Universidade Federal de Santa Catarina

Curso de Pós-Graduação em Matemática e Computação Científica

Lanc-FP: Um algoritmo para problemas discretos mal-postos de grande porte

Leonardo Silveira Borges

Orientador: Prof. Dr. Fermín S. V. Bazán

Florianópolis

Março de 2009

Universidade Federal de Santa Catarina

Curso de Pós-Graduação em Matemática e Computação Científica

Lanc-FP: Um algoritmo para problemas discretos mal-postos de grande porte

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.

Leonardo Silveira Borges

Florianópolis

Março de 2009

Lanc-FP: Um algoritmo para problemas discretos mal-postos de grande porte

por

Leonardo Silveira Borges

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

Clóvis Caesar Gonzaga

Coordenador

Comissão Examinadora

Prof. Dr. Fermín S. V. Bazán (UFSC-Orientador)

Prof. Dr. José Mario Martinez (UNICAMP)

Prof. Dra. Maria Cristina de Castro Cunha (UNICAMP)

Prof. Dr. Juliano de Bem Francisco (UFSC)

Florianópolis, Março de 2009.

Agradecimentos

Agradeço à Deus acima de tudo.

Agradeço ao professor Fermín, pela sua orientação. Todas as conversas, dicas e sugestões foram fundamentais para a realização desta pesquisa e meu aperfeiçoamento profissional.

Aos professores José Mario, Maria Cristina e Juliano, por terem aceito avaliar este trabalho e por contribuírem com suas sugestões e críticas para aperfeiçoar esta dissertação.

Aos demais colegas e professores que me acompanharam durante mais esta etapa.

Aos meus pais por terem colaborado para que eu chegasse até aqui e pelo exemplo de vida.

E por fim, especialmente à minha noiva por ter me apoiado e incentivado sempre.

Resumo

Problemas discretos mal-postos precisam ser regularizados para serem resolvidos estavelmente. Dentre vários métodos de regularização existentes na literatura, um dos mais utilizados é devido a Tikhonov [49] e a sua eficiência depende da escolha do parâmetro de regularização. A curva-L de Hansen [29, 30], o princípio da discrepância de Morozov [41] e a Validação Cruzada Generalizada de Golub, Heath e Wahba [20] são métodos que buscam determinar um bom parâmetro de regularização. Recentemente um algoritmo de ponto-fixo por Bazán [2] e em seguida uma melhoria por Bazán e Francisco [3] tem mostrado excelentes resultados, tanto de cunho teórico como prático.

Problemas de grande porte, de modo geral, são resolvidos por métodos iterativos. O algoritmo LSQR de Paige e Saunders [44, 45] é baseado em projeções em subespaços de Krylov e, assim como muitos métodos de projeção, captura boa parte das informações relevantes do problema nas primeiras iterações. Caso as iterações não sejam interrompidas, as novas soluções iteradas são dominadas pelo ruído nos dados e como consequência existe um deterioramento das iteradas. Para contornar a dificuldade inerente a esta abordagem, um critério de parada faz-se necessário.

Apresentamos um algoritmo para problemas mal-postos discretos de grande porte chamado de Lanc-FP [4], o qual resulta da combinação do algoritmo de ponto-fixo com o método LSQR. A ideia fundamental é estimar o parâmetro de Tikhonov no problema projetado construído por LSQR usando o algoritmo do ponto-fixo, e então prosseguir com as iteradas até as mesmas estacionarem. Desenvolvemos a parte teórica do algoritmo e entre outro resultados, apresentamos a demonstração de que as iteradas realmente estabilizam, o qual é o resultado mais importante deste trabalho e único para os algoritmos na área.

Por fim, os resultados teóricos são avaliados na obtenção de soluções numéricas para equações integrais e restauração de imagens.

Abstract

Discrete ill-posed problems need to be regularized. Among several existing regularization methods, perhaps the most used is due to Tikhonov [49]. For this method to be successful the choice of the regularization parameter is crucial and several parameter-choice methods have been proposed; these include the L-curve by Hansen [29, 30], the discrepancy principle by Morozov [41] and the Generalized Cross Validation by Golub, Heath and Wahba [20]. Recently a fixed-point algorithm by Bazán [2] and an improved version of it by Bazán and Francisco [3], have proved to yield excellent results.

Large problems, in general, are addressed by iterative methods. The LSQR algorithm due to Paige and Saunders [44, 45] is based on projections onto Krylov subspaces, and like several projection methods, captures the most relevant information of the solution in the first few iterations. If the iterates are not stopped, the subsequent iterates are dominated by the noise in the data. As a consequence, in order to LSQR to construct regularized approximate solutions, the iterates have to be stopped before they start to deteriorate.

We propose an algorithm for large scale discrete ill-posed problems referred to as Lanc-FP [4], which arises as a result of combining the fixed-point algorithm with LSQR. Basically, the idea is to estimate the Tikhonov regularization parameter associated to the projected problem constructed by LSQR by the fixed-point algorithm, and then proceed with the iterates until they stagnate. Among a number of theoretical results, we demonstrate, in theory and practice, that the iterates truly stagnate, which is the most important contribution of this work. Such a result is new. Finally, the theoretical results are illustrated numerically by solving first kind integral equations and image restoration problems.

Lista de Figuras

Lista de Tabelas

Lista de Símbolos

B_k	Matrix bidiagonal inferior obtida após k passos da bidiagonalização de Lanczos.
DP	Princípio da discrepância (Discrepance Principle).
FP	Algoritmo de Ponto-Fixo (Fixed-Point).
GCV	Validação Cruzada Generalizada (Generalized Cross-Validation).
GSVD	SVD Generalizada (Generalized-SVD).
LBD	Bidiagonalização de Lanczos.
LSQR	Algoritmo de projeção para problemas de grande porte.
SVD	Decomposição em valores singulares (Singular Value Decomposition).
TGSVD	GSVD Truncada (Truncated-GSVD).
TSVD	SVD Truncada (Truncated-SVD).
W-GCV	Validação Cruzada Generalizada com peso (Weighted-GCV).
α_i,β_i	Elementos da diagonal principal e da diagonal inferior, respectivamente, da matriz B_k.
$b_\perp$	Componente do vetor b que não pertence ao espaço coluna da matriz A.
$\delta_0^{(k)}$	Norma-2 da componente do vetor $\beta_1 e_1$ que não pertence ao espaço coluna da matriz B_k
e_1	Vetor canônico $[1, 0, \ldots, 0]^T$.
$\phi_\mu(\lambda)$	Função do algoritmo de ponto-fixo.
$\phi_\mu^{(k)}(\lambda)$	Função do algoritmo de ponto-fixo para o problema projetado.
$\mathbf{x}(\lambda)$	Função que representa a norma do resíduo.
$\mathbf{y}(\lambda)$	Função que representa a norma da solução.
y_k	Solução para o problema projetado (sem regularização).
$y_{k,\lambda}$	Solução para o problema projetado (com regularização).

Conteúdo

Introdução

A resolução numérica de problemas discretos mal-postos do tipo

$$x = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \|b - Ax\|_2^2$$

em que a matriz A é mal-condicionada e o vetor b contém erros, tem sido motivo de extensas pesquisas ao longo das últimas décadas, pois a solução usual de mínimos quadrados não apresenta aplicações práticas por estar contaminada por ruídos. Para contornar tal dificuldade usualmente utilizamos métodos de regularização a fim de determinar uma solução que se aproxime da solução exata e sem ruídos.

A forma mais simples da regularização de Tikhonov [49] substitui o problema usual de mínimos quadrados por

$$x_\lambda = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \{\|b - Ax\|_2^2 + \lambda^2 \|x\|_2^2\}$$

em que o parâmetro de regularização $\lambda > 0$ deve ser convenientemente escolhido.

Na literatura existem diversos métodos para encontrar tal parâmetro, dentre eles podemos citar a curva-L de Hansen [28, 29], a Validação Cruzada-Generalizada (GCV) de Golub, Heath e Wahba [20], o princípio da discrepância por Morozov [41] e recentemente um algoritmo de ponto-fixo por Bazán [2]. Outros métodos podem ser encontrados em [14, 26, 38, 39].

O critério da curva-L e a GCV são métodos que têm sido utilizados em diversos problemas com sucesso. Os mesmos são bastantes conhecidos no meio científico e vem sugerindo novas propostas como a W-GCV por Chung, Nagy e O'Leary [14] e a utilização de *ribbons* [9, 11, 12] para uma aproximação da curva-L.

Cada um destes métodos apresenta propriedades e características próprias, além, é claro, de dificuldades. A dificuldade do critério da curva-L está no caso da curva-L apresentar mais de um "canto", outras limitações podem ser encontradas em Hanke [24] e Vogel [50]. Com a GCV pode ocorrer que a função GCV tenha seu minimizador numa região muito plana e neste caso a determinação do parâmetro λ pode ser uma tarefa difícil para qualquer método de minimização. O princípio da discrepância é um método que necessita de uma estimativa para a norma do erro que está contido nos dados b, o que muitas vezes não está disponível. O algoritmo de ponto-fixo por Bazán [2] apresenta dificuldades quando existem múltiplos pontos-fixos (o que implica a existência de mais de um "canto" na curva-L), porém esta dificuldade já foi contornada em Bazán e Francisco [3]. Outra consideração que deve ser feita com relação ao algoritmo de ponto-fixo por Bazán [2] é que este calcula muito consistentemente o parâmetro de regularização λ se comparado com a curva-L e a GCV, quando o vetor de dados b está contaminado por ruído branco [2, 3].

A regularização de Tikhonov pertence à classe dos método de penalidade. Existem outras duas classes de métodos para regularizar um problema discreto mal-posto, a saber: os métodos de projeção e os métodos híbridos (combinações entre um método de penalidade com um de projeção).

