	0,1605	0,3610	0,1031	0,2514	0,1694
t_{\min}	0,1400	0,1610	0,0870	0,2070	0,1450
t_{\max}	0,2660	0,6310	0,1420	0,6520	0,2860
t_{std}	0,0227	0,1389	0,0128	0,0631	0,0222
t_{total}	0,1605	0,3610	0,1031	127,532	127,450

Tabela 5.7: Resultados obtidos para o problema *Gravity* após 50 execuções com 1% de ruído nos dados e usando a matriz L_2 .

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	4,7969e+1	4,7795e+1	-	29746e+1	5,3675
λ_{\min}	4,6968e+1	4,6879e+1	-	2,4779e+1	0,0068
λ_{\max}	4,8622e+1	4,8555e+1	-	3,3285e+1	1,0617e+1
λ_{std}	0,3799	0,4129	-	1,5517	2,9234
\bar{E}	0,0828	0,0827	0,0510	0,0653	0,7264
E_{\min}	0,0730	0,0723	0,0437	0,0564	0,0292
E_{\max}	0,0874	0,0880	0,0613	0,0714	21,0073
E_{std}	0,0025	0,0030	0,0040	0,0027	3,1788
k_{\min}	8	7	5	-	-
k_{\max}	9	18	5	-	-
t_{GSVD}	-	-	-	140,545	140,545
\bar{t}	0,1558	0,2139	0,1095	0,2305	0,1581
t_{\min}	0,1370	0,1420	0,0970	0,1990	0,1400
t_{\max}	0,2260	0,3690	0,1440	0,5930	0,2540
t_{std}	0,0191	0,0592	0,0094	0,0555	0,0202
\bar{t}_{total}	0,1558	0,2139	0,1095	140,776	140,703

Tabela 5.8: Resultados obtidos para o problema *Gravity* após 50 execuções com 2,5% de ruído nos dados e usando a matriz L_1 .

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	1,0089e+4	9,9941e+3	-	4,0954e+3	1,0360e+3
λ_{\min}	9,7410e+3	9,2509e+3	-	2,9271e+3	0,0821
λ_{\max}	1,0483e+1	1,0359e+4	-	5,0015e+3	1,9899e+3
λ_{std}	165,3178	230,2465	-	429,6415	606,1526
\bar{E}	0,0585	0,0617	0,0236	0,0257	0,6548
E_{\min}	0,0509	0,0509	0,0084	0,0209	0,0071
E_{\max}	0,0684	0,0886	0,0341	0,0325	21,1370
E_{std}	0,0037	0,0093	0,0032	0,0024	3,1879
k_{\min}	11	6	3	-	-
k_{\max}	12	28	3	-	-
t_{GSVD}	-	-	-	127,281	127,281
\bar{t}	0,2179	0,4491	0,0820	0,2434	0,1678
t_{\min}	0,1900	0,1360	0,0720	0,2100	0,1450
t_{\max}	0,3080	0,6010	0,1260	0,6970	0,2600
t_{std}	0,0187	0,1239	0,0107	0,0670	0,0169
\bar{t}_{total}	0,2179	0,4491	0,0820	127,524	127,449

Tabela 5.9: Resultados obtidos para o problema *Gravity* após 50 execuções com 2,5% de ruído nos dados e usando a matriz L_2 .

5.1.2 Phillips

Esta é a equação estudada por Phillips [44]. Dada a função

$$\zeta(p) = \begin{cases} 1 + \cos\left(p\frac{\pi}{3}\right), & |p| < 3 \\ 0, & |p| \geq 3 \end{cases}, \quad (5.5)$$

o núcleo K , a função f e a função g são

$$\begin{aligned} K(x, y) &= \zeta(x - y) \\ f(y) &= \zeta(y) \\ g(x) &= (6 - |x|) \left(1 + \frac{1}{2} \cos\left(x\frac{\pi}{3}\right) + \frac{9}{2\pi} \operatorname{sen}\left(|x|\frac{\pi}{3}\right) \right) \end{aligned}$$

em que $[-6, 6]$ é o intervalo de integração.

Na Figura 5.3 temos a solução exata e na Figura 5.4 temos os coeficientes para a inspeção visual da condição discreta de Picard

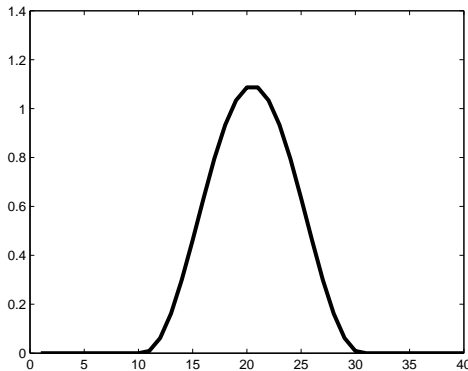


Figura 5.3: *Phillips*: Solução exata.

As tabelas 5.10 a 5.15 trazem os resultados obtidos com diferentes níveis de ruído e usando as matrizes L_1 e L_2 .

Tanto a escolha L_1 como a L_2 apresentaram resultados parecidos para todos os níveis de ruído, tendo L_1 erros um pouco

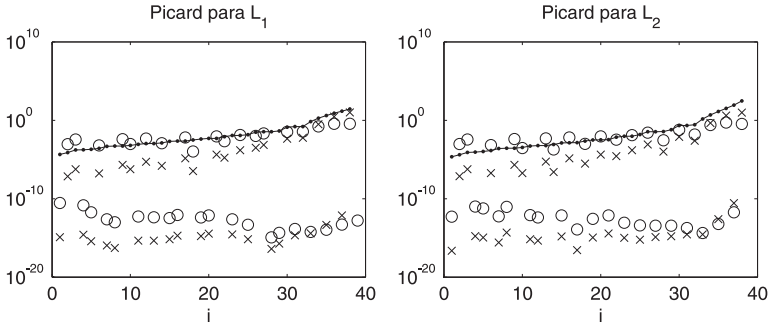


Figura 5.4: *Phillips*: Avaliação da CDP para (A, L_1) e (A, L_2) . Coeficientes $|\mathbf{u}_i^T \mathbf{b}|/\gamma_i$: o, Coeficientes de Fourier: x, Valores singulares: .

menores. Com exceção do método GGBK-FP para L_2 , os métodos apresentaram erros médios bastante bons, mantendo-se menores que 2% para qualquer nível de ruído para a escolha $L = L_1$, por exemplo.

Para ruído de 0,1% e L_1 , os erros médios para GGBK-FP e PROJ-FP foram menores que 1%, acompanhando o desempenho da Curva L e de GCV. O método G-LSQR apresentou erro um pouco superior, mas ainda razoável. Para o mesmo ruído e L_2 , PROJ-FP e G-LSQR mantiveram o bom desempenho.

Quando aumentamos o erro para 1%, os métodos avaliados (a menos do método GGBK-FP) apresentaram erros médios em torno de 2%, o que é bastante aceitável.

Quanto as dimensões dos subespaços projetados, PROJ-FP apresentou máximo de $k = 15$ e G-LSQR, de $k = 9$, sendo que o último método manteve a característica de alcançar o critério de parada em poucas iterações, como para o problema *Gravity*. Da mesma forma que no problema anterior, os desvios padrão para o erro foram pequenos, ficando na ordem de 10^{-3} . Apenas o E_{std} para o método GCV com 2,5% de ruído e L_2 foi de 0,0778.

Com 2,5% de ruído, os erros médios para L_1 mantiveram-se em torno de 2% para os métodos GGBK-FP, PROJ-FP, G-LSQR e Curva L. Para L_2 houve uma elevação dos erros médios para PROJ-FP e G-LSQR.

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	0,7148	0,7144	-	0,7104	0,8241
λ_{\min}	0,7076	0,7068	-	0,6396	0,1348
λ_{\max}	0,7219	0,7185	-	0,7925	1,0783
λ_{std}	0,0025	0,0023	-	0,0396	0,1934
\bar{E}	0,0082	0,0079	0,0223	0,0084	0,0092
E_{\min}	0,0047	0,0048	0,0080	0,0047	0,0052
E_{\max}	0,0120	0,0116	0,0226	0,0116	0,0380
E_{std}	0,0017	0,0016	0,0029	0,0017	0,0046
k_{\min}	8	12	5	-	-
k_{\max}	11	13	9	-	-
t_{GSVD}	-	-	-	111,5473	111,5473
\bar{t}	0,1975	0,2254	0,1496	0,2228	0,1518
t_{\min}	0,1400	0,1880	0,1350	0,1720	0,1250
t_{\max}	0,3440	0,3750	0,2070	0,5470	0,2180
t_{std}	0,0402	0,0340	0,0148	0,0502	0,0160
\bar{t}_{total}	0,1975	0,2254	0,1496	111,7701	111,6991

Tabela 5.10: Resultados obtidos para o problema *Phillips* após 50 execuções com 0,1% de ruído nos dados e usando a matriz L_1 .

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	6,3967e+4	6,1330e+1	-	4,5629e+1	2,5838e+1
λ_{\min}	6,3975e+4	6,1844e+1	-	5,3508e+1	3,6579e+1
λ_{\max}	6,3960e+4	6,0869e+1	-	3,4850e+1	6,5661
λ_{std}	3,3494	0,2164	-	3,7127	7,6486
\bar{E}	0,4772	0,0087	0,0174	0,0085	0,0091
E_{\min}	0,4772	0,0069	0,0084	0,0068	0,0054
E_{\max}	0,4772	0,0103	0,0266	0,0105	0,0168
E_{std}	0,0058	0,0007	0,0089	0,0007	0,0024
k_{\min}	34	12	5	-	-
k_{\max}	34	15	8	-	-
t_{GSVD}	-	-	-	96,328	96,328
\bar{t}	0,6993	0,2488	0,1460	0,2235	0,1513
t_{\min}	0,5860	0,2020	0,1110	0,196	0,1250
t_{\max}	2,8010	0,4090	0,2290	0,5810	0,2310
t_{std}	0,3108	0,0416	0,0323	0,0587	0,0247
\bar{t}_{total}	0,6993	0,2488	0,1460	96,552	96,479

Tabela 5.11: Resultados obtidos para o problema *Phillips* após 50 execuções com 0,1% de ruído nos dados e usando a matriz L_2 .

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	7,2314	7,2308	-	5,8324	3,4764
λ_{\min}	7,1733	7,1733	-	4,8187	0,4769
λ_{\max}	7,2713	7,2703	-	6,5816	5,2605
λ_{std}	0,0222	0,0222	-	0,4515	1,2918
\bar{E}	0,0200	0,0207	0,0217	0,0191	0,0297
E_{\min}	0,0142	0,0143	0,0172	0,0133	0,0125
E_{\max}	0,0259	0,0274	0,0309	0,0273	0,1341
E_{std}	0,0027	0,0032	0,0025	0,0033	0,0260
k_{\min}	7	6	6	-	-
k_{\max}	9	10	8	-	-
t_{GSVD}	-	-	-	111,5473	111,5473
\bar{t}	0,1460	0,1721	0,1290	0,2200	0,1554
t_{\min}	0,1090	0,0940	0,1070	0,1870	0,1250
t_{\max}	0,2030	0,3120	0,2230	0,5780	0,2190
t_{std}	0,0186	0,0308	0,0235	0,0531	0,0166
\bar{t}_{total}	0,1460	0,1721	0,1290	111,7673	111,7027

Tabela 5.12: Resultados obtidos para o problema *Phillips* após 50 execuções com 1% de ruído nos dados e usando a matriz L_1 .

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	6,4032e+4	0,0639e+4	-	0,0408e+4	0,0146e+4
λ_{\min}	6,3967e+4	0,0634e+4	-	0,0326e+4	0,0008e+4
λ_{\max}	6,4123e+4	0,0645e+4	-	0,0459e+4	0,0280e+4
λ_{std}	33,3053	2,8791	-	32,9215	80,2945
\bar{E}	0,4773	0,0253	0,0264	0,0244	0,0340
E_{\min}	0,4771	0,0236	0,0156	0,0220	0,0146
E_{\max}	0,4774	0,0269	0,0444	0,0277	0,1562
E_{std}	0,0001	0,0007	0,0043	0,001	0,0297
k_{\min}	34	6	5	-	-
k_{\max}	34	10	7	-	-
t_{GSVD}	-	-	-	96,328	96,328
\bar{t}	0,6847	0,1616	0,1135	0,2280	0,1536
t_{\min}	0,5980	0,1170	0,0930	0,2000	0,1380
t_{\max}	1,1730	0,2800	0,1517	0,4580	0,2020
t_{std}	0,0778	0,0226	0,0132	0,0348	0,0102
\bar{t}_{total}	0,6847	0,1616	0,1135	96,556	96,482

Tabela 5.13: Resultados obtidos para o problema *Phillips* após 50 execuções com 1% de ruído nos dados e usando a matriz L_2 .

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	1,8440e+1	1,8439e+1	-	1,1164e+1	5,6616
λ_{\min}	1,8245e+1	1,8245e+1	-	9,2087	0,4785
λ_{\max}	1,8590e+1	1,8589e+1	-	1,2680e+1	7,9574
λ_{std}	0,0794	0,0794	-	0,7595	2,0103
\bar{E}	0,0282	0,0292	0,0258	0,0246	0,0505
E_{\min}	0,0231	0,0227	0,0209	0,0174	0,0154
E_{\max}	0,0339	0,0345	0,0685	0,0335	0,3336
E_{std}	0,0025	0,0025	0,0088	0,0036	0,0682
k_{\min}	7	6	6	-	-
k_{\max}	8	9	7	-	-
t_{GSVD}	-	-	-	111,5473	111,5473
\bar{t}	0,1369	0,1565	0,1202	0,2150	0,1447
t_{\min}	0,1090	0,1090	0,1060	0,1870	0,1250
t_{\max}	0,1880	0,2810	0,1640	0,5780	0,2030
t_{std}	0,0136	0,0269	0,0121	0,0544	0,0154
\bar{t}_{total}	0,1369	0,1565	0,1202	111,7623	111,692

Tabela 5.14: Resultados obtidos para o problema *Phillips* após 50 execuções com 2, 5% de ruído nos dados e usando a matriz L_1 .

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	6,4363e+4	1,7638e+3	-	7,3682e+2	2,7554e+2
λ_{\min}	6,4204e+4	1,7349e+3	-	6,2374e+2	8,0841e+1
λ_{\max}	6,4583e+4	1,7931e+3	-	8,2355e+2	4,1633e+2
λ_{std}	81,1269 0	11,8121	-	53,1354	122,5657
\bar{E}	0,4776	0,0465	0,0284	0,0280	0,0556
E_{\min}	0,4772	0,0405	0,0209	0,0241	0,0200
E_{\max}	0,4779	0,0547	0,0645	0,0332	0,3949
E_{std}	0,0001	0,0028	0,0061	0,0021	0,0778
k_{\min}	34	6	5	-	-
k_{\max}	34	14	6	-	-
t_{GSVD}	-	-	-	96,328	96,328
\bar{t}	0,6839	0,1543	0,1116	0,2309	0,1555
t_{\min}	0,6020	0,1170	0,0930	0,2010	0,1360
t_{\max}	1,1960	0,2520	0,1590	0,4810	0,2470
t_{std}	0,0800	0,0317	0,0136	0,0377	0,0151
\bar{t}_{total}	0,6839	0,1543	0,1116	96,559	96,484

Tabela 5.15: Resultados obtidos para o problema *Phillips* após 50 execuções com 2, 5% de ruído nos dados e usando a matriz L_2 .

5.1.3 *Deriv2*

O problema *Deriv2* envolve o operador segunda derivada. O núcleo K desta equação integral de Fredholm de primeira espécie é a função de Green para a segunda derivada

$$K(x, y) = \begin{cases} x(y - 1), & x < y \\ y(x - 1), & x \geq y \end{cases}$$

e o intervalo de integração é $[0, 1]$. Neste problema dispomos de três exemplos para a função g e a solução f . São eles:

Exemplo 1 $g(x) = \frac{(x^3 - x)}{6}$ e $f(y) = y$

Exemplo 2 $g(x) = e^x + (1 - e)x - 1$ e $f(y) = e^y$

Exemplo 3

$$g(x) = \begin{cases} \frac{4x^3 - 3x}{24}, & x < 0,5 \\ \frac{-4x^3 + 12x^2 - 9x + 1}{24}, & x \geq 0,5 \end{cases}$$
$$f(y) = \begin{cases} y, & y < 0,5 \\ 1 - y, & y \geq 0,5 \end{cases}$$

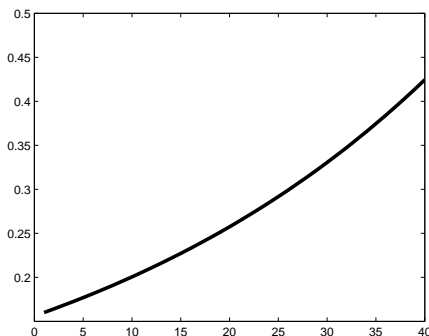


Figura 5.5: *Deriv2* exemplo 2: Solução exata.

Na Figura 5.5 temos a solução exata para o exemplo 2 e na Figura 5.6 temos os coeficientes para a inspeção visual da condição discreta de Picard.

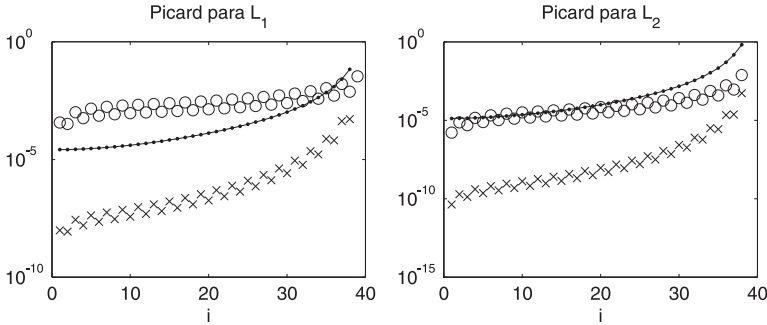


Figura 5.6: *Deriv2* exemplo 2: Avaliação da CDP para (A, L_1) e (A, L_2) . Coeficientes $|\mathbf{u}_i^T \mathbf{b}|/\gamma_i$: \circ , Coeficientes de Fourier: \times , Valores singulares: \cdot .

As tabelas 5.16 a 5.21 trazem os resultados obtidos usando o exemplo 2, com diferentes níveis de ruído e usando as matrizes L_1 e L_2 .

Para o exemplo 2 deste problema, o método PROJ-FP não alcançou bons resultados. No entanto, os métodos GGKB-FP e G-LSQR apresentaram erros médios menores que 1,7% para 0,1% de ruído, tanto para L_1 quanto para L_2 .

Para L_1 e 1% de ruído, G-LSQR apresentou o erro médio de aproximadamente 3%, mantendo o índice para 2,5% de ruído. Com 2,5% e L_2 , os erros médios para os métodos GGKB-FP, G-LSQR, Curva L e GCV foram inferiores a 3,6%, destacando que o erro do método GGKB-FP foi de 1,5%.

Novamente, os desvios padrão foram pequenos em todos os casos. Quanto aos parâmetros de regularização obtidos para L_2 foram muito maiores do que os parâmetros calculados para L_1 . Os erros médios obtidos com $L = L_2$ foram melhores do que com L_1 para todos os níveis de ruído, não ultrapassando 3% tanto com GGKB-FP quanto com G-LSQR.

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	0,1002	0,0128	-	0,0771	0,0256
λ_{\min}	0,0988	0,0125	-	0,0689	0,0048
λ_{\max}	0,1013	0,0130	-	0,0848	0,0383
λ_{std}	0,0005	0,0001	-	0,0038	0,0077
\bar{E}	0,0176	0,1831	0,0120	0,0156	0,0113
E_{\min}	0,0161	0,1807	0,0100	0,0141	0,0076
E_{\max}	0,0191	0,1855	0,0151	0,0170	0,0235
E_{std}	0,0006	0,0009	0,0012	0,0007	0,0025
k_{\min}	7	11	6	-	-
k_{\max}	7	16	6	-	-
t_{GSVD}	-	-	-	160,808	160,808
\bar{t}	0,1307	0,2739	0,1074	0,2215	0,1534
t_{\min}	0,1190	0,2030	0,1050	0,1880	0,1350
t_{\max}	0,1990	0,4430	0,1300	0,6520	0,2460
t_{std}	0,0118	0,0305	0,0039	0,0631	0,0165
\bar{t}_{total}	0,1307	0,2739	0,1074	161,030	160,961

Tabela 5.16: Resultados obtidos para o problema *Deriv2 exemplo 2* após 50 execuções com 0,1% de ruído nos dados e L_1

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	1,5517e+2	1,3245	-	7,6932e+1	1,1697e+1
λ_{\min}	1,4890e+2	1,1894	-	6,3914e+1	0,2950
λ_{\max}	1,6161e+2	1,5003	-	8,5331e+1	2,1370e+1
λ_{std}	2,5309	0,0626	-	4,8195	5,5240
\bar{E}	0,0089	0,2243	0,0021	0,0059	0,0034
E_{\min}	0,0083	0,2213	0,0014	0,0056	0,0013
E_{\max}	0,0098	0,2274	0,0040	0,0066	0,0196
E_{std}	0,0003	0,0012	0,0006	0,0002	0,0031
k_{\min}	11	12	3	-	-
k_{\max}	13	13	3	-	-
t_{GSVD}	-	-	-	178,9220	178,9220
\bar{t}	0,2023	0,2253	0,0974	0,2241	0,1542
t_{\min}	0,1740	0,1980	0,0800	0,1940	0,1370
t_{\max}	0,2490	0,3990	0,1280	0,5700	0,2150
t_{std}	0,0133	0,0290	0,0085	0,0513	0,0111
\bar{t}_{total}	0,2023	0,2253	0,0974	179,1461	179,0762

Tabela 5.17: Resultados obtidos para o problema *Deriv2 exemplo 2* após 50 execuções com 0,1% de ruído nos dados e L_2 .

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	1,2110	0,2145	-	0,9252	0,1452
λ_{\min}	1,1899	0,2068	-	0,8511	0,0113
λ_{\max}	1,2400	0,2188	-	1,0185	0,2492
λ_{std}	0,0102	0,0025	-	0,0361	0,0539
\bar{E}	0,0575	0,2936	0,0293	0,0515	0,0320
E_{\min}	0,0550	0,2896	0,0263	0,0494	0,0179
E_{\max}	0,0605	0,2968	0,0355	0,0539	0,1495
E_{std}	0,0012	0,0016	0,0022	0,0010	0,0225
k_{\min}	8	9	3	-	-
k_{\max}	8	9	3	-	-
t_{GSVD}	-	-	-	160,808	160,808
\bar{t}	0,1457	0,1716	0,0674	0,2206	0,1538
t_{\min}	0,1340	0,1520	0,0640	0,1910	0,1350
t_{\max}	0,2260	0,3400	0,0970	0,5790	0,2170
t_{std}	0,0127	0,0258	0,0057	0,0528	0,0127
\bar{t}_{total}	0,1457	0,1716	0,0674	161,028	160,962

Tabela 5.18: Resultados obtidos para o problema *Deriv2 exemplo 2* após 50 execuções com 1% de ruído nos dados e L_1 .

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	2,6180e+2	4,6095e+1	-	1,3940e+2	6,8033e+1
λ_{\min}	2,5188e+2	4,2652e+1	-	9,3696e+2	0,2992
λ_{\max}	2,6529e+2	4,8742e+1	-	1,9107e+2	1,1577e+2
λ_{std}	2,2603	1,5327	-	22,3822	38,1000
\bar{E}	0,0142	0,3484	0,0134	0,0089	0,0162
E_{\min}	0,0102	0,3459	0,0031	0,0050	0,0033
E_{\max}	0,0198	0,3505	0,0293	0,0148	0,1310
E_{std}	0,0019	0,0009	0,0061	0,0022	0,0228
k_{\min}	28	7	3	-	-
k_{\max}	29	7	3	-	-
t_{GSVD}	-	-	-	178,9220	178,9220
\bar{t}	0,6354	0,1368	0,0977	0,2285	0,1497
t_{\min}	0,5260	0,1200	0,0800	0,1940	0,1260
t_{\max}	2,5200	0,2490	0,1440	0,4640	0,2340
t_{std}	0,2753	0,0185	0,0102	0,0400	0,0180
\bar{t}_{total}	0,6354	0,1368	0,0977	179,1505	179,0717

Tabela 5.19: Resultados obtidos para o problema *Deriv2 exemplo 2* após 50 execuções com 1% de ruído nos dados e L_2 .

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	0,1002	0,0128	-	0,0771	0,0256
λ_{\min}	0,0988	0,0125	-	0,0689	0,0048
λ_{\max}	0,1013	0,0130	-	0,0848	0,0383
λ_{std}	0,0005	0,0001	-	0,0038	0,0077
\bar{E}	0,0176	0,1831	0,0123	0,0156	0,0113
E_{\min}	0,0161	0,1807	0,0100	0,0141	0,0076
E_{\max}	0,0191	0,1855	0,0151	0,0170	0,0235
E_{std}	0,0006	0,0009	0,0012	0,0007	0,0025
k_{\min}	7	11	6	-	-
k_{\max}	7	16	6	-	-
t_{GSVD}	-	-	-	160,808	160,808
\bar{t}	0,1307	0,2739	0,1123	0,2215	0,1534
t_{\min}	0,1190	0,2030	0,1040	0,1880	0,1350
t_{\max}	0,1990	0,4430	0,1321	0,6520	0,2460
t_{std}	0,0118	0,0305	0,0175	0,0631	0,0165
\bar{t}_{total}	0,1307	0,2739	0,1123	161,030	160,961

Tabela 5.20: Resultados obtidos para o problema *Deriv2 exemplo 2* após 50 execuções com 2,5% de ruído nos dados e L_1 .

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	2,5833e+2	4,2040e+2	-	1,8102e+2	1,1958e+2
λ_{\min}	2,1034e+2	3,9145e+2	-	1,1625e+2	0,2992
λ_{\max}	2,6787e+2	4,4083e+2	-	2,3460e+2	3,5755e+2
λ_{std}	1,3473e+1	9,6815	-	2,8924e+1	7,4546e+1
\bar{E}	0,0148	0,4680	0,0350	0,0121	0,0299
E_{\min}	0,0070	0,4655	0,0116	0,0050	0,0050
E_{\max}	0,0289	0,4704	0,1163	0,0273	0,3273
E_{std}	0,0044	0,0009	0,0187	0,0048	0,0496
k_{\min}	28	6	3	-	-
k_{\max}	35	6	3	-	-
t_{GSVD}	-	-	-	178,9220	178,9220
\bar{t}	0,6062	0,1199	0,0808	0,2283	0,1553
t_{\min}	0,5100	0,1060	0,0730	0,1990	0,1360
t_{\max}	1,1380	0,2320	0,0960	0,4730	0,2000
t_{std}	0,0844	0,0184	0,0030	0,0382	0,0111
\bar{t}_{total}	0,6062	0,1199	0,0808	179,1503	1779,0773

Tabela 5.21: Resultados obtidos para o problema *Deriv2 exemplo 2* após 50 execuções com 2,5% de ruído nos dados e L_2 .

A Figura 5.7 apresenta a solução exata para o exemplo 3 e na Figura 5.8, os coeficientes para observação da condição discreta de Picard. Note que, mesmo com os coeficientes de Fourier (\times) oscilando entre 10^{-20} e 10^{-5} , eles estão sempre abaixo dos valores singulares generalizados (\circ), tanto para L_1 quanto para L_2 .

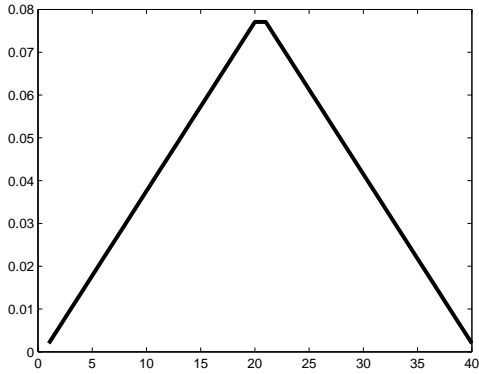


Figura 5.7: *Deriv2* exemplo 3: Solução exata.

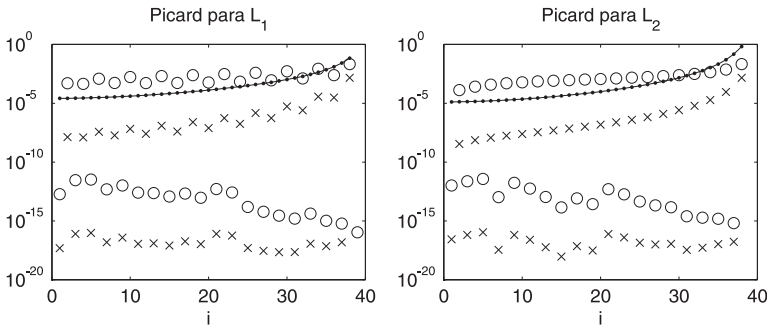


Figura 5.8: *Deriv2* exemplo 3: Avaliação da CDP para (A, L_1) (esquerda) e (A, L_2) (direita). Coeficientes $|\mathbf{u}_i^T \mathbf{b}|/\gamma_i$: \circ , Coeficientes de Fourier: \times , Valores singulares: \cdot .

As tabelas 5.22 a 5.27 trazem resultados análogos para o exemplo 3 da rotina, também para L_1 e L_2 . Note que o exemplo 3 apresenta um ponto sem derivada.

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	0,0327	0,0304	-	0,0263	0,0097
λ_{\min}	0,0322	0,0301	-	0,0236	0,0026
λ_{\max}	0,0329	0,0305	-	0,0281	0,0156
λ_{std}	0,0001	0,0001	-	0,0008	0,0028
\bar{E}	0,0333	0,0119	0,0256	0,0300	0,0233
E_{\min}	0,0296	0,0092	0,0205	0,0264	0,0160
E_{\max}	0,0371	0,0147	0,0327	0,0331	0,0516
E_{std}	0,0015	0,0012	0,0030	0,0017	0,0060
k_{\min}	7	8	6	-	-
k_{\max}	10	13	8	-	-
t_{GSVD}	-	-	-	143,279	143,279
\bar{t}	0,1591	0,2065	0,1247	0,2263	0,1468
t_{\min}	0,1250	0,1560	0,1070	0,2030	0,1400
t_{\max}	0,2340	0,4060	0,1440	0,5630	0,2180
t_{std}	0,0156	0,0423	0,0081	0,0494	0,0125
t_{total}	0,1591	0,2065	0,1247	143,5053	143,4258

Tabela 5.22: Resultados obtidos para o problema *Deriv2 exemplo3* após 50 execuções com 0,1% de ruído nos dados e L_1 .

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	6,9337	6,8757	-	5,1398	0,5080
λ_{\min}	6,7942	6,7287	-	4,0908	0,0400
λ_{\max}	7,0482	6,9981	-	5,6145	0,8813
λ_{std}	0,0644	0,0697	-	0,3135	0,2075
\bar{E}	0,0236	0,0230	0,0215	0,0208	0,0148
E_{\min}	0,0223	0,0223	0,0145	0,0193	0,0095
E_{\max}	0,0246	0,0236	0,0241	0,0222	0,0693
E_{std}	0,0004	0,0003	0,0014	0,0006	0,0088
k_{\min}	7	8	4	-	-
k_{\max}	8	9	5	-	-
t_{GSVD}	-	-	-	136,8340	136,8340
\bar{t}	0,1337	0,1804	0,0850	0,2283	0,1534
t_{\min}	0,1250	0,1560	0,0780	0,2030	0,1400
t_{\max}	0,2030	0,3900	0,1260	0,6250	0,2500
t_{std}	0,0155	0,0351	0,0095	0,0578	0,0214
t_{total}	0,1337	0,1804	0,0850	137,0623	136,9874

Tabela 5.23: Resultados obtidos para o problema *Deriv2 exemplo3* após 50 execuções com 0,1% de ruído nos dados e L_2 .

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	0,3756	0,3160	-	0,2362	0,0561
λ_{\min}	0,3665	0,3134	-	0,2114	0,0119
λ_{\max}	0,3814	0,3176	-	0,2649	0,0939
λ_{std}	0,0029	0,0009	-	0,0107	0,0158
\bar{E}	0,1029	0,0359	0,08013	0,0869	0,0579
E_{\min}	0,0968	0,0315	0,0683	0,0797	0,0239
E_{\max}	0,1088	0,0421	0,0904	0,0938	0,1780
E_{std}	0,0029	0,0026	0,0049	0,0034	0,0234
k_{\min}	8	6	3	-	-
k_{\max}	9	8	4	-	-
t_{GSVD}	-	-	-	143,2790	143,2790
\bar{t}	0,1428	0,1541	0,0769	0,2260	0,1484
t_{\min}	0,1250	0,1250	0,0640	0,2030	0,1250
t_{\max}	0,2030	0,3280	0,1160	0,5630	0,2180
t_{std}	0,0130	0,0286	0,0129	0,0499	0,0144
\bar{t}_{total}	0,0130	0,0286	0,0769	143,5050	143,4274

Tabela 5.24: Resultados obtidos para o problema *Deriv2 exemplo3* após 50 execuções com 1% de ruído nos dados e L_1 .

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	9,6040e+1	1,0040e+2	-	7,2075e+2	4,9302
λ_{\min}	9,3390e+1	9,8357e+1	-	6,62912e+1	0,2949
λ_{\max}	9,7784e+1	1,0179e+1	-	7,6227e+1	9,4140
λ_{std}	1,1004	0,8633	-	2,3750	2,4817
\bar{E}	0,0734	0,0618	0,0411	0,0713	0,0432
E_{\min}	0,0712	0,0581	0,0341	0,0682	0,0213
E_{\max}	0,0764	0,0643	0,0533	0,0755	0,2296
E_{std}	0,0013	0,0012	0,0049	0,0017	0,0350
k_{\min}	8	6	3	-	-
k_{\max}	9	6	3	-	-
t_{GSVD}	-	-	-	136,8340	136,834s
\bar{t}	0,1557	0,1443	0,0667	0,2297	0,1509
t_{\min}	0,1400	0,1250	0,0640	0,2180	0,1400
t_{\max}	0,2190	0,3130	0,0890	0,5780	0,2180
t_{std}	0,0138	0,0306	0,0042	0,0517	0,0151
\bar{t}_{total}	0,1557	0,1443	0,0667	137,0637	136,9849

Tabela 5.25: Resultados obtidos para o problema *Deriv2 exemplo3* após 50 execuções com 1% de ruído nos dados e L_2 .

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	0,7790	0,8146	-	0,3580	0,1222
λ_{\min}	0,7675	0,8074	-	0,2901	0,0117
λ_{\max}	0,7863	0,8183	-	0,4237	0,2047
λ_{std}	0,0039	0,0023	-	0,0315	0,0486
\bar{E}	0,1543	0,0566	0,0950	0,1023	0,0919
E_{\min}	0,1403	0,0492	0,0741	0,0910	0,0401
E_{\max}	0,1658	0,0631	0,1124	0,1166	0,4400
E_{std}	0,0055	0,0031	0,0094	0,0059	0,0615
k_{\min}	28	6	3	-	-
k_{\max}	28	7	3	-	-
t_{GSVD}	-	-	-	143,2790	143,2790
\bar{t}	0,5585	0,1328	0,0824	0,2191	0,1406
t_{\min}	0,5000	0,1100	0,0720	0,1870	0,1250
t_{\max}	1,2660	0,2650	0,1120	0,4680	0,2040
t_{std}	0,1090	0,0209	0,0066	0,0385	0,0135
\bar{t}_{total}	0,5585	0,1328	0,0824	143,4981	143,4196

Tabela 5.26: Resultados obtidos para o problema *Deriv2 exemplo3* após 50 execuções com 2,5% de ruído nos dados e L_1 .

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	1,9263e+2	2,8655e+2	-	9,7969e+1	1,2514e+1
λ_{\min}	1,8992e+2	2,8207e+2	-	7,8841e+1	0,2959
λ_{\max}	1,9419e+2	2,9224e+2	-	1,1014e+2	3,2865e+1
λ_{std}	0,9471	2,2377	-	7,5605	7,1262
\bar{E}	0,0913	0,1027	0,0512	0,0748	0,0759
E_{\min}	0,0808	0,1004	0,0341	0,0705	0,0307
E_{\max}	0,0996	0,1043	0,0867	0,0844	0,5730
E_{std}	0,0041	0,0008	0,0123	0,0034	0,0884
k_{\min}	28	6	3	-	-
k_{\max}	28	6	3	-	-
t_{GSVD}	-	-	-	136,8340	136,8340
\bar{t}	0,5433	0,1322	0,0810	0,2189	0,1491
t_{\min}	0,4930	0,1230	0,0700	0,2030	0,1310
t_{\max}	1,0790	0,2580	0,1110	0,4560	0,2390
t_{std}	0,0916	0,0228	0,0062	0,0377	0,0282
\bar{t}_{total}	0,5433	0,1322	0,0810	137,0529	136,9831

Tabela 5.27: Resultados obtidos para o problema *Deriv2 exemplo3* após 50 execuções com 2,5% de ruído nos dados e L_2 .

No exemplo 3 do problema *Deriv*, para ruído de 0,1% nos dados, todos os métodos tiveram desempenhos semelhantes, com erros médios de 3%, tanto para L_1 quanto para L_2 . Note que os parâmetros obtidos quando usamos L_1 são relativamente menores do que os parâmetros calculados com L_2 , mas isso não influenciou na qualidade das soluções.

Os desvios padrão, para este e os outros níveis de ruído, seguem pequenos, como nos problemas anteriores. Também, os tempos de execução foram todos menores do que 1 segundo e a dimensão dos subespaços pequena, com destaque para o método G-LSQR que obteve bons resultados em todos os casos e não utilizou mais do que 0,8% da dimensão original do problema.