Dentre os métodos de projeção podemos citar a TSVD [30], a TGSVD [26], GMRES [10] e o LSQR de Paige e Saunders [44, 45], este último é analiticamente idêntico ao Gradiente Conjugados [30]. Tais métodos podem ser considerados métodos iterativos, pois vão, iteração por iteração, construindo a solução através de uma sequência de soluções x_k, $k = 1, 2, \ldots$, e são usualmente utilizados em problemas cuja dimensão inviabiliza o cálculo da SVD devido ao alto custo computacional. Outra característica é que logo nas primeiras iterações boa parte das informações relevantes do problema são capturadas, porém se as iterações persistirem as novas componentes passam a conter mais contribuições do erro nos dados, gerando uma desestabilização na solução tornando-a inútil.

As dificuldades em se determinar bons critérios de parada para os métodos de projeção podem ser parcialmente contornadas combinando algum método de regularização de penalidade (Tikhonov, por exemplo) em cada iteração, neste caso como a dimensão do sistema projetado é consideravelmente menor do que a dimensão do sistema original, o cálculo da SVD pode ser realizado. Desta maneira as iterações tendem a estabilizar

as soluções iteradas à medida que em cada iteração seja fornecido parâmetros quase ótimos de regularização, como por exemplo o algoritmo LSQR-Tik [35, 36, 37] que utiliza a curva-L para determinar o parâmetro de regularização em cada iteração do método de projeção LSQR [44, 45]. Outras propostas podem ser encontrados em [7, 11, 14].

Desta forma, identificamos que há uma certa carência para a resolução de problemas de grande porte de maneira eficaz. Com base neste contexto e na recente proposta do algoritmo de ponto-fixo devido a Bazán [2] para a determinação do parâmetro de regularização de Tikhonov é que estamos propondo um novo algoritmo para problemas de grande porte, sendo este o objetivo desta pesquisa. Assim, segue um esquema da organização desta dissertação e a sucinta descrição de cada capítulo.

O trabalho está dividido em cinco capítulos. Nos três primeiros apresentamos uma parte da teoria existente sobre regularização, no quarto capítulo propomos um algoritmo para problemas de grande porte e no quinto apresentamos resultados numéricos obtidos em problemas testes encontrados na literatura [31, 42]. A toolbox RegularizationTools de Hansen [31] consiste numa seleção de rotinas que incluem métodos para encontrar soluções regularizadas para problemas discretos mal-postos e diversos problemas testes como *phillips, foxgood, wing, heat, baart*, etc, que serão utilizados ao longo deste trabalho e a toolbox RestoreTools [42] possui rotinas no mesmo caminho das encontradas na toolbox RegularizationTools.

No primeiro capítulo apresentamos a definição do que é um problema mal-posto e como eles podem surgir naturalmente, realizando alguns comentários sobre operadores compactos. Relembramos a decomposição em valores singulares de uma matriz A e como ela pode ser utilizada como um método de regularização. A regularização de Tikhonov é discutida brevemente para que em seguida seja realizada uma exposição de alguns métodos que buscam, cada um com suas propriedades, determinar um parâmetro de regularização satisfatório. No final deste capítulo fazemos alguns comentários sobre métodos iterativos encontrados na literatura.

No segundo capítulo expomos uma revisão detalhada da teoria do algoritmo de ponto-fixo por Bazán [2] e por Bazán e Francisco [3], exibindo exemplos de problemas em que há ponto-fixo e exemplos em que os mesmos estão ausentes.

Em seguida, no capítulo três, discutimos com detalhes o algoritmo LSQR desenvolvido

por Paige e Saunders [44, 45] e mostramos como este algoritmo pode ser útil quando aplicado à regularização de Tikhonov. Além disso discutimos o fato do LSQR possuir a propriedade de "semi-convergência", no sentido de que após o número ótimo de passos, as iteradas x_k tendem a estacionar durante alguns passos nos quais a qualidade das iteradas é quase ótima, e logo deterioram como uma consequência das iteradas começarem a serem dominadas pelo ruído. Em particular, mostramos que esta característica negativa pode ser contornada com um bom parâmetro de regularização.

No capítulo quatro surge a contribuição desta pesquisa, onde apresentamos um algoritmo para problemas de grande porte baseado nos trabalhos de Bazán [2], e Bazán e Francisco [3]. Discutimos propriedades de existência de ponto-fixo, resultados sobre convergência e uma propriedade a respeito da estabilização da solução construída pelo algoritmo. A estabilização de soluções regularizadas tem sido observada recentemente por alguns autores, veja por exemplo, Hansen [30], Chung *et. al.* [14], *apenas* através de experimentos numéricos. Aqui, a tal estabilização é *demonstrada teorica e analiticamente*.

Por fim, no capítulo cinco apresentamos e tecemos comentários sobre os resultados obtidos em simulações numéricas com sistemas de 4096 variáveis que surgem da discretização de equações integrais. Também, para fins de comparação com outros métodos baseados na SVD, comparamos os resultados do algoritmo com aqueles obtidas pelos métodos da curva-L, GCV e discrepância. Além disso, abordamos o problema de restauração de imagens usando imagens em tons de cinza e imagens coloridas que facilmente podem passar de 50.000 variáveis, comparando os resultados do método com os obtidos pelo método da Validação Cruzada Generalizada com peso (W-GCV) devido a Chung, Nagy e O'Leary [14].

No final do trabalho apresentamos as considerações finais e um apêndice sobre autovalor generalizado, uma vez que no capítulo um teremos que utilizar a GSVD, que é uma generalização da SVD para um pencil associado às matrizes A e L.

Capítulo 1

Problemas mal-postos e Regularização

O conceito de problema bem-posto remete à Hadamard que, em 1902, definiu que um problema é bem-posto quando

1. Existe solução.

2. A solução é única.

3. A solução depende continuamente dos dados.

Um problema é mal-posto quando uma das condições acima não é satisfeita.

Este conceito se aplica, por exemplo, à equação integral de Fredholm. A equação integral linear de Fredholm pode ser escrita como

$$h(x)f(x) + \int_a^b K(x,y)f(y)dy = g(x) \tag{1.1}$$

em que as funções $h(x)$, $K(x,y)$ e $g(x)$ são limitadas e geralmente contínuas. Essas equações podem ser classificadas em três tipos: (a) se $h(x) \equiv 0$ a equação é de primeira espécie; (b) se $h(x) \neq 0$ para $a \leqslant x \leqslant b$ a equação é de segunda espécie; (c) se $h(x) = 0$ para algum subconjunto $I \subsetneqq [a,b]$ então a equação é de terceira espécie. Se o domínio de integração for infinito ou se o núcleo $K(x,y)$ não for limitado, a equação é singular.

A seguir iremos fazer uma breve discussão sobre propriedades gerais das equações integrais de primeira espécie.

1.1 Operadores Compactos

O objetivo desta seção é discutir alguns resultados básicos que tratam de operadores compactos entre espaços de Hilbert tendo em vista que as equações integrais são fontes de muitos problemas mal-postos. Os resultados que serão expostos aqui são resultados básicos da análise funcional e como o objetivo desta pesquisa não está relacionado com o comportamento analítico de equações integrais em espaços de Hilbert, as demonstrações serão omitidas.

Seja $\mathcal{H}$ um espaço de Hilbert e $\mathcal{S} \subset \mathcal{H}$ um subconjunto, então $\overline{\mathcal{S}}$ denotará o fecho de $\mathcal{S}$ e $\mathcal{S}^{\perp}$ será o complemento ortogonal de $\mathcal{S}$, isto é

$$\mathcal{S}^{\perp} = \{y \in \mathcal{H} / < x, y >= 0 \ , \ \forall \ x \in \mathcal{S}\} \tag{1.2}$$

em que $< \cdot, \cdot >$ denota o produto interno do espaço de Hilbert $\mathcal{H}$ e $\| \cdot \|$ denotará a norma neste espaço.

Se T é um operador linear contínuo entre dois espaços de Hilbert $\mathcal{H}_1$ e $\mathcal{H}_2$, então seu adjunto será denotado por T^* e definido por $< Tx, y >=< x, T^*y > \forall \ x \in \mathcal{H}_1$ e $\forall \ y \in \mathcal{H}_2$. A imagem $Im(T)$ e o núcleo $\mathcal{N}(T)$ de um operador linear T com domínio $D(T)$ são definidos por

$$Im(T) = \{Tx \ / \ x \in D(T)\}, \tag{1.3}$$

$$\mathcal{N}(T) = \{x \in D(T) \ / \ Tx = 0\}. \tag{1.4}$$

O teorema a seguir estabelece uma conexão entre estes conceitos fundamentais.

Teorema 1.1. *Se $T : \mathcal{H}_1 \to \mathcal{H}_2$ for um operador linear contínuo entre dois espaços de Hilbert $\mathcal{H}_1$ e $\mathcal{H}_2$, então $Im(T)^{\perp} = \mathcal{N}(T^*)$ e $\mathcal{N}(T)^{\perp} = \overline{Im(T^*)}$.*

Para operadores lineares sabemos que um operador T é contínuo se, e só se, é limitado, isto é,

$$\|T\| = \sup_{\|x\|=1} \|Tx\| < \infty \tag{1.5}$$

e que o espectro de um operador linear $T : \mathcal{H} \to \mathcal{H}$ é o conjunto de números complexos

$\sigma(T)$ dado por

$$\sigma(T) = \{\lambda \in \mathbb{C} \ / \ T - \lambda I \text{ não tem inversa limitada}\} \tag{1.6}$$

em que I denota o operador identidade no espaço de Hilbert $\mathcal{H}$. E o raio espectral do operador T é o número real $|\sigma(T)|$ tal que

$$|\sigma(T)| = \sup\{|\lambda| \ / \ \lambda \in \sigma(T)\}. \tag{1.7}$$

Se T for um operador limitado, então $\sigma(T)$ é um conjunto fechado e $|\sigma(T)| \leqslant \|T\|$, logo $\sigma(T)$ é um conjunto compacto.

Um operador $T : \mathcal{H} \to \mathcal{H}$ é auto-adjunto se $T = T^*$. Se o operador T for linear, limitado e auto-adjunto, então $\sigma(T)$ é um conjunto não vazio de números reais. O próximo teorema é conhecido como fórmula do raio espectral.

Teorema 1.2. *Se $T : \mathcal{H} \to \mathcal{H}$ for um operador limitado e auto-adjunto então $|\sigma(T)| = \|T\|$.*

Um operador linear $K : \mathcal{H}_1 \to \mathcal{H}_2$ entre dois espaçoes de Hilbert $\mathcal{H}_1$ e $\mathcal{H}_2$ é compacto se o conjunto $\overline{K(B)}$ for compacto para cada subconjunto limitado $B \in \mathcal{H}_1$. Um operador compacto é contínuo.

Um número complexo λ é um autovalor do operador linear $T : \mathcal{H} \to \mathcal{H}$ se $Tx = \lambda x$ para algum vetor não nulo x (autovetor) associado a λ. Obviamente todo autovalor do operador T é um elemento do conjunto $\sigma(T)$ e se T for um operador auto-adjunto então autovetores associados a autovalores distintos são ortogonais.

Operadores compactos auto-adjuntos tem um espectro particularmente simples: cada elemento não nulo do espectro $\sigma(K)$ é um ponto isolado que é um autovalor do operador K. Para cada autovalor λ de um operador compacto auto-adjunto K, o autoespaço associado a λ, isto é, o conjunto $\mathcal{N}(K - \lambda I)$, tem dimensão finita e os autovalores formam uma sequência que converge para zero. O próximo teorema é o teorema espectral para operadores compactos auto-adjuntos.

Teorema 1.3. *Suponha que $K : \mathcal{H} \to \mathcal{H}$ seja um operador linear compacto auto-adjunto com autovalores $\lambda_1, \lambda_2, \ldots$ (repetidos de acordo com a dimensão do autoespaço associado)*

e autovetores ortonormais associados $w_1, w_2, \ldots$, *então para qualquer* $x \in \mathcal{H}$ *vale*

$$Kx = \sum_n \lambda_n < x, w_n > w_n. \tag{1.8}$$

A soma do teorema acima pode ser finita ou infinita dependendo se o operador K tem um número finito ou infinito de autovalores. Se o operador K tem apenas um número finito de autovalores então o operador K tem posto finito, para este caso o teorema espectral mostra que o conjunto $Im(K)$ tem dimensão finita e os autoespaços geram este conjunto.

Seja $K : \mathcal{H}_1 \to \mathcal{H}_2$ um operador compacto, então $K^*K : \mathcal{H}_1 \to \mathcal{H}_1$ é um operador compacto auto-adjunto e qualquer autovalor β do operador K^*K satisfaz

$$\beta =< \beta x, x >=< K^*Kx, x >= \|Kx\|^2 \geqslant 0 \tag{1.9}$$

se x for um autovetor associado com norma 1. Disto segue que os autovalores de K^*K podem ser enumerados por $\lambda_1^2 \geqslant \lambda_2^2 \geqslant \ldots$. Se designarmos por $v_1, v_2, \ldots$, uma sequência de autovetores ortonormais e definirmos

$$\nu_n = \lambda_n^{-1} \quad \text{e} \quad u_n = \nu_n K v_n, \tag{1.10}$$

então $\{u_n\}$ é uma sequência de vetores ortonormais em $\mathcal{H}_2$ e

$$\nu_n K^* u_n = v_n. \tag{1.11}$$

Mais ainda, pode ser mostrado usando o teorema espectral que $\{u_n\}$ é um conjunto ortonormal completo para $\overline{Im(K)} = \mathcal{N}(K^*)^\perp$ e $\{v_n\}$ é um conjunto ortonormal completo para $\overline{Im(K^*)} = \mathcal{N}(K)^\perp$. A sequência $\{u_n, v_n; \nu_n\}$ é chamada de sistema singular para o operador K e alguns autores chamam a tripla $\{u_n, v_n; \nu_n\}$ de expansão em valores singulares e os valores singulares são $\sigma_n = \nu_n^{-1}, \ \forall n$.

Agora que vimos um pouco sobre operadores compactos, vamos considerar o espaço de Hilbert $L^2(\Omega)$ de funções reais com Ω um conjunto qualquer. Seja o operador compacto

linear $K : L^2(\Omega) \to L^2(\Omega)$ definido por

$$Kx = \int_\Omega k(\cdot, t)x(t)dt. \tag{1.12}$$

À luz da teoria de operadores compactos vista anteriormente, segue que o operador K admite uma expansão em valores singulares. Isto quer dizer que existe uma sequência não-crescente de valores singulares positivos $\sigma_j, j = 1, 2, \ldots$, com funções singulares correspondentes u_j e v_j, tal que

$$Kv_j = \sigma_j u_j, \quad K^* u_j = \sigma_j v_j, \quad j = 1, 2, \ldots. \tag{1.13}$$

Vamos expandir a função g de (1.1) numa série convergente

$$g = \sum_{j=1}^{\infty} < u_j, g > u_j. \tag{1.14}$$

Com esta expansão temos a condição de Picard.

Condição de Picard: Uma função g dada por (1.14) pertence a $Im(K)$ se, e somente se,

$$\sum_{j=1}^{\infty} \left(\frac{< u_j, g >}{\sigma_j} \right)^2 < \infty. \tag{1.15}$$

Para uma função arbitrária $g \in L^2(\Omega)$, os coeficientes $< u_j, g >$ formam uma sequência em l^2, isto é, eles tenderão a zero um pouco mais rápido do que $j^{-1/2}$, mas se $g \in Im(K)$ então a condição de Picard nos diz que os coeficientes $< u_j, g >$ convergem para zero mais rápidos do que $\sigma_j j^{-1/2}$.

Se $g \in Im(K)$, então

$$x = K^{-1}g = \sum_{j=1}^{\infty} \frac{< u_j, g >}{\sigma_j} v_j \tag{1.16}$$

em que a convergência da expansão acima é garantida pela condição de Picard e o símbolo K^{-1} representa o operador inverso.

Se $g \notin Im(K)$, ainda podemos aproximar a função g pela função g_k obtida pelo

truncamento da expansão (1.14) em k termos

$$g_k = \sum_{j=1}^{k} < u_j, g > u_j \tag{1.17}$$

e obviamente a função g_k satisfaz a condição de Picard para todo $k = 1, 2, \ldots$, e temos

$$K^{-1} g_k = \sum_{j=1}^{k} \frac{< u_j, g >}{\sigma_j} v_j. \tag{1.18}$$

Disto podemos concluir que se $k \to \infty$ então $g_k \to g$, mas

$$\|K^{-1} g_k\| \to \infty, \quad k \to \infty. \tag{1.19}$$

Por esta falta de estabilidade é que equações integrais de primeira espécie são consideradas problemas mal-postos. Mais detalhes podem ser encontrados em [22, 25] e livros de análise funcional.

1.2 Problemas discretos mal-postos

Quando consideramos estudar uma equação linear não-singular de primeira espécie

$$\int_a^b K(x, y) f(y) dy = g(x) \quad (a \leqslant x \leqslant b) \tag{1.20}$$

e temos que resolvê-la numericamente, a discretização geralmente resulta num problema de minimização

$$x = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \|b - Ax\|_2^2 \tag{1.21}$$

em que a matriz $A \in \mathbb{R}^{m \times n}$, $m \geqslant n$, apresenta um número de condição elevado e os valores singulares decaem para zero sem que haja um salto notório neste decaimento e $b \in \mathbb{R}^m$ um vetor que está contaminado por erros, ou seja, $b = b^{\text{exato}} + e$, com b^{exato} sendo o vetor sem perturbações desejado e desconhecido. Em situações como esta a solução de mínimos quadrados, $x_{LS} = A^{\dagger} b$, em que $A^{\dagger}$ denota a pseudoinversa da matriz A, não tem nenhuma relação com a solução do problema e não tem utilidade prática por estar completamente

dominada pelos erros. Hansen [27] definiu esse tipo de problema como sendo *Problema Discreto Mal-Posto.*

Para contornar tais dificuldades podemos restringir o espaço no qual se encontra a solução ou podemos substituir o problema (1.21) por algum problema que esteja "próximo" do problema original de modo que esta aproximação seja menos susceptível à pequenas variações, dentre outras possibilidades.

Independente da abordagem, essas técnicas são conhecidas como técnicas de regularização e visam atenuar da melhor maneira possível os erros das mais diversas naturezas, podendo os mesmos serem de medições, ou de arredondamentos/truncamentos, ou até mesmo uma má formulação do problema real.

Uma das ferramentas mais poderosas para analisarmos a equação (1.21) e que produz um método de regularização é a SVD da matriz A.