Com ruídos de 1% e 2,5% para L_1 , o método GGKB-FP apresentou erros mais significativos que os demais. No entanto, PROJ-FP e G-LSQR nestes casos mostraram-se melhores ou equivalentes aos métodos da Curva L e GCV, com vantagem para o tempo de execução, já que eles não necessitam do cálculo da GSVD de (A, L_1) . O mesmo ocorre para o caso $L = L_2$.

Destacamos, por fim, o método PROJ-FP que obteve os melhores resultados com L_1 e G-LSQR que alcançou mais sucesso com L_2

5.2 Tratamento de Imagens

Nesta seção apresentaremos problemas onde a solução desejada é uma imagem. Na prática, imagens digitais são obtidas através de algum dispositivo eletrônico tais como câmeras ou sensores, e são compostas por elementos chamados *pixels*. Em uma imagem em níveis de cinza, cada *pixel* assume um valor entre 0 e 255, que determina a intensidade de cinza apresentada naquele *pixel*. Imagens pequenas têm cerca de 256^2 *pixels*, mas outras podem chegar facilmente a 5 ou 10 milhões de *pixels*. No caso de uma imagem colorida, pode-se convertê-la em níveis de cinza sem grande dificuldade. Maiores detalhes podem ser obtidos em literaturas especializadas.

Os problemas aqui apresentados são o de super resolução e o

de reconstrução de imagens, ambos bastante comuns em problemas reais. Resolver tais problemas significa obter uma imagem de melhor qualidade – mais nítida, com mais detalhes, etc. – a partir de uma ou de um conjunto de imagens de qualidade inferior, seja pela interferência de algum fator externo ou pelas restrições dos sensores que as captaram.

Super Resolução de Imagens

Na super resolução desejamos melhorar a resolução de imagens capturadas através de algum dispositivo de aquisição. No exemplo apresentado, imagem *Tree*, desejamos obter uma imagem com mais pixels (HR - high resolution), que fornece mais informações, a partir de uma sequência de imagens com menos pixels (LR - low resolution) e com ruído.

Neste tipo de problema, assume-se que as imagens LH são altamente correlacionadas, no sentido que pequenas diferenças de informação presente em cada imagem capturada proporciona uma nova informação que permite a recuperação de detalhes da imagem HR.

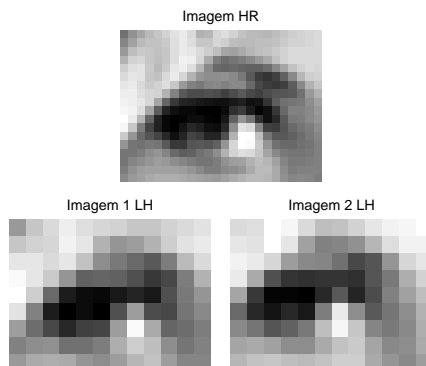


Figura 5.9: Imagem em alta resolução e duas em baixa resolução.

A Figura 5.9 mostra uma imagem original HR e duas imagens em resolução mais baixa. A imagem HR tem 18×24 pixels enquanto as imagens em baixa resolução têm 9×12 pixels cada.

Para descrever o problema abordado aqui, seja \mathbf{f} uma imagem HR de tamanho $M = M_1 \times M_2$ cuja representação usando a operação \mathbf{vec} descrita na definição 3.10 é $\mathbf{f} = [f_1, \dots, f_M]^T$, sendo M o número de pixels, e seja $\mathbf{g}_k = [\mathbf{g}_{k,1}, \mathbf{g}_{k,2}, \dots, \mathbf{g}_{k,N}]^T$, $k = 1, 2, \dots, p$ a k -ésima imagem LR de tamanho $N = N_1 \times N_2$, com $M = N_1 \times L_1$, $M_2 = N_2 \times L_2$, sendo L_1 e L_2 os fatores de subamostragem nas direções horizontal e vertical, respectivamente, do modelo que relaciona imagens HR e LR. Se assumimos que o processo de aquisição das imagens \mathbf{g}_k envolve embaçamento (*blurring*), movimento, subamostragem e ruído aditivo, um modelo linear, que relaciona a imagem HR \mathbf{f} com as imagens LR \mathbf{g}_k , pode ser escrito como

$$\mathbf{g}_k = A_k \mathbf{f} + \epsilon_k \quad (5.6)$$

com $A_k \in \mathbb{R}^{N \times M}$ e ϵ_k denotando um vetor de ruídos [43]. O problema de super resolução consiste na determinação de estimativas da imagem \mathbf{f} a partir da sequência de imagens \mathbf{g}_k . Devido ao fato dos sistemas em (5.6) serem subdeterminados, podemos encontrar infinitas soluções \mathbf{f} e assim o problema de super resolução é mal posto. Para contornar esta dificuldade, dentre vários métodos disponíveis na literatura, um deles é o método de regularização de Tikhonov, que neste contexto pode ser formulado como

$$\hat{\mathbf{f}}_\lambda = \underset{\mathbf{f}}{\operatorname{argmin}} \|\mathbf{g} - A\mathbf{f}\|_2^2 + \lambda^2 \|L\mathbf{f}\|_2^2 \quad (5.7)$$

sendo

$$\mathbf{g} = [\mathbf{g}_1^T \ \dots \ \mathbf{g}_p^T]^T, \quad A = [A_1^T \ \dots \ A_p^T]^T \in \mathbb{R}^{pN \times M} \quad (5.8)$$

e L uma versão discreta de algum operador diferencial nas direções horizontal e vertical, respectivamente. E uma vez mais o desafio é determinar um parâmetro de regularização apropriado.

5.2.1 Imagem *Tree*

A imagem exata é uma paisagem com uma árvore, como mostrado na Figura 5.10. A imagem tem tamanho 96×96 *pixels* e o objetivo é recuperar a imagem original, representado por $\mathbf{x}_{\text{exato}}$,

a partir de uma sequência de cinco imagens em resolução mais baixa, no caso, 48×48 *pixels*.

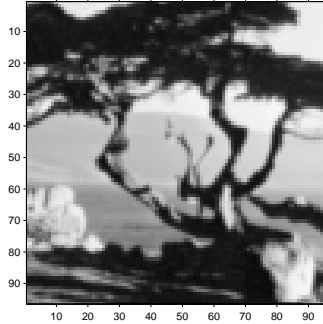


Figura 5.10: Imagem *Tree* exata e em alta resolução.

A informação das imagens em baixa resolução está contida no vetor \mathbf{b} que apresenta $5 \times 48^2 = 11520$ coordenadas. Para os testes consideramos o lado direito \mathbf{b} com um nível de ruído (NL) determinado pelo usuário, conforme mostrado no início do capítulo.

A Figura 5.11 traz quatro das cinco imagens em baixa resolução que compõem o vetor \mathbf{b} . De acordo com a discussão apresentada anteriormente para super-resolução de imagens, temos $M_1 = M_2 = 96$, $M = 9216$, $N_1 = N_2 = 48$, $N = 2304$, $p = 5$. Sendo assim, como a imagens que desejamos recuperar tem 9216 pixels, a matriz A do sistema tem dimensão 11520×9216 .

Como matriz L usamos a matriz

$$L_{\text{kron}} = \begin{bmatrix} I_N \otimes L_1 \\ L_1 \otimes I_N \end{bmatrix} \in \mathbb{R}^{18240 \times 9216} \quad (5.9)$$

com L_1 como definido em (5.1) para $N = 96$. Como L_{kron} tem mais linhas do que colunas tomamos sua decomposição QR como $L_{\text{kron}} = QR_{\text{kron}}$ em que $R_{\text{kron}} \in \mathbb{R}^{9215 \times 9216}$ e, na prática, usamos R_{kron} como matriz regularizante pois

$$\|L_{\text{kron}} \mathbf{x}\|_2 = \|QR_{\text{kron}} \mathbf{x}\|_2 = \|R_{\text{kron}} \mathbf{x}\|_2.$$

A matriz L_{kron} aplica regularização na horizontal e na vertical da imagem. Podemos pensar que cada pixel assume uma posição (i, j) na imagem I com $m \times n$ pixels e assume um valor entre 0 e 255. Assim, podemos pensar na imagem I como sendo uma função de duas variáveis. Seja $p_{i,j}$ o valor que o pixel da posição (i, j) assume. O que a matriz L_{kron} faz é, para cada j_0 fixo aplicar a matriz L_1 no vetor J_0 da forma

$$J_0 = [p_{1,j_0} \ p_{2,j_0} \ \dots \ p_{m,j_0}]$$

e para cada i_0 fixo aplicar a matriz L_1 no vetor I_0 da forma

$$I_0^T = [p_{i_0,1} \ p_{i_0,2} \ \dots \ p_{i_0,n}].$$

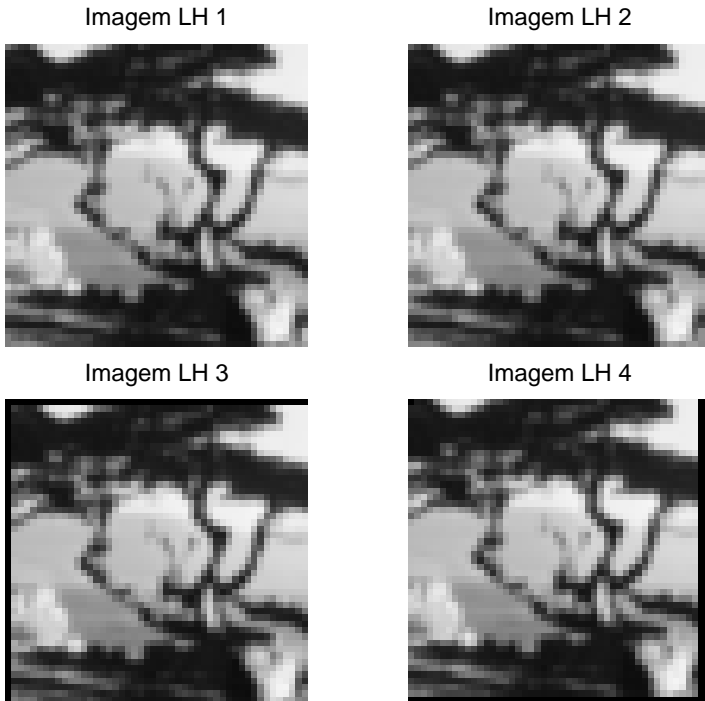


Figura 5.11: Sequências de imagens em baixa resolução, com 2304 *pixels* cada.

A seguir apresentamos os resultados obtidos pelos métodos apresentados no capítulo 3. Como para este problema, devido às dimensões, o cálculo da GSVD torna-se inviável, não foram apresentados outros métodos para comparação com GGKB-FP, PROJ-FP e G-LSQR. Também não foi possível, assim, verificar a condição discreta de Picard. No entanto, como os resultados podem ser visualizados na forma de imagens, isso permite que o leitor avalie o desempenho dos métodos.

Os métodos foram avaliados com ruído nos dados de 0, 1%, 1% e 2, 5%, sendo os resultados apresentados nas tabelas 5.28, 5.29 e 5.30, respectivamente. Como parâmetros iniciais utilizamos dimensão inicial $p = 13$, $\epsilon_1 = \epsilon_2 = 10^{-4}$ para os métodos GGKB-FP e PROJ-FP, e $\epsilon = 10^{-3}$ para G-LSQR.

	GGKB-FP	PROJ-FP	G-LSQR
$\bar{\lambda}$	0,0098	0,0026	-
λ_{\min}	0,0097	0,0025	-
λ_{\max}	0,0098	0,0026	-
λ_{std}	1,8720e-5	2,480e-5	-
\bar{E}	0,0391	0,0317	0,0390
E_{\min}	0,0391	0,0303	0,0390
E_{\max}	0,0392	0,0328	0,0391
E_{std}	2,1900e-5	0,5181e-3	0,0222e-3
k_{\min}	117	172	117
k_{\max}	117	189	117
\bar{t}	20,1165	40,9726	25,9022
t_{\min}	19,8150	37,6810	25,5200
t_{\max}	20,7530	47,8400	27,7150
t_{std}	0,2400	2,6353	0,3659

Tabela 5.28: Resultados obtidos para o problema *Tree* após 50 execuções com 0, 1% de ruído nos dados e usando a matriz L_{kron} .

De acordo com a Tabela 5.28, os erros médios obtidos pelos três métodos ficaram em torno de 3%, mantendo desvios padrão bastante pequenos, o que significa que o erro ficou próximo da média na maior parte das execuções. Isso pode também ser confirmado quando observamos os erros mínimos e máximos para GGKB-FP, PROJ-FP e G-LSQR, que também ficaram na faixa de 3%.

Note que os parâmetros encontrados foram pequenos, se comparado aos parâmetros para os problemas testes apresentados anteriormente. Este fato pode também ser observado para os resultados com 1% e 2,5% de ruído.

As dimensões dos subespaços projetados variaram de 117 a 189 para os métodos, equivalente a pouco mais de 2% da dimensão do problema original. Uma das características dos métodos de projeção é que quanto menor o erro nos dados, maior o número de iterações os métodos vão realizar. Com menos influência do ruído, pode-se executar mais iterações sem que se perca a qualidade na solução calculada.

A Figura 5.12 traz as imagens obtidas pelos métodos quando elevamos o ruído para 1% no vetor \mathbf{b} . Como no caso para 0,1%, os métodos foram executados 50 vezes com diferentes vetores \mathbf{e} e os resultados são apresentados na Tabela 5.29. A imagem exibida foi escolhida como sendo a solução com menor erro relativo, mas conforme indica a tabela, os erros nas 50 execuções foram muito próximos, para todos os métodos.

	GGKB-FP	PROJ-FP	G-LSQR
$\bar{\lambda}$	0,0365	0,0301	-
λ_{\min}	0,0360	0,0295	-
λ_{\max}	0,0374	0,0306	-
λ_{std}	0,0003	0,0002	-
\bar{E}	0,0461	0,0531	0,0478
E_{\min}	0,0455	0,0519	0,0474
E_{\max}	0,0469	0,0548	0,0484
E_{std}	0,0003	0,0005	0,0002
k_{\min}	100	39	89
k_{\max}	105	43	91
\bar{t}	17,7297	3,1046	19,3656
t_{\min}	16,4840	2,7650	18,0460
t_{\max}	19,6720	3,8160	21,3370
t_{std}	0,5575	0,1645	0,6844

Tabela 5.29: Resultados obtidos para o problema *Tree* após 50 execuções com 1% de ruído nos dados e usando a matriz L_{kron} .

Note que os erros relativos médios ficaram em torno de 4%, novamente com desvios padrão pequenos; e as imagens obtidas

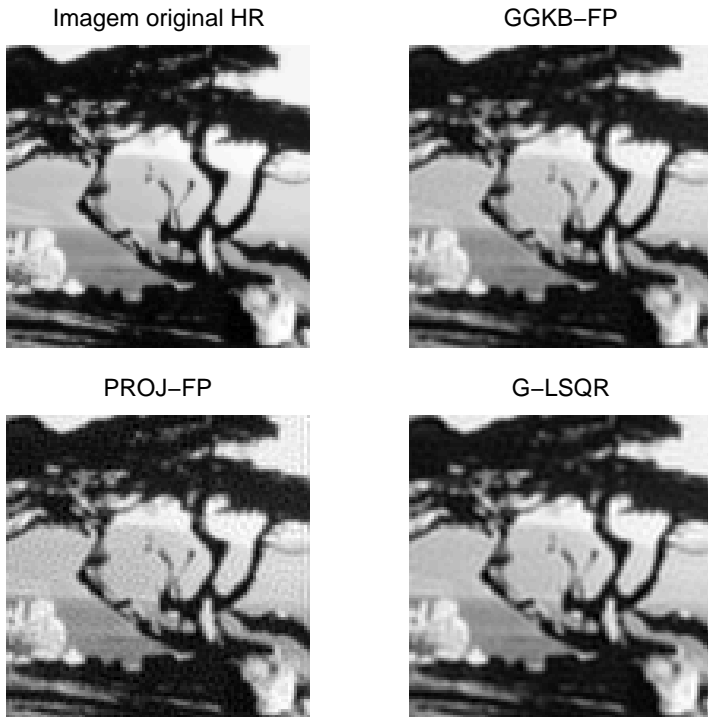


Figura 5.12: Soluções obtidas para o problema *Tree* com 1% de ruído.

ficaram bastante próximas à imagem original, mantendo o bom desempenho como no caso para 0,1% de ruído.

Para o método G-LSQR, vimos que o critério de parada é determinado tomando o mínimo da função $\bar{\Psi}$ como em (4.27). Também, o critério de parada é estabelecido de modo que o método não ande desnecessariamente. Na Figura 5.13 exibimos a função $\bar{\Psi}$ para o problema, obtida de uma das 50 execuções.

Quando o ruído no lado direito é de 2,5%, ambos os métodos novamente mostraram um desempenho bastante satisfatório. Os parâmetros de regularização, mesmo com o aumento do ruído, mantiveram-se pequenos. Os dados completos para este nível de ruído podem ser verificados na Tabela 5.30

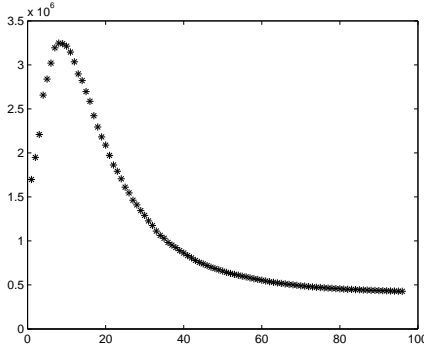


Figura 5.13: Função $\bar{\Psi}$ para o problema *Tree* com 1% de ruído.

	GGKB-FP	PROJ-FP	G-LSQR
$\bar{\lambda}$	0,0933	0,0823	0
λ_{\min}	0,0917	0,0809	0
λ_{\max}	0,0945	0,0835	0
λ_{std}	0,0006	0,0005	0
\bar{E}	0,0596	0,0709	0,0680
E_{\min}	0,0588	0,0697	0,0607
E_{\max}	0,0605	0,0723	0,2278
E_{std}	0,0004	0,0006	0,0329
k_{\min}	97	24	9
k_{\max}	98	24	68
t	15,9506	1,4577	13,0937
t_{\min}	15,0940	1,3910	12,3280
t_{\max}	21,3590	2,5940	16,4060
t_{std}	1,0275	0,1697	0,6417

Tabela 5.30: Resultados obtidos para o problema *Tree* após 50 execuções com 2,5% de ruído nos dados e usando a matriz L_{kron} .

Reconstrução de Imagens

O objetivo é obter uma imagem de melhor qualidade a partir de uma imagem que sofreu algum tipo de distorção (movimento, fora de foco, etc.). O problema *Rice*, apresentado a seguir, é exemplo de problemas desta natureza.

Consideremos um problema discreto mal posto proveniente da discretização de uma equação integral de primeira espécie em duas dimensões dada por

$$\int_0^1 \int_0^1 K(s, t; x, y) f(x, y) dx dy = g(s, t), \quad 0 \leq s, t \leq 1. \quad (5.10)$$

Na área de reconstrução de imagens, a função K é chamada de *point spread function* (PSF), f é uma imagem contínua (a “imagem exata”), e g é a imagem embaçada (*blurred image*) contínua. A determinação de estimativas de f a partir de uma imagem embaçada e contaminada por ruídos $g = \tilde{g} + e$, é conhecida como o problema de *deblurring*, que aqui chamamos de problema de reconstrução [32, 56].

Na prática, trabalha-se com imagens digitais que, informalmente falando, provém de amostragens de imagens contínuas e são, portanto, armazenadas em matrizes. Assim, se o problema (5.10) é discretizado usando alguma regra de quadratura, obtém-se um sistema de equações lineares

$$A\mathbf{f} = \mathbf{g} \quad \text{com} \quad \mathbf{f} = \mathbf{vec}(F) \quad \text{e} \quad \mathbf{g} = \mathbf{vec}(G), \quad (5.11)$$

sendo F e G as matrizes que contém as amostras de \mathbf{f} e \mathbf{g} respectivamente. Quando a PSF é do tipo $K(s, t; x, y) = K_1(s, x)K_2(t, y)$ (conhecido como o caso em que a PSF é separável), o problema discretizado torna-se

$$A_1 F A_2^T = G, \quad (5.12)$$

com $A_1 \in \mathbb{R}^{p \times m}$, $F \in \mathbb{R}^{m \times n}$, $A_2 \in \mathbb{R}^{r \times n}$ e $G \in \mathbb{R}^{p \times r}$. Usando em (5.12) a propriedade do produto de Kronecker dada em (3.63)

$$\mathbf{vec}(A_1 F A_2^T) = (A_2 \otimes A_1) \mathbf{vec}(F), \quad (5.13)$$

conclui-se que a matriz do sistema de equações lineares (5.11) é da forma

$$A = A_2 \otimes A_1. \quad (5.14)$$

Existem diversos fatores que podem fazer com que a imagem fique embaçada. O efeito *blur* pode ser, entre outros, devido à

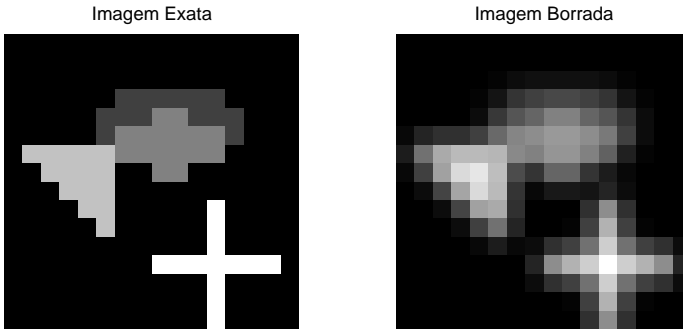


Figura 5.14: Imagem exata e imagem em que foi aplicado o efeito blur.

limitação em se representar as cores num sistema computacional ou devido a uma má qualidade das lentes que podem desfocar a imagem. A Figura 5.14 exibe uma imagem que sofreu um embaçamento.

Em alguns casos, a PSF pode ser obtida explicitamente por uma expressão matemática. Por exemplo, os elementos da PSF P , p_{ij} , para o *out-of-focus blur* (imagem fora de foco) são definidos por

$$p_{ij} = \begin{cases} \frac{1}{\pi r^2}, & se(i-k)^2 + (j-l)^2 \leq r^2 \\ 0, & \text{caso contrário} \end{cases} \quad (5.15)$$

em que (k, l) é o centro da PSF P e r o raio do embaçamento. A PSF para um embaçamento causado por turbulência atmosférica pode ser descrita como uma função Gaussiana bidimensional [32] e os elementos da PSF são dados por

$$p_{ij} = \exp\left(-\frac{1}{2} \begin{bmatrix} i-k \\ j-l \end{bmatrix}^T \begin{bmatrix} s_1^2 & \rho^2 \\ \rho^2 & s_2^2 \end{bmatrix}^{-1} \begin{bmatrix} i-k \\ j-l \end{bmatrix}\right), \quad (5.16)$$

em que os parâmetros s_1 , s_2 e ρ determinam a largura e a orientação da PSF, que é centrada no elemento (k, l) . Diferentes PSFs produzem diferentes matrizes e efeitos na imagem original. Para outras PSFs, ver [29].

Em [24] é disponibilizada a rotina *blur* que gera uma matriz A que modela o embaçamento de uma imagem $N \times N$, causado por turbulência atmosférica.

A PSF usada pela rotina é descrita pela função Gaussiana

$$h(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

em que o parâmetro σ controla a largura da PSF e, então, o nível de suavidade (quanto maior o valor de σ , mais espalhada será a função e mais mal posto o problema).

A matriz $A \in \mathbb{R}^{N^2 \times N^2}$ é simétrica e Toeplitz e pode ser escrita como

$$A = (2\pi\sigma^2)^{-1} T \otimes T, \quad (5.17)$$

onde $T \in \mathbb{R}^{N \times N}$ é uma matriz simétrica, banda e de Toeplitz, cuja primeira linha é

$$\mathbf{z} = [1 \quad e^{-\frac{1}{2\sigma^2}} \quad e^{-\frac{4}{2\sigma^2}} \quad e^{-\frac{9}{2\sigma^2}} \quad \dots \quad e^{-\frac{(\text{band}-1)}{2\sigma^2}} \quad 0 \dots 0] \in \mathbb{R}^N$$

em que o parâmetro **band** é a metade da largura de banda da matriz T . Os parâmetros padrão são **band**=3 e $\sigma = 0,7$.

A rotina também fornece os vetores \mathbf{x} e \mathbf{b} , em que o vetor \mathbf{x} contém os dados da imagem exata enquanto \mathbf{b} mantém os dados da imagem embaçada, isto é, $\mathbf{b} = A\mathbf{x}$; ambos na forma de um vetor de \mathbb{R}^{N^2} , obtido pela operação *vec*. Para visualizar as imagens, basta usar os comandos

```
imshow(reshape(x,N,N))
```

```
imshow(reshape(b,N,N))
```

5.2.2 Imagem *Rice*

Neste problema a imagem exata é a imagem de grãos de arroz sobre uma superfície plana como mostrado na Figura 5.15. A imagem tem tamanho 64×64 *pixels* e nosso objetivo é recuperar a imagem original, representada por $\mathbf{x}_{\text{exato}}$, a partir de uma imagem embaçada pelo sistema óptico usado para gravá-la. Em outras palavras, temos um problema de reconstrução de imagem. A

imagem exata rice.png está contida na *toolbox* Image Processing do MATLAB. A matriz A utilizada é a gerada pelo rotina *blur* para $N = 64$, $\text{band} = 16$ e $\sigma = 2$, o que nos fornece $c = (8\pi)^{-1}$. Neste caso, o número de condição da matriz A é $2 \cdot 10^{16}$.

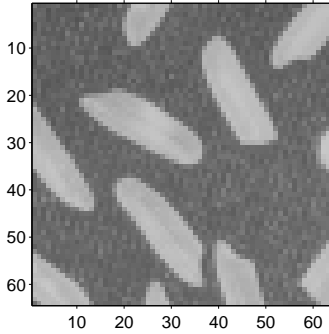


Figura 5.15: Imagem rice.png da *toolbox* Image Processing

A imagem embaçada sem ruído, denotada por $\mathbf{b}_{\text{exato}}$, é calculada como $\mathbf{b}_{\text{exato}} = A\mathbf{x}_{\text{exato}}$. A imagem embaçada e com ruído, representada por \mathbf{b} é obtida como descrito no início do capítulo, sendo que o nível de ruído (NL) é determinado pelo usuário.

Como matriz L usamos novamente a matriz

$$L_{\text{kron}} = \begin{bmatrix} I_N \otimes L_1 \\ L_1 \otimes I_N \end{bmatrix} \in \mathbb{R}^{8064 \times 4096}$$

com L_1 como definido na equação (5.1) para $N = 64$. Também, como para a imagem *Tree*, tomamos a fatoração QR de L_{kron} com $R_{\text{kron}} \in \mathbb{R}^{4096 \times 4096}$ e usamos R_{kron} como matriz regularizante.

No capítulo 3, mostramos que os métodos aqui propostos utilizam a matriz do sistema (A ou $\bar{A} = AL_A^\dagger$) apenas para produtos matriz-vetor envolvendo a matriz e sua transposta. Note que $A \in \mathbb{R}^{N^2 \times N^2}$ é um produto de Kronecker de matrizes de ordem $N \times N$, como descrito na equação (5.17). Assim, ao invés de montarmos a matriz A e em seguida usá-la para efetuar $\mathbf{z} = A\mathbf{x}$ ou $\bar{\mathbf{z}} = A^T\mathbf{x}$, utilizamos somente a matriz $T \in \mathbb{R}^{N \times N}$.

Para efetuar produtos da forma $\mathbf{y} = c(A_1 \otimes A_2)\mathbf{x}$, com c

<p>Entrada: A_1, A_2, \mathbf{x} e c.</p> <p>Saída: Vetor $\mathbf{y} = c(A_1 \otimes A_2) \mathbf{x}$</p> <ol style="list-style-type: none"> 1. Determina dimensão das matrizes A_1 e A_2 como $m_1 \times n_1$ e $m_2 \times n_2$, respectivamente. 2. Para i de 1 até m_1 faça $Y = \text{zeros}(m_2, 1)$; <ol style="list-style-type: none"> 2.1. Para j de 1 até n_1 faça Se $A_1(i, j) = 0$ faça $a = \text{zeros}(m_2, 1)$; Senão faça $a = A_1(i, j) * (A_2 * (\mathbf{x}(n_2 * (j - 1) + 1 : n_2 * j)))$ Fim Se $Y = Y + a$; Fim Para $\mathbf{y}(m_2 * (i - 1) + 1 : m_2 * i, 1) = Y$; Fim Para 3. $\mathbf{y} = c * \mathbf{y}$;

Tabela 5.31: Algoritmo *mkron*

constante, inclusive o produto $T \otimes T$ descrito anteriormente, usamos a rotina *mkron*, mostrada na tabela 5.31.

Para executar os métodos propostos na capítulo 3, são necessários vários produtos envolvendo a matriz dos coeficientes A , devido ao processo de Bidiagonalização de Golub-Kahan. A estratégia de utilizar a matriz T para este problema, trouxe uma redução de mais de 70% no tempo gasto para executar os métodos GGKB-FP e PROJ-FP.

Como parâmetros iniciais utilizamos dimensão inicial $p = 7$, $\epsilon_1 = \epsilon_2 = 10^{-4}$ para os métodos GGKB-FP e PROJ-FP e $\epsilon = 10^{-2}$ para G-LSQR. Também, como tínhamos disponível a decomposição GSVD de (A, L) , foi possível calcular soluções pelos métodos da Curva L e GCV. Mas ressaltamos que aos tempos de execução apresentados para estes métodos, deve ser acrescentado o tempo destinado ao cálculo da GSVD.

Na Figura 5.16 temos as imagens obtidas pelos seis métodos com 1% de ruído no lado direito \mathbf{b} . A imagem plotada é a solução

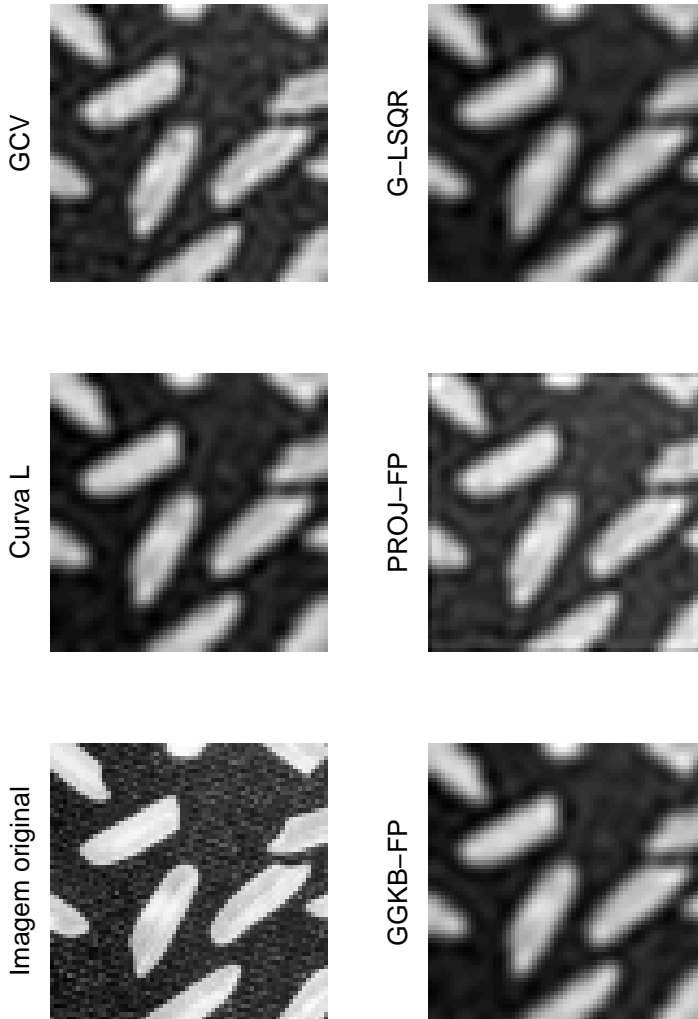


Figura 5.16: Imagens estimadas com 1% de ruído nos dados e a matriz L_{kron} .

que apresentou o menor erro relativo de 50 execuções. E na Figura 5.17 exibimos a função $\bar{\Psi}$ para o problema, obtida de uma das 50 execuções.

As Tabelas 5.32, 5.33 e 5.34 trazem os resultados obtidos após

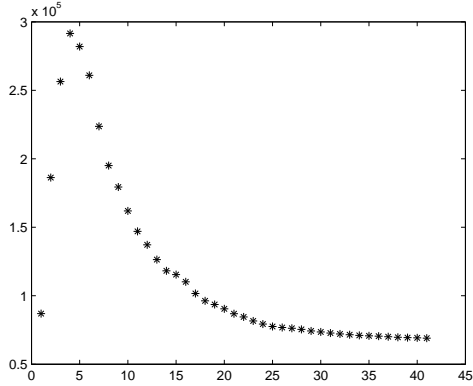


Figura 5.17: Função $\bar{\Psi}$ para o problema *Rice* com 1% de ruído.

50 execuções para os métodos GGKB-FP, PROJ-FP, G-LSQR, Curva L e GCV com diferentes níveis de ruído de 0,1%, 1% e 2,5%, respectivamente.

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	0,0730	0,0072	-	0,0064	0,0055
λ_{\min}	0,0727	0,0071	-	0,0063	0,0048
λ_{\max}	0,0731	0,0073	-	0,0066	0,0061
λ_{std}	0,0001	0,0001	-	0,0001	0,0002
\bar{E}	0,0733	0,0654	0,0848	0,0641	0,0641
E_{\min}	0,0717	0,0649	0,0812	0,0637	0,0637
E_{\max}	0,0740	0,0657	0,0876	0,0645	0,0647
E_{std}	0,0007	0,0002	0,0011	0,0002	0,0002
k_{\min}	80	80	36	-	-
k_{\max}	92	92	51	-	-
\bar{t}	7,9387	7,4499	9,8337	13,0226	1,7151
t_{\min}	7,5160	6,8440	8,2360	12,8910	1,5940
t_{\max}	9,1410	8,5940	12,0570	13,4070	1,9380
t_{std}	0,3895	0,4096	1,5805	0,0938	0,0829

Tabela 5.32: Resultados obtidos para o problema *Rice* após 50 execuções com 0,1% de ruído nos dados e usando a matriz L_{kron} .

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	0,1108	0,0816	-	0,0787	0,0382
λ_{\min}	0,1088	0,0807	-	0,0765	0,0331
λ_{\max}	0,1121	0,0830	-	0,0814	0,0441
λ_{std}	0,0007	0,0005	-	0,0011	0,0022
\bar{E}	0,0826	0,0822	0,0931	0,0801	0,0775
E_{\min}	0,0821	0,0807	0,0871	0,0796	0,0760
E_{\max}	0,0833	0,0831	0,0933	0,0810	0,0788
E_{std}	0,0003	0,0005	0,0022	0,0003	0,0007
k_{\min}	61	27	15	-	-
k_{\max}	63	32	33	-	-
\bar{t}	6,2532	2,1279	6,0552	14,4147	1,8517
t_{\min}	5,5780	1,7030	3,3700	12,8750	1,6250
t_{\max}	10,8130	4,2970	7,5280	25,7440	2,4310
t_{std}	0,8716	0,4625	0,6918	2,8422	0,1711

Tabela 5.33: Resultados obtidos para o problema *Rice* após 50 execuções com 1% de ruído nos dados e usando a matriz L_{kron} .

	GGKB-FP	PROJ-FP	G-LSQR	CURVA L	GCV
$\bar{\lambda}$	0,2236	0,2129	-	0,2035	0,0939
λ_{\min}	0,2205	0,2100	-	0,1972	0,0833
λ_{\max}	0,2266	0,2164	-	0,2100	0,1065
λ_{std}	0,0016	0,0015	-	0,0030	0,0060
\bar{E}	0,0947	0,0960	0,1066	0,0909	0,0872
E_{\min}	0,0940	0,0943	0,0995	0,0902	0,0853
E_{\max}	0,0954	0,0981	0,1072	0,0919	0,0891
E_{std}	0,0003	0,0008	0,0025	0,0004	0,0008
k_{\min}	40	15	15	-	-
k_{\max}	41	18	16	-	-
\bar{t}	4,4100	1,3340	3,4466	13,9853	1,8613
t_{\min}	3,6090	0,8910	3,3610	12,8750	1,6250
t_{\max}	7,5460	2,4690	3,9620	26,0630	2,5320
t_{std}	7,5460	2,4690	0,1154	26,0630	2,5320

Tabela 5.34: Resultados obtidos para o problema *Rice* após 50 execuções com 2,5% de ruído nos dados e usando a matriz L_{kron} .

Considerações Finais

Nosso objetivo, trabalhando com regularização na forma geral, foi evidenciar a importância da escolha da matriz L e mostrar métodos que possam atender ao caso em que ela é diferente da identidade. Finalizamos este trabalho ressaltando novamente que a escolha da matriz L não é uma tarefa fácil. Escolher uma matriz que consiga incorporar ao problema o máximo de informações sobre a solução exata é um problema tão complexo quanto obter uma solução que atenda as exigências do problema original.

Entendemos que problemas de grande porte podem chegar facilmente a milhões de variáveis, mas lembramos que as dimensões usadas aqui vão ao encontro das limitações das máquinas utilizadas (computadores). Os exemplos apresentados no capítulo ilustram a possibilidade de utilizar as idéias desenvolvidas em cada um dos três métodos do capítulo 4 para resolver problemas com um número muito maior de variáveis.

O estudo detalhado de projeções oblíquas e pseudo-inversa oblíqua foi importante para a compreensão da transformação da forma geral para a forma padrão. Por meio dele, pudemos entender melhor as ferramentas utilizadas e, conseqüentemente, desenvolver a teoria subjacente de forma consistente.

Os novos métodos obtiveram resultados próximos aos dos métodos já consolidados como Curva L e GCV, destacando que as três propostas não necessitam do cálculo da GSVD das matrizes (A, L) . Essa não dependência permite abordar problemas com um grande número de variáveis, também pelo fato de que a dimensão dos subespaços projetados é consideravelmente menor

que a dimensão original do problema.