Teorema 1.4. *(SVD) Seja $A \in \mathbb{R}^{m\times n}$ uma matriz. Então existem matrizes ortogonais $U \in \mathbb{R}^{m\times m}$ e $V \in \mathbb{R}^{n\times n}$ tais que*

$$U^T AV = \text{diag}(\sigma_1, \ldots, \sigma_p) \in \mathbb{R}^{m\times n}, \quad p = \min\{m, n\} \tag{1.22}$$

e os valores singulares σ_i, $i = 1, \ldots, p$, ordenados de modo não-crescente

$$\sigma_1 \geqslant \ldots \geqslant \sigma_p \geqslant 0.$$

Como dito, a SVD pode ser utilizada como um método de regularização; neste caso a solução de (1.21) é

$$x = \sum_{i=1}^{r} \frac{u_i^T b}{\sigma_i} v_i \quad , \quad r = \text{posto}(A). \tag{1.23}$$

Sendo $b = b^{\text{exato}} + e$ temos

$$x = \sum_{i=1}^{r} \frac{u_i^T b}{\sigma_i} v_i = \sum_{i=1}^{r} \left(\frac{u_i^T b^{\text{exato}}}{\sigma_i} v_i + \frac{u_i^T e}{\sigma_i} v_i \right) \tag{1.24}$$

e podemos perceber que para valores singulares pequenos os coeficientes $\frac{u_i^T e}{\sigma_i}$ são grandes fazendo com que a parcela do erro seja dominante, tornando assim, esta abordagem inútil.

Se a série for truncada em $s < r$, com s próximo de r, podemos amenizar o efeito do

erro e no cálculo da solução x, porém, se s for pequeno, deixamos de capturar informações importantes do problema. Disso segue que a escolha do 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*), em outras palavras, a TSVD é um método de regularização cuja solução é escrita da forma

$$x_s = \sum_{i=1}^{s} \frac{u_i^T b}{\sigma_i} v_i \tag{1.25}$$

em que s é o parâmetro de regularização que deve ser determinado.

Para ilustrar, na figura 1.1 temos quatro soluções para o problema *baart*, cuja solução exata é representada pela linha pontilhada. Podemos perceber que o simples fato de termos adicionado o termo $\frac{u_5^T b}{\sigma_5} v_5$ destruiu completamente a solução; neste caso o parâmetro de regularização pode ser $s = 3$ uma vez que o erro relativo $\|x_3 - x^{\text{exato}}\|_2 / \|x^{\text{exato}}\|_2$ é o menor.

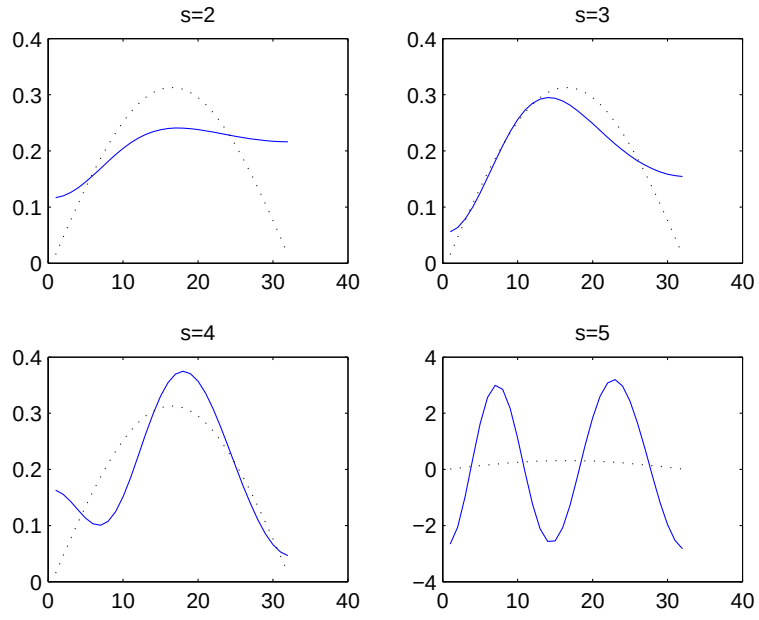

Figura 1.1: Solução x_s para $s = 2, 3, 4, 5$ para o problema *baart* com dimensão $n = 32$ e erro relativo de 1% nos dados (vetor b).

1.3 Regularização de Tikhonov

A regularização de Tikhonov surgiu em 1963 quando Tikhonov [49] considerou o problema $Kf = g$ em que f, g são funções e K um operador integral.

Tikhonov propôs substituir o problema

$$f = \underset{f \in \mathcal{H}}{\operatorname{argmin}} \|g - Kf\|_2^2 \tag{1.26}$$

por

$$f_\lambda = \underset{f \in \mathcal{H}}{\operatorname{argmin}} \{\|g - Kf\|_2^2 + \lambda^2 \Omega(f)\} \tag{1.27}$$

em que $\Omega(f) = \displaystyle\int_a^b [v(s)f(s)^2 + w(s)f'(s)^2)]ds$ com v, w funções de peso positivas e $\mathcal{H}$ um espaço de funções apropriado. Detalhes a respeito da teoria da regularização de Tikhonov fogem ao escopo deste trabalho, mais informações podem ser encontradas em [22].

Para o problema discretizado (1.21), a regularização de Tikhonov substitui o problema

$$x = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \|b - Ax\|_2^2 \tag{1.28}$$

por

$$x_\lambda = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \{\|b - Ax\|_2^2 + \lambda^2 \|L(x - x_0)\|_2^2\} \tag{1.29}$$

em que λ é o parâmetro de regularização. O desafio é escolher um parâmetro λ tal que x_λ aproxime satisfatoriamente a solução exata x^{exato}. A matriz L é geralmente $L = I$ (matriz identidade), ou uma aproximação discreta do operador diferencial definido pela primeira ou segunda derivada. O vetor x_0 é uma aproximação inicial para a solução caso esteja disponível, caso contrário definimos $x_0 = 0$.

Talvez o primeiro autor a descrever um esquema que é equivalente à regularização de Tikhonov foi James Riley em 1955 [47] que propôs resolver o sistema $(A + \alpha I)x = b$ em que α é uma constante positiva pequena, Riley também sugeriu um esquema de iteração que hoje é conhecido como a regularização de Tikhonov iterada.

Em 1962, Phillips escreveu um artigo [43] voltado para problemas um pouco mais gerais. Nesse artigo a matriz A é uma matriz quadrada obtida da equação integral de Fredholm de primeira espécie por meio de uma regra de quadratura e L uma matriz

tridiagonal; a utilização desta matriz L será explicada mais adiante.

Golub foi o primeiro autor a propôr, em 1965, uma maneira apropriada de resolver o problema (1.29). A ideia é tratar (1.29) como um problema de mínimos quadrados

$$x_\lambda = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}} \left\| \begin{pmatrix} A \\ \lambda L \end{pmatrix} x - \begin{pmatrix} b \\ \lambda L x_0 \end{pmatrix} \right\|_2^2 . \tag{1.30}$$

Para entendermos melhor esta abordagem, notemos que o problema (1.21) é equivalente a resolver o sistema das equações normais

$$A^T A x = A^T b, \tag{1.31}$$

enquanto que as equações normais para (1.30) são

$$\begin{pmatrix} A \\ \lambda L \end{pmatrix}^T \begin{pmatrix} A \\ \lambda L \end{pmatrix} x_\lambda = \begin{pmatrix} A \\ \lambda L \end{pmatrix}^T \begin{pmatrix} b \\ \lambda L x_0 \end{pmatrix}, \tag{1.32}$$

ou seja,

$$(A^T A + \lambda^2 L^T L) x_\lambda = A^T b + \lambda^2 L^T L x_0. \tag{1.33}$$

A expressão (1.33) é chamada de as equações normais regularizadas. Se nessa equação considerarmos $L = I$ então o problema está na forma padrão, caso contrário o problema está na forma geral. Ambos os problemas são equivalentes no sentido de que é possível transformar um problema no outro, como veremos a seguir. Alguns problemas podem apresentar soluções mais úteis usando a forma padrão outros usando a forma geral.

A solução para a equação (1.33) pode ser escrita, para $x_0 = 0$ (o que mais ocorre), como

$$x_\lambda = (A^T A + \lambda^2 L^T L)^{-1} A^T b. \tag{1.34}$$

Para a unicidade exigimos que $\mathcal{N}(A) \cap \mathcal{N}(L) = 0$.

O ponto importante aqui é que todo problema na forma geral sempre pode ser transfor-

mado em outro equivalente na forma padrão. A idéia elementar é transformar o problema

$$x_\lambda = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\{\|b - Ax\|_2^2 + \lambda^2\|L(x - x_0)\|_2^2\} \tag{1.35}$$

em

$$\bar{x}_\lambda = \underset{\bar{x}\in\mathbb{R}^n}{\operatorname{argmin}}\{\|\bar{b} - \bar{A}\bar{x}\|_2^2 + \lambda^2\|(\bar{x} - \bar{x}_0)\|_2^2\}. \tag{1.36}$$

No caso mais simples em que L é uma matriz quadrada e não-singular, a transformação é dada por $\bar{A} = AL^{-1}$, $\bar{b} = b$, $\bar{x}_0 = Lx_0$ e a transformação reversa se torna $x_\lambda = L^{-1}\bar{x}_\lambda$.

Porém, em aplicações, o mais comum é a matriz L não quadrada. Neste caso usamos a inversa generalizada (com peso-A) de L

$$L_A^\dagger \equiv \left(I_n - \left(A\left(I_n - L^\dagger L\right)\right)^\dagger A\right) L^\dagger. \tag{1.37}$$

Para detalhes a respeito desta inversa generalizada, ver em Eldén [17].

Com a utilização de uma matriz $L \neq I$ faz-se necessária a introdução da Decomposição em Valores Singulares Generalizados (GSVD) [30]. A generalização que trabalhamos aqui está intimamente ligada com o problema de autovalor generalizado, apêndice A, o que quer dizer que, sob certas condições, os valores singulares generalizados estão relacionados com os autovalores generalizados de um par matricial (A, L).