Como limitação dos métodos GGKB-FP e G-LSQR destacamos o fato de trabalharem apenas com o caso em que L tem menos linhas do que colunas. Estratégias para contornar esta característica foram também estudadas durante o período de realização deste trabalho, no entanto, o tempo de computação para estes casos ultrapassou o que entendemos por aceitável.

Planos futuros estabelecem-se com intuito de:

1. Implementar GGKB-FP e G-LSQR eficientemente para o caso em que L tem mais linhas do que colunas.
2. Conhecer melhor a influência da matriz L na solução do problema e assim, poder escolher uma matriz que produza melhores soluções.

Apêndice A

SVD e GSVD: história e propriedades

A.1 A Decomposição em Valores Singulares

A decomposição em valores singulares — também denominada *SVD*, do inglês, *Singular Value Decomposition* — consiste na fatoração de uma matriz A ($m \times n$) como o produto $U\Sigma V^T$, onde U e V são matrizes ortogonais de ordem $m \times m$ e $n \times n$, respectivamente; e Σ é uma matriz da forma $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_p) \in \mathbb{R}^{m \times n}$ em que $p = \min\{m, n\}$. Tal fatoração foi introduzida por Eugenio Beltrami em 1873 e independentemente por Camille Jordan, no ano seguinte. Outros matemáticos como J. J. Sylvester, E. Schmidt, L. Autonne e H. Weyl também contribuíram para o desenvolvimento da teoria.

A SVD tornou-se largamente conhecida na Matemática na década de 60, quando G. H. Golub e outros matemáticos mostraram que poderia ser calculada eficazmente e usada como base para muitos algoritmos estáveis. Informações sobre o desenvolvimento histórico da decomposição em valores singulares podem ser encontradas em [52].

Quanto às suas aplicações, a SVD é largamente conhecida nas áreas de compressão de imagens e ainda possui diversas aplicações teóricas.

Formalmente, podemos enunciar a SVD de uma matriz real como no Capítulo 1:

Teorema A.1. *Para cada $A \in \mathbb{R}^{m \times n}$, existem matrizes ortogonais $U = [\mathbf{u}_1, \dots, \mathbf{u}_m] \in \mathbb{R}^{m \times m}$ e $V = [\mathbf{v}_1, \dots, \mathbf{v}_n] \in \mathbb{R}^{n \times n}$ tais que*

$$U^T A V = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_p) \in \mathbb{R}^{m \times n}, \quad p = \min\{m, n\}$$

com valores singulares σ_i , para cada $i \in \{1, 2, \dots, p\}$, ordenados de modo não-crescente $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$.

A demonstração deste teorema pode ser encontrada em [18].

Vamos denotar por $\sigma(A)$ o conjunto dos valores singulares não-nulos de A , isto é,

$$\sigma(A) = \{\sigma_i; i \in \{1, 2, \dots, r\}\}$$

em que $r = \text{posto}(A)$. Reciprocamente, se A tem r valores singulares não-nulos então $\text{posto}(A) = r$. Também $\sigma(A) = \lambda(A^T A)^2 = \lambda(AA^T)^2$, desta forma

$$\sigma \in \sigma(A) \Rightarrow \det(A^T A - \sigma^2 I_n) = 0. \quad (\text{A.1})$$

As colunas de U são denominadas vetores singulares à esquerda e são os autovetores ortonormais de AA^T , já as colunas de V são denominadas vetores singulares à direita e são os autovetores ortonormais de $A^T A$. Por causa da seguinte relação entre as colunas de U e V

$$A \mathbf{v}_i = \sigma_i \mathbf{u}_i \quad \text{e} \quad A^T \mathbf{u}_i = \sigma_i \mathbf{v}_i$$

para $i \in \{1, 2, \dots, p\}$, podemos escrever a SVD de uma matriz A como

$$A = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^T + \dots + \sigma_r \mathbf{u}_r \mathbf{v}_r^T,$$

com $\sigma_1 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_p = 0$.

Também podemos observar que

$$\begin{aligned} \{\mathbf{v}_1, \dots, \mathbf{v}_r\} &\text{ é uma base ortonormal para } \mathcal{R}(A^T) \\ \{\mathbf{v}_{r+1}, \dots, \mathbf{v}_n\} &\text{ é uma base ortonormal para } \mathcal{N}(A) \\ \{\mathbf{u}_1, \dots, \mathbf{u}_r\} &\text{ é uma base ortonormal para } \mathcal{R}(A) \\ \{\mathbf{u}_{r+1}, \dots, \mathbf{u}_m\} &\text{ é uma base ortonormal para } \mathcal{N}(A^H) \end{aligned} \quad (\text{A.2})$$

Geometricamente, os valores singulares de uma matriz A são os comprimentos dos semi-eixos do hiper-elipsóide E definido por $E = \{A\mathbf{x}; \|\mathbf{x}\| = 1\}$. Ou seja, para cada $i \in \{1, 2, \dots, p\}$ temos

$$\sigma_i = \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|} \quad (\text{A.3})$$

onde \mathbf{x} é autovetor de $A^T A$ associado ao autovalor σ_i^2 . Note que a fração em (A.3) pode ser nula quando $\mathcal{N}(A) = \mathcal{N}(A^T A) \neq \{0\}$.

Corolário A.1. *Seja $A \in \mathbb{R}^{m \times n}$, $m \geq n$, $\text{posto}(A) = r \leq n$ e $A = U\Sigma V^T$ a decomposição em valores singulares de A . Então podemos reescrever U e V da forma*

$$U = [U_1 \ U_2], \quad V = [V_1 \ V_2]$$

com $U_1 \in \mathbb{R}^{m \times r}$, $V_1 \in \mathbb{R}^{n \times r}$ e $\tilde{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$ tais que $A = U_1 \tilde{\Sigma} V_1^T$.

Este corolário define a chamada *Decomposição em Valores Singulares Reduzida*, onde são considerados apenas os elementos positivos da diagonal de Σ e o conjunto ortonormal $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_r\}$.

Também podemos utilizar a SVD reduzida da matriz A para definir projeções ortogonais sobre $\mathcal{R}(A)$ e $\mathcal{N}(A)$. Desta forma,

$$\begin{aligned} V_1 V_1^T &\text{ é projeção ortogonal sobre } \mathcal{N}(A)^\perp = \mathcal{R}(A^T) \\ V_2 V_2^T &\text{ é projeção ortogonal sobre } \mathcal{N}(A) \\ U_1 U_1^T &\text{ é projeção ortogonal sobre } \mathcal{R}(A) \\ U_2 U_2^T &\text{ é projeção ortogonal sobre } \mathcal{R}(A)^\perp = \mathcal{N}(A^T) \end{aligned} \quad (\text{A.4})$$

Também

$$\sigma \in \sigma(A) \Rightarrow \sigma^2 \text{ é um valor estacionário da função } \eta(\mathbf{x}) = \frac{\|A\mathbf{x}\|_2^2}{\|\mathbf{x}\|_2^2}. \quad (\text{A.5})$$

De fato, se σ é um valor singular de A então existe $\mathbf{x} \neq 0$ tal que $A^T A \mathbf{x} = \sigma^2 \mathbf{x}$. Por outro lado, note que $\eta(\mathbf{x})$ é equivalente a função

$$\bar{\eta}(\mathbf{u}) = \mathbf{u}^T A^T A \mathbf{u} \quad \text{para} \quad \|\mathbf{u}\|_2 = 1. \quad (\text{A.6})$$

Assim, \mathbf{u} é um ponto crítico de $\bar{\eta}$ se existe $\alpha \in \mathbb{R}$ tal que $A^T A \mathbf{u} = \alpha \mathbf{u}$. Tomando $\alpha = \sigma^2$ temos o resultado desejado.

A SVD de uma matriz A pode fornecer fórmulas simples para certas expressões que envolvem normas de matrizes. Como exemplos podemos citar:

- $\|A\| = \max_{\mathbf{x} \neq 0} \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|} = \sigma_1$;
- $0 < \min_{\mathbf{x} \neq 0} \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|} = \sigma_p$ se $p = \text{posto}(A)$;
- $\|A\|_F^2 = \text{traço}(A^T A) = \sigma_1^2 + \sigma_2^2 + \dots + \sigma_p^2$.

Definição A.1. A matriz pseudo-inversa de Moore-Penrose de $A \in \mathbb{R}^{m \times n}$ é a matriz $A^\dagger \in \mathbb{R}^{n \times m}$ que satisfaz as seguintes condições:

$$\begin{array}{ll} (i) & AA^\dagger A = A \quad (iii) \quad (AA^\dagger)^T = AA^\dagger \\ (ii) & A^\dagger AA^\dagger = A^\dagger \quad (iv) \quad (A^\dagger A)^T = A^\dagger A \end{array}$$

Observação A.1. Seja $A \in \mathbb{R}^{m \times n}$ com $\text{posto}(A) = r$ e $A = U_1 \tilde{\Sigma} V_1^T$ a SVD reduzida de A , onde $U_1 \in \mathbb{R}^{m \times r}$ e $V_1 \in \mathbb{R}^{r \times n}$ tem colunas ortonormais e

$$\tilde{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_r) \in \mathbb{R}^{r \times r}.$$

A matriz pseudo-inversa de A é dada por $A^\dagger = V_1 \tilde{\Sigma}^{-1} U_1^T$

Observação A.2. Para $A \in \mathbb{R}^{m \times n}$, usando a SVD de A , verifica-se que

$$A^\dagger = (A^T A)^\dagger A^T.$$

Se $\text{posto}(A) = n \leq m$ então $A^T A$ é inversível, $A^\dagger = (A^T A)^{-1} A^T$ e $A^\dagger A = I_n$. Por outro lado, se $\text{posto}(A) = m < n$ então AA^T é inversível, $A^\dagger = A^T (AA^T)^{-1}$ e $AA^\dagger = I_m$.

Outros resultados bastante conhecidos, e que podem ser verificados rapidamente através da SVD, são expressões matriciais para projetores ortogonais sobre $\mathcal{R}(A)$ e $\mathcal{N}(A)$, entre outros, como mostrado na observação a seguir

Observação A.3. Da SVD reduzida de A temos que $\mathcal{R}(A) = \mathcal{R}(U_1)$ e $\mathcal{R}(A^T) = \mathcal{R}(V_1)$. Também

a) AA^\dagger é a projeção sobre $\mathcal{R}(A)$

b) $I_m - AA^\dagger$ é a projeção sobre $\mathcal{R}(A)^\perp = \mathcal{N}(A^T)$

c) $A^\dagger A$ é a projeção sobre $\mathcal{R}(A^T)$

d) $I_n - A^\dagger A$ é a projeção sobre $\mathcal{R}(A^T)^\perp = \mathcal{N}(A)$

Proposição A.1. Sejam $X \in \mathbb{R}^{m \times p}$ e $Y \in \mathbb{R}^{p \times n}$ com $\text{posto}(X) = p = \text{posto}(Y)$. Então $(XY)^\dagger = Y^\dagger X^\dagger$ e $\text{posto}(XY) = p$.

Demonstração. Vamos verificar se a matriz $Y^\dagger X^\dagger$ satisfaz as condições de Moore-Penrose, dadas na Definição A.1. De fato,

$$\begin{aligned} XY(Y^\dagger X^\dagger)XY &= X(YY^\dagger)(X^\dagger X)Y = XI_p I_p Y = XY \\ Y^\dagger X^\dagger(XY)Y^\dagger X^\dagger &= Y^\dagger(X^\dagger X)(YY^\dagger)X^\dagger = Y^\dagger I_p I_p X^\dagger = Y^\dagger X^\dagger \\ (Y^\dagger X^\dagger XY)^T &= (Y^\dagger I_p Y)^T = (Y^\dagger Y)^T = Y^\dagger Y = Y^\dagger X^\dagger XY \\ (XY Y^\dagger X^\dagger)^T &= (X I_p X^\dagger)^T = (X X^\dagger)^T = X X^\dagger = X Y Y^\dagger X^\dagger \end{aligned}$$

Agora vamos verificar que $\text{posto}(XY) = p$. É imediato que $\mathcal{R}(XY) \subseteq \mathcal{R}(X)$. Também, $p = \dim \mathcal{R}(Y)$ e $\mathcal{R}(Y) \subseteq \mathbb{R}^p$ então $\mathcal{R}(Y) = \mathbb{R}^p$. Assim, dado $\mathbf{y} \in \mathbb{R}^p$, existe $\mathbf{z} \in \mathbb{R}^n$ tal que $Y\mathbf{z} = \mathbf{y}$.

Para $\mathbf{x} \in \mathcal{R}(X)$ segue que $X\mathbf{y} = \mathbf{x}$ para algum $\mathbf{y} \in \mathbb{R}^p$, que por sua vez é $Y\mathbf{z} = \mathbf{y}$ com $\mathbf{z} \in \mathbb{R}^n$. Portanto $\mathbf{x} = XY\mathbf{z}$ e $\mathbf{x} \in \mathcal{R}(XY)$.

Então $\mathcal{R}(XY) = \mathcal{R}(X)$, o que implica que $\text{posto}(XY) = p$.

■

Proposição A.2. Seja $A \in \mathbb{R}^{m \times n}$ com $m \geq n = \text{posto}(A)$ e $A = [A_1 \ A_2]$. Então

$$A^\dagger = \begin{bmatrix} B_1^\dagger \\ B_2^\dagger \end{bmatrix} \quad (\text{A.7})$$

onde $B_1 = (I - A_2 A_2^\dagger) A_1$ e $B_2 = (I - A_1 A_1^\dagger) A_2$

Demonstração. Como A tem posto coluna completo, segue que $A^T A$ é não singular. Então, da Observação A.2 temos que

$$(A^T A) A^\dagger = A^T.$$

Considerando a divisão de A dada anteriormente, temos

$$\begin{bmatrix} A_1^T A_1 & A_1^T A_2 \\ A_2^T A_1 & A_2^T A_2 \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} A_1^T \\ A_2^T \end{bmatrix} \quad (\text{A.8})$$

onde $A^\dagger = \begin{bmatrix} X \\ Y \end{bmatrix}$ com X e Y a serem determinadas.

Como as matrizes A_1 e A_2 tem posto coluna completo, então $A_1^T A_1$ e $A_2^T A_2$ são matrizes não singulares. Assim, multiplicando à esquerda da equação (A.8) a matriz

$$\begin{bmatrix} (A_1^T A_1)^{-1} & 0 \\ 0 & (A_2^T A_2)^{-1} \end{bmatrix} \quad (\text{A.9})$$

obtemos o seguinte sistema equivalente

$$\begin{cases} X + A_1^\dagger A_2 Y = A_1^\dagger \\ A_2^\dagger A_1 X + Y = A_2^\dagger \end{cases} \quad (\text{A.10})$$

lembrando que $A_i^\dagger = (A_i^T A_i)^{-1} A_i^T$ para $i = 1, 2$.

Isolando Y da segunda equação de (A.10) e substituindo na primeira equação resulta

$$A_1^\dagger (I - A_2 A_2^\dagger) A_1 X = A_1^\dagger (I - A_2 A_2^\dagger).$$

Multiplicando a última igualdade por $A_1^T A_1$ à esquerda segue a equação

$$A_1^T (I - A_2 A_2^\dagger) A_1 X = A_1^T (I - A_2 A_2^\dagger). \quad (\text{A.11})$$

Note que $I - A_2 A_2^\dagger = P_{\mathcal{R}(A_2)^\perp}$, onde $P_{\mathcal{X}}$ denota a projeção ortogonal sobre o subespaço \mathcal{X} . Assim, $I - A_2 A_2^\dagger$ é uma matriz simétrica e idempotente, e portanto

$$(I - A_2 A_2^\dagger) = (I - A_2 A_2^\dagger)^T (I - A_2 A_2^\dagger) \quad (\text{A.12})$$

Assim, a equação (A.11) pode ser escrita como

$$\left[\left(I - A_2 A_2^\dagger \right) A_1 \right]^T \left[\left(I - A_2 A_2^\dagger \right) A_1 \right] X = \left[\left(I - A_2 A_2^\dagger \right) A_1 \right]^T.$$

Se denotarmos $B_1 \equiv \left(I - A_2 A_2^\dagger \right) A_1$, com $X = B_1^\dagger$ na equação acima obtemos

$$B_1^T B_1 B_1^\dagger = B_1^T P_{\mathcal{R}(B_1)} = \left(P_{\mathcal{R}(B_1)} B_1 \right)^T = B_1^T. \quad (\text{A.13})$$

Portanto, $X = B_1^\dagger$ é uma solução da equação (A.11).

Analogamente, obtemos $Y = B_2^\dagger$. ■

Definição A.2. *Seja $X \in \mathbb{R}^{p \times n}$ com $\text{posto}(X) = r$. Dada a fatoração QR de X , com $Q \in \mathbb{R}^{p \times p}$ matriz ortogonal e $R \in \mathbb{R}^{p \times n}$ triangular superior e com o mesmo posto de X , considere*

$$Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \quad e \quad R = \begin{bmatrix} R_1 \\ O \end{bmatrix} \quad (\text{A.14})$$

com $Q_1 \in \mathbb{R}^{p \times r}$ com colunas ortonormais e $R_1 \in \mathbb{R}^{r \times n}$. Definimos a fatoração QR reduzida de X por $X = Q_1 R_1$.

A.2 Generalizações para a SVD

Generalizações para a decomposição SVD foram inicialmente propostas por Charles F. Van Loan em sua tese de Doutorado, defendida em 1973 na Universidade de Michigan, EUA, e seus estudos foram publicados em março de 1976. No artigo [54], ele apresenta duas generalizações para a decomposição em valores singulares do par de matrizes A e B , intituladas por BSVD e S,T-SVD e que serão apresentadas mais adiante. A ideia fundamental destas generalizações é trabalhar propriedades já conhecidas dos valores singulares de uma matriz A dada e, a partir disso, apresentar uma generalização do conceito de valores singulares para o par de matrizes (A, B) , em que A e B tem o mesmo número de colunas e A possui o número de linhas maior ou igual ao números de

colunas. Finalizando o artigo, são apresentados problemas onde os valores singulares generalizados são aplicáveis.

Em 1981, C. C. Paige e M. A. Saunders apresentaram uma reformulação da versão de Van Loan, trabalhando com matrizes A e B de quaisquer ordem, desde que com o mesmo número de colunas.

Recentemente generalizações ainda mais amplas da decomposição em valores singulares têm sido propostas. Como exemplo, podemos citar o trabalho de B. De Moor e H. Zha [11] que tem como objetivo fornecer uma decomposição em valores generalizados para um número qualquer de matrizes. Basicamente, a generalização para k matrizes formam uma árvore de fatorações onde, no nível k , existem 2^k fatorações, nem todas independentes. Uma importante propriedade desta generalização é que se pode obter uma decomposição em valores singulares para k matrizes a partir da decomposição para $k - 1$ matrizes.

A versão generalizada da SVD como utilizada atualmente é apresentada na equação (1.22) e provém da generalização apresentada em 1981.

A seguir, detalharemos as generalizações de Van Loan e de Paige e Saunders. No entanto, em [11] são apresentadas ideias interessantes para tal generalização além de mais referências para o assunto.