Teorema 1.5. *(GSVD) Seja o par matricial (A, L) em que*

$$A \in \mathbb{R}^{m\times n}\ ,\ L \in \mathbb{R}^{p\times n}\ ,\ m \geqslant n \geqslant p\ ,\ posto(L) = p.$$

Então existem matrizes $U \in \mathbb{R}^{m\times n}$, $V \in \mathbb{R}^{p\times p}$ com $U^TU = I_n$, $V^TV = I_p$ e uma matriz não-singular $X \in \mathbb{R}^{n\times n}$ tais que

$$\begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix}^T \begin{bmatrix} A \\ L \end{bmatrix} X = \begin{bmatrix} \Sigma \\ M \end{bmatrix} = \begin{bmatrix} \Sigma_p & 0 \\ 0 & I_0 \\ M_p & 0 \end{bmatrix} \begin{matrix} p \\ n-p \\ p \end{matrix}$$

com $\Sigma_p = \text{diag}(\sigma_1, \ldots, \sigma_p)$ *e* $M_p = \text{diag}(\mu_1, \ldots, \mu_p)$. *Os coeficientes* σ_i *e* μ_i *satisfazem*

$$0 \leqslant \sigma_1 \leqslant \ldots \leqslant \sigma_p \leqslant 1 \quad e \quad 1 \geqslant \mu_1 \geqslant \ldots \geqslant \mu_p \geqslant 0$$

além disso, $\Sigma_p^2 + M_p^2 = I_p$. *Por fim, os valores singulares generalizados do par matricial* (A, L) *são definidos como*

$$\gamma_i = \frac{\sigma_i}{\mu_i}.$$

Com isto podemos perceber que a ordenação dos valores singulares generalizados de um par matricial (A, L) é oposta a dos valores singulares de uma matriz A. Na figura 1.2 temos os valores singulares para a matriz A para o problema *shaw* e os valores singulares generalizados do par matricial (A, L) sendo L uma discretização para a segunda derivada. Podemos perceber uma certa simetria dos valores.

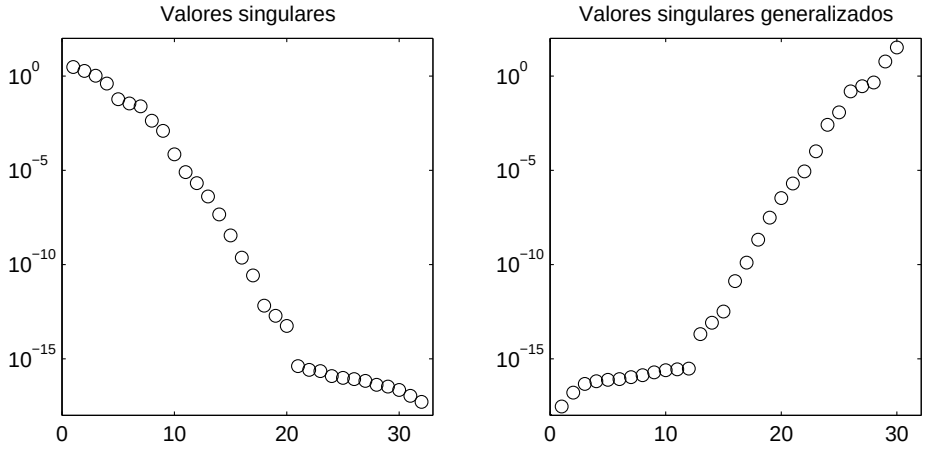

Figura 1.2: Os 32 valores singulares σ_i da matriz A e os 30 valores singulares generalizados γ_i do par matricial (A, L) para o problema *shaw* com L sendo uma aproximação para o operador segunda derivada.

Uma outra característica que a SVD e a GSVD tem em comum é o comportamento oscilatório dos vetores singulares à medida que os valores singulares (generalizados) se aproximam de zero. Conforme mostrado na figura 1.3 (problema teste *shaw*, dimensão $n = 32$ e L uma discretização para o operador segunda derivada) podemos constatar que os vetores singulares tendem a ampliar sua frequência quando os valores singulares diminuem, isto é, quando os valores singulares se aproximam de zero temos que os vetores singulares tendem a cruzar mais vezes o eixo das abscissas.

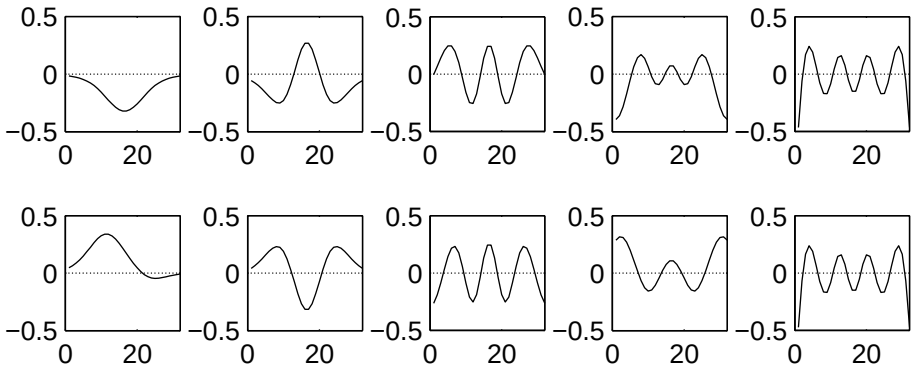

Figura 1.3: Vetores singulares u_i, i=1,3,5,7,9 (acima) e vetores singulares generalizados u_i, i=32,30,28,26,24 (abaixo).

Assim como a SVD pode ser utilizada como um método de regularização (TSVD), a GSVD também pode ser utilizada como um método de regularização, ou seja, a solução para

$$(A^T A + \lambda^2 L^T L)x_\lambda = A^T b \tag{1.38}$$

pode ser escrita como

$$x_\lambda = \sum_{i=1}^{r} \frac{\gamma_i^2}{\gamma_i^2 + \lambda^2} \frac{u_i^T b}{\gamma_i} v_i \tag{1.39}$$

e truncando o somatório em $s < r$ temos um método de regularização, TGSVD. Mais detalhes podem ser encontrados em [26].

Agora que vimos um pouco sobre a GSVD podemos voltar para a transformação da forma geral para a forma padrão. No caso em que $p \geqslant n$ então $L_A^\dagger = L^\dagger$; entretanto, em geral $L_A^\dagger$ é diferente da pseudoinversa da matriz L, $L^\dagger$, quando $p < n$. Além disso precisamos da componente $x_{\mathcal{N}(L)}$ da solução regularizada em $\mathcal{N}(L)$, dada por

$$x_{\mathcal{N}(L)} \equiv \left(A\left(I_n - L^\dagger L\right)\right)^\dagger b. \tag{1.40}$$

Dada a GSVD do par matricial (A, L), a matriz $L_A^\dagger$ e o vetor $x_{\mathcal{N}(L)}$ podem ser expressos por

$$L_A^\dagger = X \begin{pmatrix} M^{-1} \\ 0 \end{pmatrix} V^T, \qquad x_{\mathcal{N}(L)} = \sum_{i=p+1}^{n} u_i^T b x_i. \tag{1.41}$$

então a matriz $\bar{A}$, e os vetores $\bar{b}$ e $\bar{x}_0$ têm a forma

$$\bar{A} = AL_A^{\dagger}, \quad \bar{b} = b - Ax_{\mathcal{N}(L)}, \quad \bar{x}_0 = Lx_0 \tag{1.42}$$

enquanto que a transformação da solução é dada por

$$x_\lambda = L_A^{\dagger}\bar{x}_\lambda + x_{\mathcal{N}(L)}. \tag{1.43}$$

Uma das vantagens da transformação para a forma padrão é que existe uma relação entre a GSVD do par matricial (A, L) e a SVD da matriz $\bar{A}$. Se a matriz U_p consiste das primeiras p colunas de U, isto é, $U_p = (u_1, \ldots, u_p)$, então a expressão

$$AL_A^{\dagger} = U_p \Sigma M^{-1} V^T \tag{1.44}$$

mostra que os valores singulares generalizados γ_i são os valores singulares da matrix $AL_A^{\dagger}$, exceto pela ordem reversa. Mais ainda, os vetores u_i e v_i, $i = 1, \ldots, p$, são os vetores singulares a esquerda e a direita da matriz $AL_A^{\dagger}$, respectivamente.

Outra vantagem da transformação para a forma padrão é que a simples relação $Lx = \bar{x}$ (devido a $LL_A^{\dagger} = I_p$ e $Lx_{\mathcal{N}(L)} = 0$) e $Ax - b = \bar{A}\bar{x} - \bar{b}$ leva imediatamente as equações

$$\|Lx\|_2 = \|\bar{x}\|_2, \qquad \|Ax - b\|_2 = \|\bar{A}\bar{x} - \bar{b}\|_2. \tag{1.45}$$

Para problemas de grande porte a transformação para a forma padrão pode ser inviável do ponto de vista computacional uma vez que precisa da GSVD do par matricial (A, L). Como contornar esta dificuldade é assunto de pesquisas atuais.

Vamos considerar que estamos resolvendo um problema na forma padrão e que não temos nenhuma aproximação inicial para a solução, isto é, com L sendo a matriz identidade e o vetor $x_0 = 0$. Então substituindo na equação (1.33) temos

$$(A^T A + \lambda^2 I)x_\lambda = A^T b \tag{1.46}$$

e usando a SVD da matriz A segue que

$$x_\lambda = \sum_{i=1}^{r} f_i \frac{u_i^T b}{\sigma_i} v_i \quad , \quad r = \text{posto}(A) \tag{1.47}$$

em que

$$f_i = \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} \cong \begin{cases} 1 & , \quad \sigma_i \gg \lambda \\ \dfrac{\sigma_i^2}{\lambda^2} & , \quad \sigma_i \ll \lambda \end{cases} \tag{1.48}$$

são chamados fatores de filtros para a regularização de Tikhonov. O resíduo associado pode ser escrito como

$$r_\lambda = b - Ax_\lambda = \sum_{i=1}^{r} (1 - f_i) u_i^T b u_i + b_\perp \tag{1.49}$$

em que o vetor $b_\perp = b - \sum_{i=1}^{r} u_i^T b u_i$ é a componente do vetor b que não pertece ao espaço coluna da matriz A.

Os fatores de filtro "filtram" as componentes do erro da solução. Assim como na TSVD, se muita regularização for imposta (λ muito grande) teremos uma solução que pode não ter incorporado boa parte das informações do problema, porém se pouca regularização for imposta (λ muito pequeno) pouco ruído pode ser filtrado e permaneceremos com uma solução ainda inútil.

Em termos matriciais sabemos que a solução das equações normais pode ser escrita como

$$x_{LS} = A^\dagger b = V \Sigma^\dagger U^T b. \tag{1.50}$$

Com a inclusão dos fatores de filtro a solução pode ser escrita como

$$x_\lambda = A^\# b = V F \Sigma^\dagger U^T b \tag{1.51}$$

em que a matriz $A^\#$ é a pseudoinversa para o sistema $(A^T A + \lambda^2 I) x_\lambda = A^T b$ e F é a matriz dos fatores de filtro, neste caso os coeficientes de Fourier são $f_i u_i^T b$.

Para o caso da TSVD, os fatores de filtros são mais simples, $f_i = 1$ para $i = 1, \ldots, s$ e $f_i = 0$ para $i = s+1, \ldots, r$.

1.4 Condição Discreta de Picard

No início deste capítulo comentamos que a matriz A em (1.21) pode ter os valores singulares se aproximando continuamente de zero (indo para zero sem que haja um salto). A figura 1.2 é ilustrativa: os primeiros 20 valores singulares tem um decaimento quase que constante e os últimos 12 são de ordem menor do que 10^{-15}, ou seja, estes últimos atingiram a precisão da máquina.

A taxa de decaimento dos valores singulares da matriz A ou do par matricial (A, L) tem um papel importante na análise de problemas discretos.

Seja $K(x,y) = \sum_{i=1}^{\infty} \sigma_i u_i(x) v_i(y)$ a expansão em valores singulares do operador compacto K e seja $g(x) = \sum_{i=1}^{\infty} \beta_i u_i(x)$. Sabemos que para a equação integral $Kf = g$ ter uma solução de quadrado integrável é necessário e suficiente que g satisfaça a seguinte condição:

A Condição de Picard: A função g em $Kf = g$ satisfaz a Condição de Picard se

$$\sum_{i=1}^{\infty} \left(\frac{\langle u_i, g \rangle}{\sigma_i} \right)^2 < \infty \tag{1.52}$$

em que $\langle \cdot, \cdot \rangle$ denota o produto interno usual para integrais.

Os valores singulares decaem gradativamente para zero, contudo, a equação (1.52) implica que a partir de um índice i_0, os coeficientes de Fourier $\langle u_i, g \rangle$ vão mais rápido para zero do que os valores singulares tendo em vista que a série é limitada.

Para o caso finito a equação (1.52) é sempre satisfeita, no entanto, a razão entre o decaimento dos valores singulares da matriz A e o decaimento dos coeficientes de Fourier de b, $<u_i, b> = u_i^T b$, ainda representam um papel importante no sucesso da regularização de Tikhonov pois isso determina quão boa a solução regularizada x_λ aproxima a solução exata.

O resultado central é que se o módulo dos coeficientes de Fourier, $|u_i^T b|$, decaem mais rápido para zero do que os valores singulares (generalizados) então as soluções regularizadas x_λ e x_s têm aproximadamente as mesmas propriedades da solução exata x^{exato}.

Existem duas importantes exceções para esse requerimento para os coeficientes de

Fourier. A primeira exceção é que o menor valor singular generalizado do par matricial (A, L) pode ser numericamente zero, isto é, menor do que alguma tolerância ε refletindo os erros na matriz A. Nesse caso é natural considerar valores singulares generalizados γ_i menores do que $\varepsilon\|L^\dagger\|_2$ como sendo numericamente nulos devido a relação $\gamma_i = \sigma_i(1-\sigma_i^2)^{-1/2} \cong \sigma_i$ para σ_i pequeno.

A segunda exceção é que alguns coeficientes de Fourier $u_i^T b$ podem ser numericamente nulos com respeito a alguma tolerância δ refletindo os erros em b. Isso nos leva a definir a condição discreta de Picard.

A Condição Discreta de Picard (CDP): Seja b^{exato} o vetor de dados não perturbado. Então b^{exato} satisfaz a Condição Discreta de Picard se, para todo valor singular generalizado numericamente não nulo $\gamma_i > \varepsilon\|L^\dagger\|_2$ os coeficientes de Fourier correspondentes decaem em média para zero mais rápido do que γ_i.

Quando resolvemos problemas do mundo real, em que o vetor b, e as vezes a matriz A, são dominados por erros, raramente a condição discreta de Picard é satisfeita. No entanto, se a solução exata do problema satisfaz a condição discreta de Picard então é possível encontrar um λ e um s de modo que o problema regularizado satisfaça a condição discreta de Picard.

Um exemplo dessa situação é a equação integral de Fredholm de primeira espécie (que satisfaz a condição de Picard). Então, devido a forte conexão entre a expansão em valores singulares do núcleo K e a SVD da matriz A, a condição discreta de Picard também é satisfeita. Porém, devido aos erros nos dados, todos (ou alguns) os coeficientes $u_i^T b$ geralmente não satisfazem a condição discreta de Picard.

Na figura 1.4 temos uma equação de Fredholm [43] discretizada com 32 pontos e usamos dois vetores b, um sem perturbações e um com 5% de erro relativo. Podemos perceber que a condição discreta de Picard é satisfeita quando consideramos os coeficientes de Fourier para o vetor b livre de erros (gráfico da direita), notemos os coeficientes de Fourier ($\times$) abaixo dos valores singulares ($-\bullet-$). Contudo, usando o vetor b contendo erros (gráfico da esquerda) os coeficientes de Fourier ($\times$), a partir de $i = 9$ estão, de modo geral, por cima dos valores singulares ($-\bullet-$) fazendo com que a razão $|u_i^T b|/\sigma_i$ seja grande à medida

que i cresce.

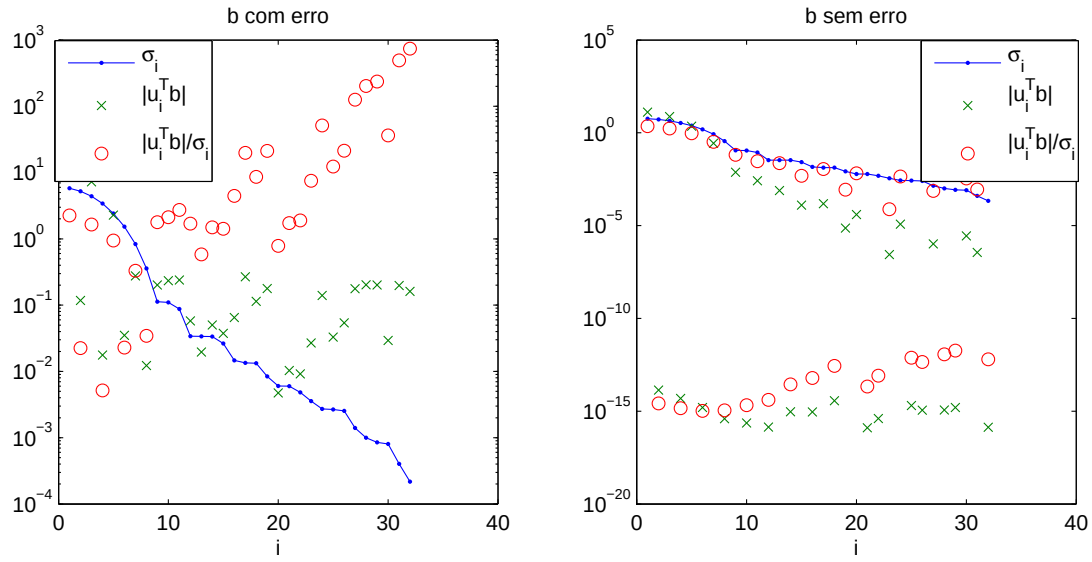

Figura 1.4: Plotagem dos valores singulares para a matriz A do problema *phillips*, coeficientes de Fourier e a razão entre eles. Lado esquerdo: vetor b contendo erro relativo de 5%; Lado direito: b livre de erros.

No entanto, com uma escolha apropriada do parâmetro de regularização de Tikhonov é possível garantir que os coeficientes de Fourier para o problema regularizado, $f_i u_i^T b$, satisfaçam a condição discreta de Picard como mostrado na figura 1.5. A escolha deste parâmetro será discutida posteriormente.

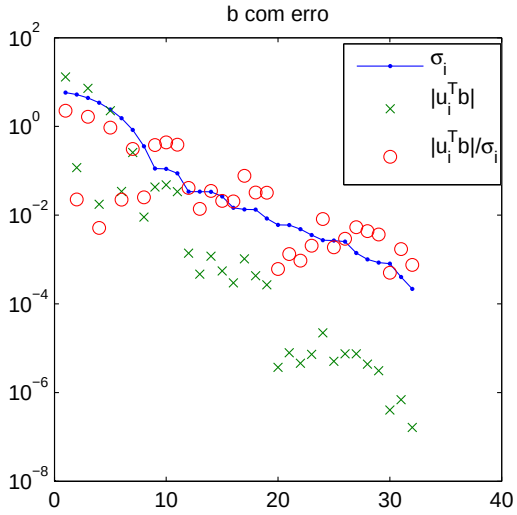

Figura 1.5: Plotagem dos valores singulares para a matriz A do problema *phillips*, coeficientes de Fourier e a razão entre eles considerando a utilização de um parâmetro de regularização λ apropriado.