A.2.1 Decomposição CS

A decomposição CS descende do trabalho de C. Jordan em [33] sobre ângulos canônicos entre subespaços; e está implícita em artigos de C. Davis e W. Kahan [10] e de A. Björck e G. H. Golub [6]. A forma explícita e o nome são devidos a G. W. Stewart [49], sendo a forma generalizada para uma matriz dividida em submatrizes não quadradas proposta por Paige e Saunders em [40]. Este caso mais geral é apresentado no Teorema A.3 e contém ideias utilizadas na prova do teorema da decomposição em valores singulares generalizada apresentada por Paige e Saunders no mesmo artigo. Em [42] podemos encontrar uma pesquisa histórica sobre a decomposição CS e algoritmos computacionais

podem ser obtidos em [50].

Teorema A.2 (Decomposição CS - Versão I). *Considere uma matriz Q com colunas ortonormais decomposta como*

$$Q = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \quad (\text{A.15})$$

em que $Q_1 \in \mathbb{R}^{m_1 \times n}$ e $Q_2 \in \mathbb{R}^{m_2 \times n}$ para $m_1, m_2 \geq n$, então existem matrizes ortogonais $U_1 \in \mathbb{R}^{m_1 \times m_1}$, $U_2 \in \mathbb{R}^{m_2 \times m_2}$ e $V_1 \in \mathbb{R}^{n \times n}$ tais que

$$\begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix}^T Q V_1 = \begin{bmatrix} C \\ S \end{bmatrix} \quad (\text{A.16})$$

sendo

$$C = \text{diag}(\cos \theta_1, \dots, \cos \theta_n) \in \mathbb{R}^{m_1 \times n} \quad (\text{A.17})$$

$$S = \text{diag}(\text{sen} \theta_1, \dots, \text{sen} \theta_n) \in \mathbb{R}^{m_2 \times n} \quad (\text{A.18})$$

para $\frac{\pi}{2} \geq \theta_n \geq \dots \geq \theta_2 \geq \theta_1 \geq 0$.

Demonstração. Consulte [18] para maiores detalhes.

Note que se escrevermos

$$C = \begin{bmatrix} \tilde{C} & \\ & 0 \end{bmatrix} \quad \text{e} \quad S = \begin{bmatrix} \tilde{S} & \\ & 0 \end{bmatrix} \quad (\text{A.19})$$

com $\tilde{C} \in \mathbb{R}^{n \times n}$ e $\tilde{S} \in \mathbb{R}^{n \times n}$ temos que $\hat{C}^2 + \hat{S}^2 = I$. Denotando $c_i = \cos \theta_i$ e $s_i = \text{sen} \theta_i$ temos

$$1 \geq c_1 \geq c_2 \geq \dots \geq c_n \geq 0 \quad \text{e} \quad 1 \geq s_n \geq s_{n-1} \geq \dots \geq s_1 \geq 0 \quad (\text{A.20})$$

Agora consideremos a decomposição CS de uma matriz ortogonal $Q \in \mathbb{R}^{n \times n}$ decomposta da forma

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \quad (\text{A.21})$$

em que as submatrizes não são necessariamente quadradas. A partir deste modelo mais geral temos a seguinte decomposição CS.

Teorema A.3 (Decomposição CS - Versão II). *Seja $Q \in \mathbb{R}^{n \times n}$ matriz ortogonal como na equação (A.21). Então existem matrizes ortogonais*

$$U = \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} \quad e \quad V = \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix} \quad (\text{A.22})$$

tais que

$$U^T Q V = \left[\begin{array}{ccc|ccc} I & 0 & 0 & 0 & 0 & 0 \\ 0 & C & 0 & 0 & S & 0 \\ 0 & 0 & 0 & 0 & 0 & I \\ \hline 0 & 0 & 0 & I & 0 & 0 \\ 0 & S & 0 & 0 & -C & 0 \\ 0 & 0 & I & 0 & 0 & 0 \end{array} \right] \quad (\text{A.23})$$

em que

$$C = \text{diag}(c_1, \dots, c_p) \in \mathbb{R}^p, \quad 1 > c_1 \geq \dots \geq c_p > 0 \quad (\text{A.24})$$

$$S = \text{diag}(s_1, \dots, s_p) \in \mathbb{R}^p, \quad 1 > s_n \geq \dots \geq s_1 > 0 \quad (\text{A.25})$$

com $p \leq \frac{n}{2}$.

Demonstração. A demonstração pode ser encontrada em [40].

A seguir apresentaremos cada uma das GSVD mais detalhadamente, enfatizando aspectos destacados pelos autores.

A.2.2 A Primeira Versão de Van Loan

Vamos agora discutir as generalizações apresentadas por Van Loan. Para tal objetivo, serão consideradas as propriedades descritas em (A.1) e (A.5). Os valores singulares generalizados para Van Loan seriam os valores reais não negativos γ que satisfizerem algumas das propriedades abaixo:

$$\det(A^T A - \gamma^2 B^T B) = 0 \quad (\text{A.26})$$

$$\gamma \text{ é um valor estacionário de } \eta(\mathbf{x}) = \frac{\|A\mathbf{x}\|_S}{\|\mathbf{x}\|_T}, \quad (\text{A.27})$$

onde $\|\mathbf{x}\|_P$ já foi definido anteriormente. Para obter a generalização sugerida por (A.26) considere a definição de “ B -valores singulares”.

Definição A.3 (*B*-valores singulares). *Os B-valores singulares de uma matriz A são elementos do conjunto $\sigma(A, B)$ definidos por $\gamma(A, B) = \{\gamma; \gamma \geq 0, \det(A^T A - \gamma^2 B^T B) = 0\}$, onde $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{p \times n}$ e $m \geq n$.*

A seguinte decomposição é uma generalização da SVD baseada na definição de *B*-valores singulares.

Teorema A.4 (BSVD). *Suponha $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{p \times n}$ e $m \geq n$. Então existem matrizes ortogonais $U \in \mathbb{R}^{m \times m}$ e $V \in \mathbb{R}^{p \times p}$ e uma matriz não singular $X \in \mathbb{R}^{n \times n}$ tal que*

$$\begin{aligned} U^T A X &= \Sigma_A = \text{diag}(\alpha_1, \dots, \alpha_n) \text{ com } \alpha_i \geq 0 \\ V^T B X &= \Sigma_B = \text{diag}(\beta_1, \dots, \beta_q) \text{ com } \beta_i \geq 0 \end{aligned} \quad (\text{A.28})$$

onde $q = \min\{p, n\}$, $r = \text{rank}(B)$ e $\beta_1 \geq \dots \geq \beta_r > \beta_{r+1} = \dots = \beta_q = 0$. Se $\alpha_i = 0$ para algum $r + 1 \leq i \leq n$, então $\gamma(A, B) = \{\gamma; \gamma \geq 0\}$. Caso contrário,

$$\gamma(A, B) = \left\{ \frac{\alpha_i}{\beta_i}; i = 1, \dots, r \right\}.$$

Demonstração. Seja $\begin{bmatrix} A \\ B \end{bmatrix}$ de posto k e sua SVD dada por

$$Q^T \begin{bmatrix} A \\ B \end{bmatrix} Z_1 = \text{diag}(\gamma_1, \dots, \gamma_n) \quad (\text{A.29})$$

onde $\gamma_1 \geq \dots \geq \gamma_k > \gamma_{k+1} = \dots = \gamma_n = 0$. Defina $D = \text{diag}(\gamma_1, \dots, \gamma_k) \in \mathbb{R}^{k \times k}$ e escreva $Z_1 = [Z_{11} \ Z_{12}] \in \mathbb{R}^{n \times n}$ com $Z_{11} \in \mathbb{R}^{n \times k}$. Assim $A_1 = A Z_{11} D^{-1} \in \mathbb{R}^{m \times k}$ e $B_1 = B Z_{11} D^{-1} \in \mathbb{R}^{p \times k}$ satisfazem $A_1^T A_1 + B_1^T B_1 = I_k$. De fato,

$$A_1^T A_1 + B_1^T B_1 = D^{-1} Z_{11} (A^T A + B^T B) Z_{11} D^{-1} \quad (\text{A.30})$$

e

$$A^T A + B^T B = Z_{11} D^T D Z_{11}^T. \quad (\text{A.31})$$

O resultado segue do fato de $Z_{11}^T Z_{11} = I_k$. Se $r = \text{posto}(B)$, $s = \min(p, k)$ e

$$V^T B_1 Z_2 = \text{diag}(\beta_1, \dots, \beta_s) \quad (\text{A.32})$$

sendo a SVD de B_1 com $\beta_1 \geq \dots \geq \beta_r > \beta_{r+1} = \dots = \beta_s = 0$, então

$$V^T B Z_1 \begin{bmatrix} D^{-1} Z_2 & 0 \\ 0 & I_{n-k} \end{bmatrix} = \Sigma_B = \text{diag}(\beta_1, \dots, \beta_q) \quad (\text{A.33})$$

onde $q = \min\{p, n\}$ e $\beta_{s+1} = \dots = \beta_q = 0$. Agora, observe que as colunas de $A_1 Z_2 \in \mathbb{R}^{m \times k}$ são ortogonais pois

$$(A_1 Z_2)^T A_1 Z_2 = Z_2^T (I_k - B_1^T B_1) Z_2 = \text{diag}(1 - \beta_1^2, \dots, 1 - \beta_k^2) \quad (\text{A.34})$$

pois Z_2 é ortogonal e usando equação (A.32). Seja $A_1 Z_2 = UR$ a decomposição QR de $A_1 Z_2$, com $U \in \mathbb{R}^{m \times m}$ e

$$R = \begin{bmatrix} \widehat{R} \\ 0 \end{bmatrix}, \widehat{R} \in \mathbb{R}^{k \times k} \quad (\text{A.35})$$

para $\widehat{R} \in \mathbb{R}^{k \times k}$ triangular superior com elementos da diagonal todos positivos. Denotemos tais elementos por $\alpha_1, \dots, \alpha_k$. Da equação (A.34) segue que a matriz \widehat{R} é diagonal e assim temos que

$$U^T A_1 Z_2 = \text{diag}(\alpha_1, \dots, \alpha_k) \in \mathbb{R}^{m \times k}. \quad (\text{A.36})$$

Definindo $\alpha_i = 0$ para $i = k + 1, \dots, n$ vemos que

$$U^T A Z_1 \begin{bmatrix} D^{-1} Z_2 & 0 \\ 0 & I_{n-k} \end{bmatrix} = \Sigma_A = \text{diag}(\alpha_1, \dots, \alpha_n) \quad (\text{A.37})$$

Denotando $X = Z_1 \begin{bmatrix} D^{-1} Z_2 & 0 \\ 0 & I_{n-k} \end{bmatrix}$ e das equações (A.33) e (A.37) segue as decomposições desejadas. Note que X é inversível pois Z_1 e Z_2 são matrizes ortogonais.

Para obtermos os B -valores singulares note que

$$\begin{aligned} A^T A - \gamma^2 B^T B &= X^{-T} \Sigma_A^T \Sigma_A X^{-1} - \gamma^2 X^{-T} \Sigma_B^T \Sigma_B X^{-1} \\ &= X^{-T} (\Sigma_A^T \Sigma_A - \gamma^2 \Sigma_B^T \Sigma_B) X^{-1}. \end{aligned}$$

Assim,

$$\det(A^T A - \gamma^2 B^T B) = \det(X)^{-2} \prod_{i=1}^r (\alpha_i^2 - \gamma^2 \beta_i^2) \prod_{i=r+1}^n (\alpha_i^2).$$

Como X é inversível segue que $\det(X) \neq 0$, o que completa a prova. ■

Observação A.4. *As matrizes U , V e X não são únicas. Em particular, podemos substituir X em (A.28) por XD onde $D = \text{diag}(\delta_1, \dots, \delta_n)$ com $\delta_i > 0$. Se $A^T A$ e $B^T B$ comutam então é possível escolher X ortogonal. Se $B = I_n$ então $V = X\Sigma_B^{-1}$ e $U^T A (X\Sigma_B^{-1}) = \Sigma_A \Sigma_B$ é a SVD de A .*

Observação A.5. *Note que $\gamma(A, B) = \{\gamma; \gamma \geq 0\} \Leftrightarrow \mathcal{N}(A) \cap \mathcal{N}(B) \neq \{\mathbf{0}\}$. De fato, se $\mathbf{x} \in \mathcal{N}(A) \cap \mathcal{N}(B)$ então $A^T A \mathbf{x} = \mathbf{0}$ e $B^T B \mathbf{x} = \mathbf{0}$. Assim*

$$(A^T A - \gamma^2 B^T B) \mathbf{x} = \mathbf{0} \quad \forall \gamma \geq 0 \Rightarrow \gamma(A, B) = \{\gamma; \gamma \geq 0\}. \quad (\text{A.38})$$

Reciprocamente, se $\det(A^T A - \gamma^2 B^T B) = 0 \quad \forall \gamma \geq 0 \dots$

Neste caso, $\mathcal{N}(A) \cap \mathcal{N}(B) = \mathcal{N}\left(\begin{bmatrix} A \\ B \end{bmatrix}\right)$ é o espaço gerado pelas $n - k$ colunas de X , onde $k = \text{posto}\left(\begin{bmatrix} A \\ B \end{bmatrix}\right)$. Também $\mathcal{N}(A) \cap \mathcal{N}(B)$ é ortogonal ao espaço gerado pelas k primeiras colunas de X .

Observação A.6. *Se \mathbf{x}_i denota as colunas de X então do teorema A.4 segue que*

$$\|A \mathbf{x}_i\|_2 = \alpha_i, \quad i = 1, \dots, n \quad (\text{A.39})$$

$$\|B \mathbf{x}_i\|_2 = \beta_i, \quad i = 1, \dots, q. \quad (\text{A.40})$$

Se as colunas de X são escalonadas de forma que se tornem vetores unitários então

$$\sigma_{\min}(A) \leq \alpha_i \leq \sigma_{\max}(A) \quad (\text{A.41})$$

$$\sigma_{\min}(B) \leq \beta_i \leq \sigma_{\max}(B). \quad (\text{A.42})$$

Se $\text{posto}(A) = n$, a BSVD de A pode ser calculada de um modo mais simples do que os passos mostrados no teorema A.4. Os passos neste caso são mostrados abaixo.

1. Usando fatoração QR com pivoteamento obtemos $Q_1^T AP = \begin{bmatrix} R \\ 0 \end{bmatrix}$, com $R \in \mathbb{R}^{n \times n}$ inversível;
2. Calculando a SVD de BPR^{-1} temos $V^T (BPR^{-1}) Z = \Sigma_B$;
3. Faça $X = PR^{-1}Z$ e $U = Q_1 \begin{bmatrix} Z & 0 \\ 0 & I_{m-n} \end{bmatrix}$

A segunda generalização proposta por Van Loan segue da propriedade em (A.27) e necessita das seguintes definições.

Definição A.4 (Matriz P ortogonal). *Seja $P \in \mathbb{R}^{n \times n}$ matriz positiva definida. Uma matriz $Q \in \mathbb{R}^{n \times n}$ é P ortogonal se $Q^T P Q = I_n$.*

Definição A.5 (S,T-valores singulares). *Sejam $S \in \mathbb{R}^{m \times m}$ e $T \in \mathbb{R}^{n \times n}$ matrizes simétricas positivas definidas com $m \geq n$. Os S,T-valores singulares de $A \in \mathbb{R}^{m \times n}$ são os elementos do conjunto $\gamma_{S,T}(A)$ definidos por*

$$\gamma_{S,T}(A) = \left\{ \gamma; \gamma \geq 0, \gamma \text{ é um valor estacionário de } \frac{\|Ax\|_S}{\|x\|_T} \right\}.$$

Teorema A.5 (S,T-SVD). *Sejam $S \in \mathbb{R}^{m \times m}$ e $T \in \mathbb{R}^{n \times n}$ matrizes simétricas positivas definidas com $m \geq n$ e $A \in \mathbb{R}^{m \times n}$. Então existe uma matriz S ortogonal $U \in \mathbb{R}^{m \times m}$ e uma T ortogonal $V \in \mathbb{R}^{n \times n}$ tais que*

$$U^{-1}AV = \text{diag}(\gamma_1, \dots, \gamma_n) \quad (\text{A.43})$$

$$e \gamma_{S,T}(A) = \{\gamma_1, \dots, \gamma_n\}.$$

Demonstração. Sejam $S = LL^T$ e $T = KK^T$ as fatorações Choleski de S e T , respectivamente. Defina $C = L^T AK^{-T} \in \mathbb{R}^{m \times n}$ e seja $Q^T CZ = \Sigma_C = \text{diag}(\gamma_1, \dots, \gamma_n)$ a SVD de C . Assim,

$$\Sigma_C = Q^T CZ = Q^T L^T AK^{-T} V = (LQ)^T A(K^{-T} Z). \quad (\text{A.44})$$

Denotando $U = L^{-T}Q \in \mathbb{R}^{m \times m}$ e $V = K^{-T}Z \in \mathbb{R}^{n \times n}$ temos

$$U^T S U = (Q^T L^{-1}) L L^T (L^{-T} Q) = I_m \quad (\text{A.45})$$

$$V^T T V = (Z^T K^{-1}) K K^T (K^{-T} Z) = I_n \quad (\text{A.46})$$

Considere a função $\eta(\mathbf{x}) = \frac{\|A\mathbf{x}\|_S^2}{\|\mathbf{x}\|_T^2}$ e seu gradiente dado por

$\nabla \eta(\mathbf{x}) = \frac{2A^T S A \mathbf{x} - 2T \mathbf{x}}{\mathbf{x}^T T \mathbf{x}}$. Como T é definida positiva segue que os valores estacionários de η são as raízes de $\det(A^T S A - \gamma^2 T) = 0$. Também, escrevendo $A^T S A$ em termos de D , Z e K e usando o fato de Z ser ortogonal temos que $A^T S A - \gamma^2 T = K Z (D^T D - \gamma^2 I_n) Z^T K^T$. Assim

$$\begin{aligned} \det(A^T S A - \gamma^2 T) &= \det(K Z) \det(D^T D - \gamma^2 I_n) \det(Z^T K^T) \\ &= \det(K K^T) \det(D^T D - \gamma^2 I_n) \\ &= \det(T) \prod_{i=1}^n (\gamma_i^2 - \gamma^2). \end{aligned}$$

Como T é simétrica definida positiva segue que $\det(T) \neq 0$. Logo, os elementos de $\gamma_{S,T}$ são exatamente os elementos da diagonal de D . ■

Existem algumas conexões entre as duas decomposições propostas por Van Loan. Por exemplo, se $\text{posto}(B) = n \leq p$ no teorema A.4 então

$$\mathbf{a)} \quad \gamma(A, B) = \gamma_{I_m, B^T B}(A).$$

De fato, da definição A.5 temos

$$\gamma_{I_m, B^T B}(A) = \left\{ \gamma; \gamma \geq 0, \gamma \text{ é um valor estacionário de } \frac{\|A\mathbf{x}\|_{I_m}}{\|\mathbf{x}\|_{B^T B}} \right\}$$

e $\|A\mathbf{x}\|_{I_m} = \|A\mathbf{x}\|$ e $\|\mathbf{x}\|_{B^T B} = \|B\mathbf{x}\|$. Com isso, $\gamma \in \gamma_{I_m, B^T B}(A)$ se

$$\gamma = \frac{\|A\mathbf{x}\|}{\|B\mathbf{x}\|} \Rightarrow (A^T A - \gamma^2 B^T B) \mathbf{x} = \mathbf{0}$$

para $\mathbf{x} \neq \mathbf{0}$.

b) a matriz $X \text{diag}(\beta_1^{-1}, \dots, \beta_n^{-1})$ é $B^T B$ ortogonal.