Por outro lado, se o problema exato não satisfaz a condição discreta de Picard (ou mesmo a condição de Picard), então geralmente não é possível calcular uma solução boa por Tikhonov ou qualquer outro método relacionado. Um exemplo disto é o problema teste *ursell*, este problema não tem uma solução de quadrado integrável, logo não satisfaz a CDP como mostrado na figura 1.6, portanto não é possível calcular uma boa solução.

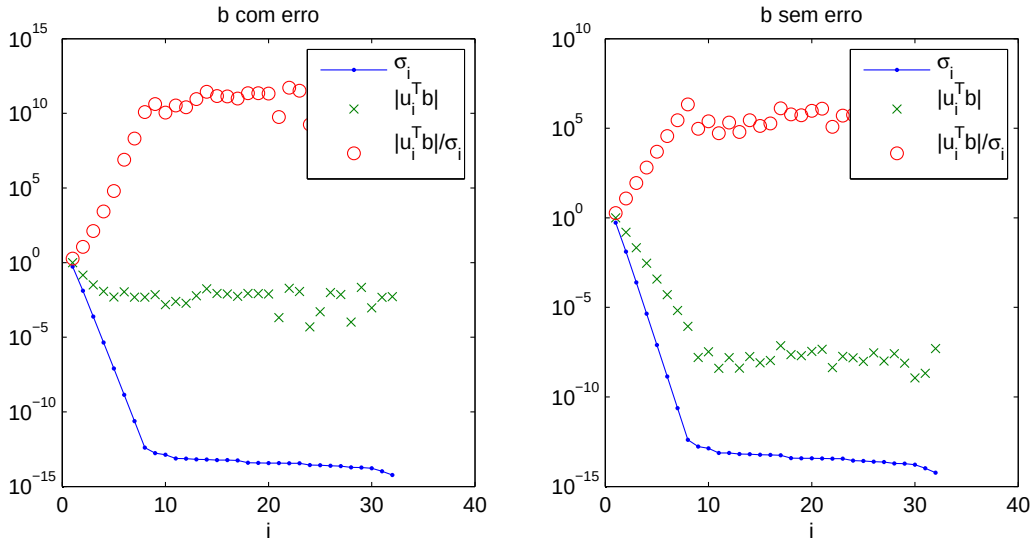

Figura 1.6: Plotagem dos valores singulares para a matriz A do problema *ursell*, coeficientes de Fourier e a razão entre eles considerando o vetor b sem erros e com erro relativo de 5%.

1.5 Métodos para determinação do parâmetro de regularização de Tikhonov

A determinação de um bom parâmetro de regularização não é uma tarefa fácil. Se pouca regularização for imposta (pequeno parâmetro de regularização) podemos não filtrar o suficiente o ruído e ainda termos uma solução inútil e se muita regularização for imposta podemos perder informações do problema.

Veremos nesta seção algumas das técnicas existentes na literatura para a determinação do parâmetro de regularização de Tikhonov.

1.5.1 GCV e W-GCV

A Validação Cruzada Generalizada, do inglês Generalized Cross-Validation (GCV), foi desenvolvida por Golub, Heath e Wahba [20], é um método muito popular para a escolha do parâmetro de regularização λ e é baseada em considerações estatísticas. A GCV sugere que um bom valor para o parâmetro de regularização deve prever dados no vetor b que estejam faltando ou que foram retirados. Mais precisamente, se um elemento (uma equação do sistema de equações $Ax = b$) arbitrário b_i do vetor 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

$$G_{A,b}(\lambda) = \frac{n\|(I - AA_\lambda^\dagger)b\|_2^2}{\Big(tr(I - AA_\lambda^\dagger)\Big)^2} \tag{1.53}$$

em que $A_\lambda^\dagger = (A^TA + \lambda^2 I)^{-1}A^T$ representa a pseudoinversa da matriz $\bar{A}$ do sistema $\bar{A}x_\lambda = \bar{b}$ em que $\bar{A} = (A^TA + \lambda^2 I)$ e $\bar{b} = A^Tb$, e a solução regularizada, x_λ, pode ser escrita como $x_\lambda = A_\lambda^\dagger b$.

Substituindo a SVD da matriz A na função GCV temos

$$G_{A,b}(\lambda) = \frac{n\left(\sum_{i=1}^{n}\left(\frac{\lambda^2 u_i^T b}{\sigma_i^2 + \lambda^2}\right)^2 + \sum_{i=n+1}^{m}\left(u_i^T b\right)^2\right)}{\left((m-n) + \sum_{i=1}^{n}\frac{\lambda^2}{\sigma_i^2 + \lambda^2}\right)^2} \tag{1.54}$$

o que torna a função GCV computacionalmente conveniente para ser avaliada e utilizada por algoritmos de minimização.

Em outras palavras, se $([b_j - [Ax]_j)^2$ for removido, o método GCV procura minimizar o erro quando x é o minimizador de

$$\sum_{i=1, i\neq j}^{m}(b_i - [Ax]_i)^2 + \lambda^2\|x\|_2^2. \tag{1.55}$$

Seja $E_j = \mathrm{diag}(1, \ldots, 1, 0, 1, \ldots, 1) \in \mathbb{R}^{m\times m}$ em que o elemento 0 aparece na j-ésima

posição. Então a minimização acima é equivalente a

$$x = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \left\{ \|E_j(b - Ax)\|_2^2 + \lambda^2 \|x\|_2^2 \right\}. \tag{1.56}$$

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 pode acontecer do parâmetro encontrado ser ineficaz produzindo uma solução não satisfatória.

Recentemente um novo método apareceu na literatura que é baseado na GCV chamado de Weighted-GCV ou simplesmente W-GCV [14].

Em vez de minimizar a função GCV, o método W-GCV busca minimizar a função

$$G_{A,b}(\omega, \lambda) = \frac{n\|(I - AA_\lambda^\dagger)b\|_2^2}{\left(tr(I - \omega AA_\lambda^\dagger)\right)^2}. \tag{1.57}$$

Seja $0 < \omega < 1$ e $F_j = \operatorname{diag}(1, \ldots, 1, \sqrt{1-\omega}, 1 \ldots, 1) \in \mathbb{R}^{m \times m}$ 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

$$x = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \left\{ \|F_j(b - Ax)\|_2^2 + \lambda^2 \|x\|_2^2 \right\}. \tag{1.58}$$

Usando a SVD da matriz A podemos escrever o denominador em (1.57) como

$$\begin{aligned} tr(I - \omega AA_\lambda^\dagger) &= \sum_{i=1}^{n} \frac{(1-\omega)\sigma_i^2 + \lambda^2}{\sigma_i^2 + \lambda^2} + (m - n) \\ &= \sum_{i=1}^{n} (1-\omega) f_i + \sum_{i=1}^{n} \frac{\lambda^2}{\sigma_i^2 + \lambda^2} + (m - n). \end{aligned} \tag{1.59}$$

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 um. O denominador se torna zero para alguns valores maiores do que um, ou seja, $\omega > 1$, nesse caso a função W-GCV tem um pólo. No artigo [14] podem ser encontradas informações a respeito de como, de

maneira adaptativa, o coeficiente ω pode ser determinado. Além disso, no mesmo artigo um algoritmo híbrido para problemas de médio/grande porte também é discutido.

Na figura 1.7 temos o gráfico da função GCV para o problema teste *heat* com dimensão $n = 32$ e erro relativo nos dados de 4%. Na esquerda temos o gráfico das funções GCV e W-GCV com $\omega = 0,8$ e na direita o gráfico das funções GCV e W-GCV com $\omega = 1,2$.

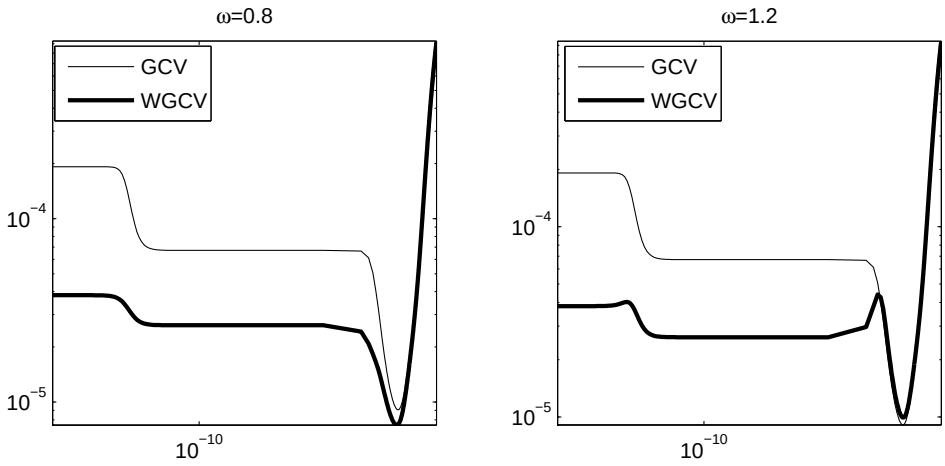

Figura 1.7: Curvas GCV e W-GCV com $\omega = \{0,8\ ,\ 1,2\}$ para o problema *heat* com 4% de erros nos dados.

Comparando as curvas da figura 1.7 podemos perceber que o valor do parâmetro ω influencia especialmente o comportamento da curva para valores pequenos de λ.