De fato, do teorema A.4 temos que $B = V \Sigma_B X^{-1}$ e, denotando $\Sigma_B^{-1} \text{diag}(\beta_1^{-1}, \dots, \beta_n^{-1})$, segue que $B^T B = X^{-T} \Sigma_B^T \Sigma_B X^{-1}$. Utilizando a definição A.5 segue que

$$(X \Sigma_B^{-1})^T B^T B (X \Sigma_B^{-1}) = \Sigma_B^{-T} X^T X^{-T} \Sigma_B^T \Sigma_B X^{-1} X \Sigma_B^{-1} = I_n$$

A.2.3 Melhoramentos por Paige e Saunders

No artigo de 1981, os autores trabalhavam com matriz complexa, no entanto, apresentaremos aqui os resultados para matrizes reais, fazendo as devidas adaptações.

Novamente, estamos interessados na generalização da SVD para quaisquer duas matrizes $A \in \mathbb{R}^{m \times n}$ e $B \in \mathbb{R}^{p \times n}$ com o mesmo número de colunas. Segundo os autores, uma das motivações para buscar tal generalização vem do fato que se $p = n$ e B é não singular, existem problemas teóricos e numéricos que precisam da SVD de $C \equiv AB^{-1} = USV^T$. No entanto, para B é mal condicionada, formar AB^{-1} e encontrar sua SVD pode levar a erros numéricos grandes e desnecessários.

Sugere-se, então, estabelecer um resultado da forma $U^T A = SV^T B$, e se $S^B \in \mathbb{R}^{m \times m}$ é diagonal definida positiva e $S_A \equiv S_B S$ então

$$S_B U^T A = S_A V^T B \tag{A.47}$$

onde $S_A = \text{diag}(\alpha_1, \alpha_2, \dots)$ e $S_B = \text{diag}(\beta_1, \dots, \beta_m)$ em que $\alpha_i = \beta_i \sigma_i$ para $i = 1, \dots, \min\{m, n\}$.

Uma decomposição desta forma seria a decomposição em valores singulares generalizados (GSVD) para A e B não singular. Neste caso, Paige e Saunders chamaram (α_i, β_i) de um *par de valores singulares* para A e B . O uso de pares evita distinção entre A e B , como apresentada por Van Loan [54].

Se B é não singular e $m \geq n$, segue do Teorema A.4 que

$$U^T A X = \Sigma_A \quad \text{e} \quad V^T B X = \Sigma_B \tag{A.48}$$

com U e V matrizes ortogonais e X não singular. Então

$$U^T A = \Sigma_A \Sigma_B^{-1} V^T B \quad (\text{A.49})$$

pois Σ_B também é não singular.

Como S_B é positiva definida e, portanto, não singular, segue que (A.49) é equivalente a (A.47). De fato, se escrevemos

$$S_B = \begin{bmatrix} \tilde{S}_B & 0 \\ 0 & \hat{S}_B \end{bmatrix}, \quad S_A = S_B S = \begin{bmatrix} \tilde{S}_B \tilde{S} \\ 0 \end{bmatrix} \equiv \begin{bmatrix} \tilde{S}_A \\ 0 \end{bmatrix} \quad (\text{A.50})$$

com $\tilde{S}_B \in \mathbb{R}^{n \times n}$ então

$$\begin{aligned} S_B U^T A &= S_A V^T B \\ U^T A &= S_B^{-1} S_A V^T B \\ U^T A &= \begin{bmatrix} \tilde{S}_B^{-1} & 0 \\ 0 & \hat{S}_B^{-1} \end{bmatrix} \begin{bmatrix} \tilde{S}_A \\ 0 \end{bmatrix} V^T B \\ U^T A &= \begin{bmatrix} \tilde{S}_B^{-1} \tilde{S}_A \\ 0 \end{bmatrix} V^T B \\ U^T A &= \begin{bmatrix} \tilde{S}_A \tilde{S}_B^{-1} \\ 0 \end{bmatrix} V^T B \\ U^T A &= \begin{bmatrix} \tilde{S}_A \\ 0 \end{bmatrix} \tilde{S}_B^{-1} V^T B \\ U^T A &= S_A \tilde{S}_B^{-1} V^T B \end{aligned}$$

Então (A.49) fornece uma generalização de (A.47) para B não quadrada quando $m \geq n$.

A seguir apresentamos a generalização apresentada por Paige e Saunders, sem restrições para as dimensão das matrizes A e B , a menos da fato de que ambas devem ter o mesmo número de colunas.

Nesta abordagem, são utilizadas transformações ortogonais da forma: dada uma matriz C , podemos encontrar matrizes ortogonais P e Q tais que

$$P^T C Q = \begin{bmatrix} R & 0 \\ 0 & 0 \end{bmatrix}, \quad (\text{A.51})$$

onde R é não singular com $\text{posto}(R) = \text{posto}(C)$. A equação (A.51) pode ser a SVD de C ou, por exemplo, uma fatoração QR de C tal que $CE = QR$, onde E é uma matriz de permutação escolhida para que o módulo da diagonal de R seja decrescente. Se a fatoração QR é usada, a matriz R é triangular superior mas em ambos os casos, $\sigma(R) = \sigma(C)$. Detalhes em [18].

Teorema A.6 (GSVD 1981). *Para $A \in \mathbb{R}^{m \times n}$ e $B \in \mathbb{R}^{p \times n}$ matrizes dadas e $C = \begin{bmatrix} A \\ B \end{bmatrix}$ com $k = \text{posto}(C)$, existem matrizes ortogonais $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{p \times p}$, $W \in \mathbb{R}^{k \times k}$ e $Q \in \mathbb{R}^{n \times n}$ tais que*

$$U^T A Q = \Sigma_A [W^T R \quad O], \quad V^T B Q = \Sigma_B [W^T R \quad O] \quad (\text{A.52})$$

em que $O \in \mathbb{R}^{k \times (n-k)}$ é matriz nula, $R \in \mathbb{R}^{k \times k}$ é não singular com valores singulares iguais aos valores singulares não-nulos de C e

$$\Sigma_A \equiv \begin{bmatrix} I_A & & \\ & S_A & \\ & & O_A \end{bmatrix} \in \mathbb{R}^{m \times k}, \quad (\text{A.53})$$

$$\Sigma_B \equiv \begin{bmatrix} O_B & & \\ & S_B & \\ & & I_B \end{bmatrix} \in \mathbb{R}^{p \times k}, \quad (\text{A.54})$$

para

$$\left. \begin{array}{l} I_A \in \mathbb{R}^{r \times r} \\ I_B \in \mathbb{R}^{(k-r-s) \times (k-r-s)} \end{array} \right\} \text{ matrizes identidades,}$$

$$\left. \begin{array}{l} O_A \in \mathbb{R}^{(m-r-s) \times (k-r-s)} \\ O_B \in \mathbb{R}^{(p-k+r) \times r} \end{array} \right\} \text{ matrizes nulas e}$$

$$\left. \begin{array}{l} S_A \equiv \text{diag}(\alpha_{r+1}, \dots, \alpha_{r+s}) \in \mathbb{R}^{s \times s} \\ S_B \equiv \text{diag}(\beta_{r+1}, \dots, \beta_{r+s}) \in \mathbb{R}^{s \times s} \end{array} \right\}$$

matrizes reais satisfazendo

$$1 > \alpha_{r+1} \geq \dots \geq \alpha_{r+s} > 0 \quad (\text{A.55})$$

$$0 < \beta_{r+1} \leq \dots \leq \beta_{r+s} < 1 \quad (\text{A.56})$$

$$\alpha_i^2 + \beta_i^2 = 1 \quad \text{para } i = r+1, \dots, r+s. \quad (\text{A.57})$$

Demonstração. Seja $C = \begin{bmatrix} A \\ B \end{bmatrix} \in \mathbb{R}^{(m+p) \times n}$ e considere uma fatoração de C como na equação (A.51). Neste caso, $R \in \mathbb{R}^{k \times k}$ com $k = \text{posto}(R) = \text{posto}(C)$. Decomponha P e Q como

$$Q = [Q_1 \ Q_2] \quad \text{e} \quad P = [P_1 \ P_2] = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$$

em que Q_1 e P_1 tem k colunas, $P_{11} \in \mathbb{R}^{m \times k}$ e $P_{21} \in \mathbb{R}^{p \times k}$.

Assim

$$\|P_{11}\|_2 \leq \|P_1\|_2 = 1$$

pois P_1 tem colunas ortonormais e, portanto, $0 \leq \sigma(P_{11}) \leq 1$. Também, se $P_{11} = U\Sigma_A W^T$ é a decomposição SVD de P_{11} então Σ_A é como nas equações (A.53) e (A.55).

Agora considere uma decomposição¹ de $P_{21}W$ como

$$P_{21}W = VL$$

em que V é ortogonal e $L = \begin{bmatrix} 0 \\ \widehat{L} \end{bmatrix}$ com \widehat{L} triangular inferior com diagonal não-negativa.

Note que a matriz

$$\begin{bmatrix} U^T & 0 \\ 0 & V^T \end{bmatrix} P_1 W$$

tem colunas ortonormais pois é um produto de matrizes com colunas ortonormais e

$$\begin{bmatrix} U^T & 0 \\ 0 & V^T \end{bmatrix} P_1 W = \begin{bmatrix} U^T & 0 \\ 0 & V^T \end{bmatrix} \begin{bmatrix} P_{11} \\ P_{21} \end{bmatrix} W = \begin{bmatrix} \Sigma_A \\ L \end{bmatrix}.$$

Como $\Sigma_A = \begin{bmatrix} I_A & & \\ & S_A & \\ & & O_A \end{bmatrix}$ e as colunas de $\begin{bmatrix} \Sigma_A \\ L \end{bmatrix}$ tem norma 1, segue que $L = \Sigma_B$ da equação (A.54) e

$$\Sigma_A^T \Sigma_A + \Sigma_B^T \Sigma_B = I_k.$$

¹Mais detalhes sobre esse tipo de decomposição ver [18].

Logo, se α_i e β_i são elementos de Σ_A e Σ_B , respectivamente, tais que $0 < \alpha_i, \beta_i < 1$ então a equação (A.57) é satisfeita, completando a demonstração. ■

Os autores enfatizam que, diferentemente da BSVD apresentada por Van Loan, nesta generalização A e B são tratadas da mesma forma, a menos talvez pela ordenção dos elementos em Σ_A e Σ_B . No entanto, os blocos diagonais de Σ_A podem ser trocadas se aqueles de Σ_B forem trocados do mesmo modo e similarmente para os elementos da diagonal de S_A e S_B . Por isso, esta fatoração é apresentada como a GSVD das matrizes A e B .

Mesmo com essa distinção, podemos relacionar as decomposições em (A.52) ao trabalho de Van Loan [54] escrevendo

$$X \equiv Q \begin{bmatrix} R^{-1}W & 0 \\ 0 & I \end{bmatrix} \quad (\text{A.58})$$

o que implica

$$U^T AX = [\Sigma_A \ 0], \quad V^T BX = [\Sigma_B \ 0]. \quad (\text{A.59})$$

Podemos atribuir n pares de valores singulares (α_i, β_i) , $i = 1, \dots, n$ para A e B , correspondente para cada coluna de $U^T AX$ e $V^T BX$ em (A.59). Seguindo (A.53) e (A.54) determinamos os k primeiros desses pares como

$$\alpha_i = 1, \quad \beta_i = 0 \quad \text{para } i = 1, \dots, r, \quad (\text{A.60})$$

$$\alpha_i, \beta_i \quad \text{para } i = r + 1, \dots, r + s, \quad (\text{A.61})$$

$$\alpha_i = 0, \quad \beta_i = 1 \quad \text{para } i = r + s + 1, \dots, k, \quad (\text{A.62})$$

e chamamos estes pares de não-triviais. Note que para i entre $r + 1$ e $r + s$, $\alpha_i \in S_A$ e $\beta_i \in S_B$. Como os restantes correspondem a colunas nulas em (A.59) e não tem valor numérico particular, chamaremos estes $n - k$ de pares triviais de valores singulares de A e B .

Quando B é não singular, os valores singulares de AB^{-1} são da forma $\frac{\alpha_i}{\beta_i}$ e para manter a analogia, Paige e Saunders denominam os pares de valores singulares como valores singulares generalizados:

- infinitos para α_i e β_i como em (A.60),
- ordinários para α_i e β_i como em (A.61),
- nulos para α_i e β_i como em (A.62),

e ao restante como valores singulares generalizados arbitrários ou indefinidos.

Em [54] Van Loan mostrou que esses pares de valores singulares correspondem a uma generalização do conceito de valor singular de uma matriz C para duas matrizes $A \in \mathbb{R}^{m \times n}$ e $B \in \mathbb{R}^{p \times n}$:

Encontrar os pares de valores singulares (α, β) satisfazendo $\alpha^2 + \beta^2 \neq 0$ para $\alpha, \beta \geq 0$ tais que $\det(\beta^2 A^T A - \alpha^2 B^T B) = 0$

Note que

$$\beta^2 A^T A - \alpha^2 B^T B \tag{A.63}$$

\Updownarrow

$$\begin{bmatrix} R^T W \\ 0 \end{bmatrix} \begin{bmatrix} \beta^2 I_A & & \\ & \beta^2 S_A^2 - \alpha^2 S_B^2 & \\ & & -\alpha^2 I_B \end{bmatrix} \begin{bmatrix} W^T R & 0 \end{bmatrix} \tag{A.64}$$

Assim, $\det(\beta^2 A^T A - \alpha^2 B^T B) = 0$ se o determinante da matriz $n \times n$ dada em (A.64) for nulo. Se esta matriz tem posto no máximo k , podemos dizer há $n - k$ pares de valores singulares arbitrários (α, β) , sendo estes os pares triviais. Os demais pares são apenas os pares não triviais dados em (A.60), (A.61) e (A.62).

Apêndice B

Métodos para Escolha do Parâmetro de Regularização

B.1 Método da Validação Cruzada Generalizada (GVC) e Weighted-GCV

A Validação Cruzada Generalizada, do inglês Generalized Cross-Validation (GCV), desenvolvida por G. H. Golub, M. T. Heath e G. Wahba [17], é um método muito popular para a escolha do parâmetro de regularização baseada em considerações estatísticas. A GCV sugere que um bom valor para o parâmetro de regularização deve prever dados no vetor \mathbf{b} que estejam faltando ou que foram retirados. Mais precisamente, se um elemento (uma equação do sistema de equações $A\mathbf{x} = \mathbf{b}$) arbitrário \mathbf{b}_j do vetor \mathbf{b} for removido, então a solução regularizada correspondente deve prever bem essa falta. Baseado nesse princípio, o parâmetro de regularização é o valor λ que minimiza a função GCV

$$G_{A,\mathbf{b}}(\lambda) = \frac{n \left\| (I - AA_\lambda^\dagger) \mathbf{b} \right\|_2^2}{\left(\text{tr}(I - AA_\lambda^\dagger) \right)} \quad (\text{B.1})$$

em que $A_\lambda^\dagger = (A^T A + \lambda^2 I)^{-1} A^T$ representa a pseudo-inversa da matriz \bar{A} do sistema $\bar{A}\mathbf{x}_\lambda = \bar{\mathbf{b}}$ em que $\bar{A} = (A^T A + \lambda^2 I)$ e $\bar{\mathbf{b}} = A^T \mathbf{b}$, e a solução regularizada, \mathbf{x}_λ , pode ser escrita como $\mathbf{x}_\lambda = A_\lambda^\dagger \mathbf{b}$. Substituindo a SVD da matriz A na função GSV temos

$$G_{A,b}(\lambda) = \frac{n \left(\sum_{i=1}^n \left(\frac{\lambda^2}{\sigma_i^2 + \lambda^2} \mathbf{u}_i^T \mathbf{b} \right)^2 + \sum_{i=n+1}^m (\mathbf{u}_i^T \mathbf{b})^2 \right)}{\left((m-n) + \sum_{i=1}^n \frac{\lambda^2}{\sigma_i^2 + \lambda^2} \right)^2} \quad (\text{B.2})$$

o que torna a função GCV computacionalmente conveniente para ser avaliada e utilizada por algoritmos de minimização. Em outras palavras, se $(\mathbf{b}_j - [A\mathbf{x}]_j)^2$ for removido, o método GCV procura minimizar o erro quando \mathbf{x} é o minimizador de

$$\sum_{i=1, j \neq i}^m (\mathbf{b}_i - [A\mathbf{x}]_i)^2 + \lambda^2 \|\mathbf{x}\|_2^2. \quad (\text{B.3})$$

Seja $E_j = \text{diag}(1, \dots, 0, 1, \dots, 1) \in \mathbb{R}^{m \times m}$ em que o elemento 0 aparece na j -ésima posição. Então a minimização acima é equivalente a

$$\mathbf{x} = \text{argmin} \left\{ \|E_j(\mathbf{b} - A\mathbf{x})\|_2^2 + \lambda^2 \|\mathbf{x}\|_2^2 \right\} \quad (\text{B.4})$$

Em estudos comparativos encontrados na literatura podemos apontar que uma desvantagem do método GCV para a determinação do parâmetro de regularização de Tikhonov é que nem sempre este método funciona, no sentido de que dependendo de como o erro está distribuído nos dados, o parâmetro encontrado pode produzir uma solução não satisfatória.

Recentemente um novo método apareceu na literatura que é baseado na GCV chamado de Weighted-GCV ou simplesmente W-GCV [9]. Em vez de minimizar a função GCV, o método W-GCV busca minimizar a função

$$G_{A,b}(\omega, \lambda) = \frac{n \left\| (I - \omega A A_\lambda^\dagger) \mathbf{b} \right\|_2^2}{\left(\text{tr}(I - \omega A A_\lambda^\dagger) \right)^2}. \quad (\text{B.5})$$

Seja $0 < \omega < 1$ e $F_j = \text{diag}(1, \dots, 1, \sqrt{1 - \omega}, 1, \dots, 1) \in \mathbb{R}^{m \times m}$ em que $\sqrt{1 - \omega}$ aparece na j -ésima posição e, assim como no caso da GCV, procuramos uma solução para o problema de minimização

$$\mathbf{x} = \underset{\mathbf{x} \in \mathbb{R}^n}{\text{argmin}} \left\{ \|F_j(b - A\mathbf{x})\|_2^2 + \lambda^2 \|\mathbf{x}\|_2^2 \right\}. \quad (\text{B.6})$$

Neste caso a j -ésima observação é mantida (com peso $\sqrt{1 - \omega}$) se $0 < \omega < 1$, mas desaparece se $\omega = 1$. Uma conclusão similar é obtida se $\omega \geq 1$ e $F_j = \text{diag}(1, \dots, 1, \sqrt{1 - \omega}, 1, \dots, 1)$. Usando SVD da matriz A podemos escrever o denominador em B.5 como

$$\text{tr} \left(I - \omega A A^\dagger_\lambda \right) = \sum_{i=1}^n \frac{(1 - \omega)\sigma_i^2 + \lambda^2}{\sigma_i^2 + \lambda^2} + (m - n) \quad (\text{B.7})$$

$$= \sum_{i=1}^n (1 - \omega)f_i + \sum_{i=1}^n \frac{\lambda^2}{\sigma_i^2 + \lambda^2} + (m - n) \quad (\text{B.8})$$

onde os f_i são os fatores de filtro definidos anteriormente. Então, se $\omega < 1$ estamos adicionando um múltiplo da soma dos fatores de filtro ao traço do termo original, se $\omega > 1$ estamos subtraindo um múltiplo. O gráfico da função GCV também sofre mudanças quando ω deixa de assumir o valor 1. O denominador se torna zero para alguns valores maiores do que 1, ou seja, $\omega > 1$, nesse caso a função W-GCV tem um pólo. No artigo [9] podem ser encontradas informações a respeito de como, de maneira adaptativa, o coeficiente ω pode ser determinado.

B.2 Princípio da Discrepância

O método mais utilizado que baseia-se na estimativa da norma do erro e é o princípio da discrepância, atribuído a Morozov [38]. Se o problema mal posto é consistente no sentido de que $A\mathbf{x}_{\text{exato}} = \mathbf{b}_{\text{exato}}$ vale, então a ideia é simplesmente escolher o parâmetro de regularização λ tal que a norma do resíduo seja igual a uma cota superior δ para $\|e\|_2$, isto é, devemos determinar λ da equação não-linear

$$\|\mathbf{b} - A\mathbf{x}_\lambda\|_2 = \delta, \quad \|e\|_2 \leq \delta. \quad (\text{B.9})$$

Usando a SVD da matriz A na equação (1.19) temos que a norma do resíduo pode ser escrita como

$$\|r_\lambda\|_2^2 = \sum_{i=1}^r ((1 - f_i) u_i^T b) + \|b_\perp\|_2^2 \quad (\text{B.10})$$

e podemos perceber que a norma do resíduo é uma função crescente e monótona. Então, resolver (B.9) equivale a encontrar a intersecção entre a curva da norma do resíduo e a reta horizontal $z = \delta$.

O princípio da discrepância é um dos poucos métodos existentes na literatura que tem análise de erro, ao contrário dos critérios da GCV e da Curva-L que não possuem. Além disso, o método da discrepância pode ser ótimo no sentido de minimizar $\|\mathbf{x}_{\text{exato}} - \mathbf{x}_{\lambda_e}\|_2$ em que λ_e denota o parâmetro determinado pelo princípio da discrepância, quando a norma do erro e é conhecida e o erro e tiver certa estrutura [19]. Da teoria geral de regularização de Tikhonov [19] sabemos que um limitante superior para o erro na solução \mathbf{x}_{λ_e} é

$$\|\mathbf{x}_{\text{exato}} - \mathbf{x}_{\lambda_e}\|_2 = \mathcal{O}(\|e\|_2) \quad (\text{B.11})$$

em que \mathcal{O} é utilizado para descrever a cota

$$\|\mathbf{x}_{\text{exato}} - \mathbf{x}_{\lambda_e}\|_2 \leq 2\|A^\dagger\|_2 \|e\|_2 \quad (\text{B.12})$$

A grande dificuldade deste método é que precisamos de uma estimativa para a norma do erro e . Caso essa estimativa seja muito grande, podemos encontrar um parâmetro de regularização muito grande causando os inconvenientes já discutidos. O mesmo acontecendo com uma estimativa menor o que, neste caso, pode ser mais indesejável pelo fato dos erros serem dominantes na construção da solução.

B.3 Curva L

O método da Curva L escolhe como parâmetro de regularização o valor que maximiza a curvatura da curva parametrizada por

$\lambda \in [0, \infty)$ dada por

$$\mathcal{L}(\lambda) = \{(a, b) ; a = \log(\|\mathbf{r}_\lambda\|_2^2), b = \log(\|\mathbf{x}_\lambda\|_2^2)\} \quad (\text{B.13})$$

com \mathbf{x}_λ solução regularizada e $\mathbf{r}_\lambda = \mathbf{b} - A\mathbf{x}_\lambda$ o resíduo associado.

Este método surgiu em 1992 introduzido por P. C. Hansen [23] e é muito popular, sendo utilizado com sucesso em diversos problemas.

Essa escolha de λ baseia-se nas expressões para as normas de \mathbf{x}_λ e \mathbf{r}_λ obtidas com a SVD (ou GSVD) de A (ou (A, L))

$$\begin{aligned} \|\mathbf{x}_\lambda\|_2^2 &= \sum_{i=1}^r \left(f_i \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \right)^2 \\ \|\mathbf{r}_\lambda\|_2^2 &= \sum_{i=1}^r ((1 - f_i) \mathbf{u}_i^T \mathbf{b})_2^2 + \|\mathbf{b}_\perp\|_2^2. \end{aligned}$$

com \mathbf{b}_\perp como em (1.19) e na influência do ruído no vetor \mathbf{b} na forma da curva $\mathcal{L}(\lambda)$.

Considerando $\mathbf{b} = \mathbf{b}_{\text{exato}} + \mathbf{e}$ temos

$$\mathbf{u}_i^T \mathbf{b} = \mathbf{u}_i^T \mathbf{b}_{\text{exato}} + \mathbf{u}_i^T \mathbf{e}.$$

Dependendo do λ , são os coeficientes $\mathbf{u}_i^T \mathbf{b}_{\text{exato}}$ ou os coeficientes $\mathbf{u}_i^T \mathbf{e}$ que dominam. Então, a curva-L resultante consiste, essencialmente, de uma parte da curva-L para $\mathbf{b}_{\text{exato}}$ e outra parte para \mathbf{e} e em algum lugar existe uma faixa de valores que corresponde à transição entre as duas curvas. O desenho de uma curva-L genérica pode ser conferido na Figura B.1.

A região da curva indicada com o termo “Menos filtro” refere-se a valores de λ que resultam em fatores de filtro f_i próximos de 1. Neste caso, conforme a análise dos fatores de filtro feita na seção 1.4, a contribuição do erro não é filtrada o suficiente para garantir uma solução regularizada apropriada. Já a solução calculada com λ na região indicada com o termo “Mais filtro” pode não ter incorporado informação suficiente da solução do problema.

Baseado em considerações desse tipo, Hansen propôs escolher o parâmetro de regularização λ localizando a região em que se

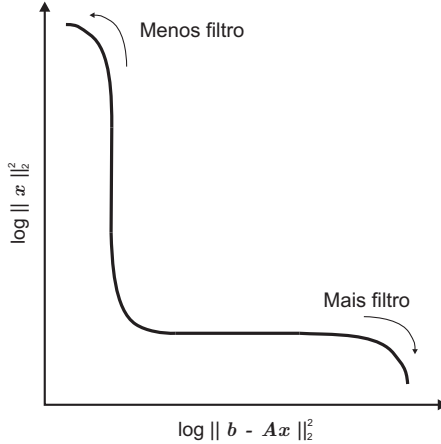


Figura B.1: Curva-L genérica

encontra a transição entre as curvas. Mais precisamente, o valor que maximiza a curvatura

$$\kappa(\lambda) = \rho\eta \frac{\rho\eta/|\eta'| - \lambda^2 - \lambda^4\eta}{(\rho^2 + \lambda^4\eta^2)^{3/2}}$$

em que $\rho = \|\mathbf{b} - A\mathbf{x}_\lambda\|_2$ e $\eta = \|\mathbf{x}_\lambda\|_2$.

O critério curva-L como método para a determinação do parâmetro de regularização de Tikhonov apresenta duas limitações. A primeira refere-se à reconstrução de soluções exatas muito suaves, isto é, soluções $\mathbf{x}_{\text{exato}}$ tal que os coeficientes da SVD $|\mathbf{v}_i^T \mathbf{x}_{\text{exato}}|$ decaem rápido para zero tal que a solução $\mathbf{x}_{\text{exato}}$ seja dominada pelas primeiras componentes da SVD. Para situações como estas o parâmetro encontrado pelo critério da curva-L normalmente é consideravelmente menor do que o parâmetro ótimo, entendemos por parâmetro ótimo como sendo aquele que $\frac{\|\mathbf{x}_\lambda - \mathbf{x}_{\text{exato}}\|_2}{\|\mathbf{x}_{\text{exato}}\|_2}$ seja o menor possível e, quanto mais suave for a solução, pior o parâmetro. Outra limitação refere-se ao fato de que a curva L pode apresentar mais de um “canto”. Detalhes a respeito das limitações da curva-L são apontados em [20, 27, 55].

Referências Bibliográficas

- [1] ADAMS, R. A.. *Sobolev Spaces*. New York: Academic Press, 1975.
- [2] BAKER, C. T. H.. *The Numerical Treatment of Integral Equations*, Oxford: Clarendon Press, pp. 665, 1977.
- [3] BAZÁN, F. S. V.. *Fixed-point iterations in determining the Tikhonov regularization parameter*, Inverse Problems, Vol. 24, 2008.
- [4] BAZÁN, F. S. V.; FRANCISCO, J. B.. *An improved fixed-point algorithm for determining the Tikhonov regularization parameter*, Inverse Problems, Vol. 25, No 4, 2009.
- [5] BAZÁN, F. S. V.; BORGES, L. S.. *GKB-FP: an algorithm for large-scale discrete ill-posed problems*. BIT Numerical Mathematics, DOI 10.1007/s10543-010-0275-3, 2010.
- [6] BJÖRCK, A.; GOLUB, G. H.. *Numerical methods for computing angles between linear subspaces*, Mathematics of Computation, Vol. 27, No 123, pp. 579-594, 1973.
- [7] BJÖRCK, A.; GRIMME, E.; VAN DOOREN, P.. *An implicit shift bidiagonalization algorithm for ill-posed systems of linear equations*, BIT Numerical Mathematics Vol. 34, pp. 510-534, 1994.

- [8] BJÖRCK, A. *Numerical Methods for Least Squares Problems*, Philadelphia: SIAM, 1996.
- [9] CHUNG, J.; NAGY, J. G.; O'LEARY, D. P.. *A Weighted-GCV method for Lanczos-Hybrid regularization*, Electronic Transaction on Numerical Analysis, Vol. 28, pp 149-167, 2008.
- [10] DAVIS, C.; KAHAN, W.. *The rotation of eigenvectors by a perturbation, III*, SIAM Journal on Numerical Analysis, Vol. 7, No 1, pp. 1-46, 1970.
- [11] DE MOOR, B.; ZHA, Z.. *A tree of generalizations of the ordinary singular value decomposition*, Linear Algebra and its Applications, Vol. 147, pp. 469-500, 1991.
- [12] DEMMEL, J.; KAHAN, W.. *Accurate singular values of bidiagonal matrices*, SIAM Journal on Scientific and Statistical Computing, Vol. 11, No 5, pp. 873-912, 1990.
- [13] ELDÉN, L.. *A weighted pseudoinverse, generalized singular values, and constrained least square problems*. BIT, Vol. 22, pp. 487-501, 1982.
- [14] ENGL, H. W.; HANKE, M; NEUBAUER, A.. *Regularization of Inverse Problems*, Dordrecht: Kluwer Academic Publishers, 1996.
- [15] GOLUB, G. H.; KAHAN, W.. *Calculating the singular values and pseudo-inverse of a matrix*, SIAM Journal on Numerical Analysis, Vol. 2, No 2, pp. 205-224, March, 1965.
- [16] GOLUB, G. H.; REINSCH, C.. *Singular value decomposition and least squares solution*, Numerische Mathematik, Vol. 14, No 5, pp. 403-420, 1970.
- [17] GOLUB, G. H.; HEATH, M. T.; WAHBA, G.. *Generalized cross-validation as a method for choosing a good ridge parameter*. Technometrics, Vol. 21, pp. 215-223, 1979.

- [18] GOLUB, G. H.; VAN LOAN, C. F.. *Matrix Computations*. 3. ed. Maryland: Johns Hopkins University Press, 1996.
- [19] GROETSCH, C. W.. *The theory of Tikhonov regularization for Fredholm equations of the first kind*, Research Notes in Mathematics, 105, Boston: Pitman, 1984.
- [20] HANKE, M.. *Limitations of the L-curve method in ill-posed problems*, BIT 36, pp. 287-301, 1996.
- [21] HANSEN, P. C.. *Regularization, GSVD and Truncated GSVD*, BIT 29, pp. 491-504, 1989.
- [22] HANSEN, P. C.. *The Discrete Picard Conditions For Discrete Ill-Posed Problems*, BIT 30, pp. 658-672, 1990.
- [23] HANSEN, P. C.. *Analysis of discrete ill-posed problems by means of the L-curve*, SIAM Review, Vol. 34, No 4, pp. 561-580, 1992.
- [24] HANSEN, P. C.. *Regularization tools: A MATLAB package for analysis and solution of discrete ill-posed problems.*, Numer. Algorithms 6, pp. 1-35 1994.
- [25] HANSEN, P. C.. *Regularization tools*, Disponível em: <http://www2.imm.dtu.dk/~pch/Regutools/>.
- [26] HANSEN, P. C.. *Rank-deficient and discrete ill-posed problems*, Philadelphia: SIAM, 1998.
- [27] HANSEN, P. C.. *The L-curve and its use in the numerical treatment of inverse problems*. Capítulo convidado em P. Johnston (Ed.), Computational Inverse Problems in Electrocardiology, Southhampton: WIT Press, pp. 119-142, 2001.
- [28] HANSEN, P. C.. *Oblique projections, pseudoinverses, and standard-form transformations*, Informatics and Mathematical Modelling, DTU, 2004.
- [29] HANSEN, P. C.; NAGY, J. G.; O'LEARY, D. P.. *Deblurring Images. Matrices, Spectra and Filtering*. SIAM, 2006.

- [30] HANSEN, P. C.. *Discrete inverse problems: insight and algorithm*, Philadelphia: SIAM, 2010.
- [31] JACOBSEN, M.; HANSEN, P. C.; SAUNDERS, M. A.. *Subspace preconditioned LSQR for discrete ill-posed problems*, BIT, 43, pp. 975-989, 2003.
- [32] JAIN, A. K.. *Fundamentals of Digital Image Processing*. Prentice-Hall, New Jersey: Englewood Cliffs, 1989.
- [33] JORDAN, C.. *Essai sur la géométrie à n dimensions*. Buletin de la Société Mathématique de France, Vol. 3, pp. 103-174, 1875.
- [34] KILMER, M. E.; HANSEN, P. C.; ESPAÑOL, M. I.. *A projection-based approach to general-form Tikhonov regularization*, SIAM J. Sci. Comput., Vol. 29, No. 1, p. 315-330, 2007.
- [35] KIRSCH, A.. *An Introduction to the Mathematical Theory of Inverse Problems*, Applied Mathematical Sciences Vol. 120, New York: Springer, 1996.
- [36] LAWSON, C.; HANSON, R.. *Solving Least Squares Problems*. Prentice-Hall Inc., New Jersey: Englewood Cliffs, 1974.
- [37] MEYER, C. D.. *Matrix Analysis and Applied Linear Algebra*. Philadelphia: SIAM, 2000.
- [38] MOROZOV, V. A.. *On the solution of functional equations by the method of regularization*, Soviet Math. Dokl, Vol. 7, pp. 414-417, 1966.
- [39] MOROZOV, V. A.. *The errors principle in the solution of operational equations by the regularization method*, USSR Comput. Math. and Phys, Vol. 8, No. 2, pp. 63-87, 1968.
- [40] PAIGE, C. C.; SAUNDERS, M. A.. *Towards a generalized singular value decomposition*, SIAM Journal on Numerical Analysis, Vol. 18, No 3, pp. 398-405, 1981.

- [41] PAIGE, C. C.; SAUNDERS, M. A.. *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Softw., Vol. 8, No. 1, pp 43-71, 1982.
- [42] PAIGE, C. C.; WEI, M.. *History and generality of the CS decomposition*, Linear Algebra and its Applications, Vol. 208/209, pp. 303-326, 1994.
- [43] PARK, S. C.; PARK, M. K.; KANG, M. G.. *Super-resolution image reconstruction: a technical overview*, IEEE Signal Processing Magazine, Vol. 20, No 3, pp. 21-36, 2003.
- [44] PHILLIPS, D. L.. *A technique for the numerical solution of certain integral equation of the first kind*. Journal of the ACM, Vol. 9, pp. 84-97, 1962.
- [45] REGIŃSKA, T.. *A regularization parameter in discrete ill-posed problems*, SIAM Journal on Scientific Computing, Vol. 17, No 3, pp. 740-749, 1996.
- [46] SAAD, Y.; SCHULTZ, M. H.. *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*. SIAM Journal on Scientific and Statistical Computing, Vol. 7, pp. 856-869, 1986.
- [47] SAUNDERS, M. A.. *Computing projections with LSQR*, BIT, 37, pp. 96-104, 1997.
- [48] SILVA NETO, A. J.; MOURA NETO, F. D.. *Problemas Inversos: Conceitos fundamentais e aplicações*, Rio de Janeiro: UERJ, 2005.
- [49] STEWART, G. W.. *On the perturbation of pseudo-inverse, projections, and linear least squares problems*. SIAM Review, Vol. 19, No 4, pp. 634-662, 1977.
- [50] STEWART, G. W.. *Computing the CS decomposition of a partitioned orthogonal matrix*. Numerische Mathematik, Vol. 40, No 3, pp. 297-306, 1982.

- [51] STEWART, G. W.. *A method for computing the generalized singular value decomposition*. Matrix Pencils, ed. B. Kågström and A. Ruhe, New York: Springer-Verlag, pp. 207-220, 1983.
- [52] STEWART, G. W.. *On the early history of the singular value decomposition*. SIAM Review, Vol. 35, No 4, pp. 551-565, 1993.
- [53] TIKHONOV, A. N.. *Solution of incorrectly formulated problems and the regularization method*, Soviet Math. Dokl., Vol. 4, pp. 1035-1038, 1963.
- [54] VAN LOAN, C. F.. *Generalizing the Singular Value Decomposition*, SIAM Journal on Numerical Analysis, v. 13, n^o 1, pp. 76-83, March, 1976.
- [55] VOGEL, C. R.. *Non-convergence of the L-Curve regularization parameter selection method*, Inverse Problems, Vol. 12, pp. 535-547, 1996.
- [56] VOGEL, C. R.. *Computational Methods for Inverse Problems*, Frontiers in Applied Mathematics, Philadelphia: SIAM, 2002.
- [57] ZIBETTI, M. W.; BAZAN, F. S. V.; MAYER, J.. *Estimation of the parameter in regularized simultaneous Super-resolution*. Pattern Recognition Letters, DOI 10.1016/j.patrec.2009.12.009, 2010.