1.5.2 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 [41]. Se o problema mal-posto é consistente no sentido de que $Ax^{\text{exato}} = b^{\text{exato}}$ vale, então a idéia 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

$$\|b - Ax_\lambda\|_2 = \delta \qquad , \qquad \|e\|_2 \leqslant \delta. \tag{1.60}$$

Usando a SVD da matriz A na equação (1.49) temos que a norma do resíduo pode ser

escrita como

$$\|r_\lambda\|_2^2 = \sum_{i=1}^{r} \left((1 - f_i)u_i^T b\right)^2 + \|b_\perp\|_2^2, \tag{1.61}$$

com $b_\perp$ introduzido em (1.49), e podemos perceber que a norma do resíduo é uma função crescente e monótona com λ. Então, a equação (1.60) é o mesmo que encontrar o ponto de intersecção entre a curva da norma do resíduo e a reta δ.

O princípio da discrepância é um dos poucos métodos existentes na literatura que tem análise de erro; os critérios da GCV e da curva-L (que será visto mais adiante) que não possuem esta análise.

Além disso, o método da discrepância pode ser ótimo no sentido de minimizar $\|x^{\text{exato}} - x_{\lambda_e}\|_2$, em que λ_e denota o parâmetro determinado pelo princípio da discrepância, quando a norma do erro e é conhecida e este erro tiver certa estrutura [22]. Porém, na prática este método é quase ótimo, pois, mesmo que o erro seja conhecido, dificilmente irá satisfazer a condição ideal.

Da teoria geral da regularização de Tikhonov [22] sabemos que uma limitante superior para o erro na solução x_{λ_e} é

$$\|x^{\text{exato}} - x_{\lambda_e}\|_2 = \mathcal{O}(\|e\|_2) \tag{1.62}$$

em que $\mathcal{O}$ é utilizado para descrever a cota

$$\|x^{\text{exato}} - x_{\lambda_e}\|_2 \leqslant 2\|A^\dagger\|_2\|e\|_2 \tag{1.63}$$

A grande dificuldade deste método para escolha do parâmetro de regularização λ 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.

Na figura 1.8 temos três reconstruções para a solução do problema *foxgood* (problema teste localizado na toolbox RegularizationTools [31]) com dimensão 32 e erro relativo nos dados de 3%. A curva tracejada representa a solução exata enquanto que a outra curva

representa a solução obtida. No gráfico de baixo a reconstrução foi feita usando o valor correto para $\|e\|_2$, no gráfico da esquerda e da direita foram utilizados, respectivamente, $\delta = 0,9\|e\|_2$ e $\delta = 1,1\|e\|_2$. No caso em que usamos $\delta = 0,9\|e\|_2$ podemos perceber que a solução obtida é inútil no sentido de não se parecer absolutamente em nada com a solução correta, contudo, usando $\delta = 1,1\|e\|_2$ a solução obtida se torna mais interessante. Para os dois casos em que não foi utilizado $\delta = \|e\|_2$ os erros relativos na solução são de 1.852,30% e 7,09%, e para $\delta = \|e\|_2$ obtemos um erro relativo na solução de 1,21%.

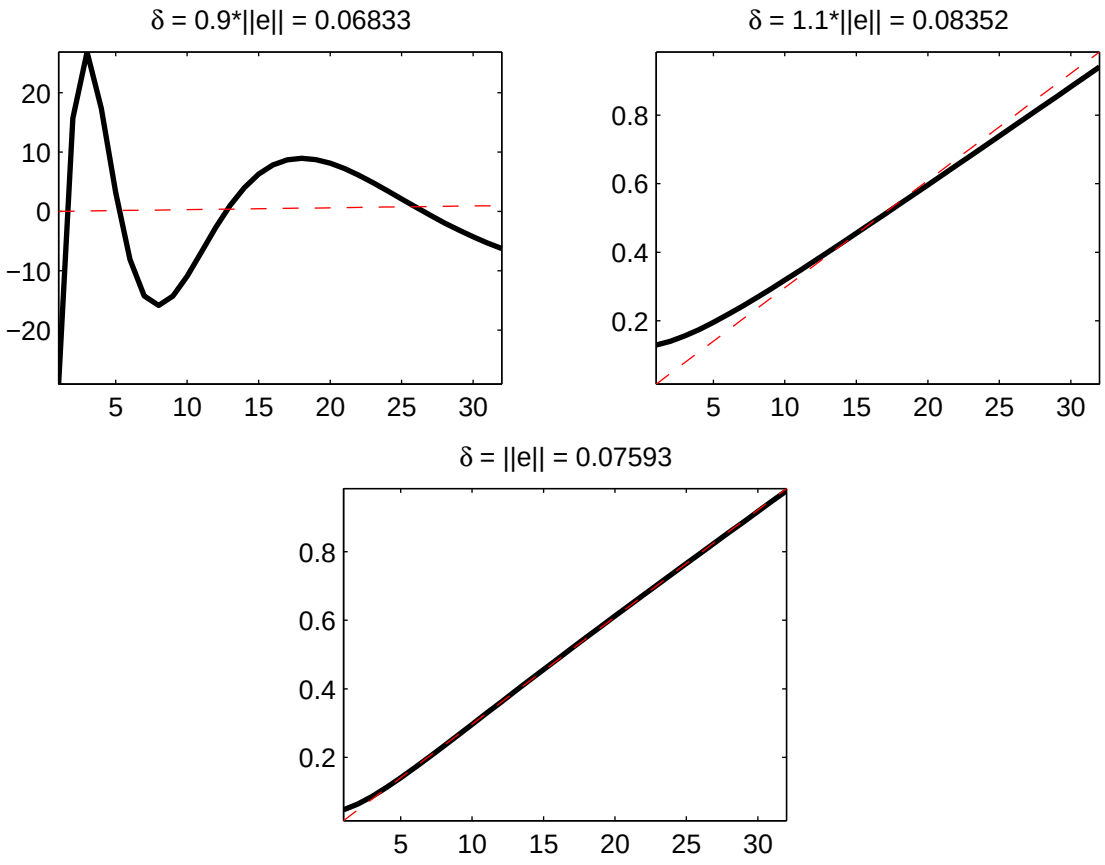

Figura 1.8: Três soluções encontradas pelo princípio da discrepância para diferentes estimativas para a norma do erro e, $\|e\|_2$.

1.5.3 Curva-L

Usando a SVD da matriz A vimos que a solução e o resíduo para a regularização de Tikhonov são dadas pelas equações (1.47) e (1.49). Com essas expressões as respectivas

normas são obtidas por

$$\|x_\lambda\|_2^2 = \sum_{i=1}^{r} \left(f_i \frac{u_i^T b}{\sigma_i} \right)^2, \tag{1.64}$$

$$\|r_\lambda\|_2^2 = \sum_{i=1}^{r} \left((1 - f_i) u_i^T b \right)^2 + \|b_\perp\|_2^2. \tag{1.65}$$

Essas expressões formam a base para a análise do método da curva-L. A curva-L é uma curva parametrizada por $\lambda \in [0, \infty[$ dada por

$$\mathcal{L}(\lambda) = \{(a, b)/a = \log(\|r_\lambda\|_2^2), b = \log(\|x_\lambda\|_2^2)\}. \tag{1.66}$$

O método apareceu em 1992 e foi definido por Hansen [28]. O método é muito popular e tem sido utilizado com sucesso em diversos problemas.

Consideremos primeiro a curva-L correspondente aos dados exatos b^{exato} e que a condição discreta de Picard seja satisfeita (essa condição garante que existe uma solução fisicamente útil para o problema inverso em questão). Uma consequência imediata é que os coeficientes de Fourier da solução exata, $|v_i^T x^{\text{exato}}| = |\frac{u_i^T b^{\text{exato}}}{\sigma_i}|$, em que $x^{\text{exato}} = A^\dagger b^{\text{exato}}$, também satisfazem a condição discreta de Picard.

Vamos assumir que $\lambda \in [\sigma_r, \sigma_1]$, o que é razoável à luz de (1.48), e que existam aproximadamente k fatores de filtros satisfazendo $f_i \cong 1$. Segue de (1.64) que

$$\|x_\lambda^{\text{exato}}\|_2^2 \cong \sum_{i=1}^{k} (v_i^T x^{\text{exato}})^2 \cong \sum_{i=1}^{r} (v_i^T x^{\text{exato}})^2 = \|x^{\text{exato}}\|_2^2 \tag{1.67}$$

pois estamos usando o fato de que os coeficientes de Fourier $|v_i^T x^{\text{exato}}|$ decaem de tal modo que os últimos $r - k$ termos contribuem muito pouco com o somatório. A expressão (1.67) vale enquanto λ não é muito grande.

Podemos perceber que quando $\lambda \to \infty$ (e $k \to 0$) temos $x_\lambda^{\text{exato}} \to 0$, por outro lado, quando $\lambda \to 0$ temos $x_\lambda^{\text{exato}} \to x^{\text{exato}}$ (x^{exato} corresponde à solução para o problema cujos dados considerados são exatos, sem ruídos).

O resíduo correspondente a x_λ^{exato} satisfaz

$$\|b - Ax_\lambda^{\text{exato}}\|_2^2 \cong \sum_{i=k+1}^{r} (u_i^T b^{\text{exato}})^2 \tag{1.68}$$

mostrando que a norma do resíduo é uma função crescente de $\|b_\perp\|_2$ a $\|b\|_2$. Logo a curva-L correspondente a b^{exato} é uma curva plana em $\|x_\lambda^{\text{exato}}\|_2 \cong \|x^{\text{exato}}\|_2$ exceto para grandes valores de $\|b - Ax_\lambda^{\text{exato}}\|_2$ em que a curva se aproxima do eixo das abscissas.

Vamos considerar agora a curva-L correspondente ao erro e e também assumir que este erro seja branco, isto é, a matriz de covariância para o vetor e é um múltiplo da matriz identidade. Essa suposição implica que os coeficientes $u_i^T e$ sejam independentes de i, o que é razoável o erro ser "igual" em todas as direções, então

$$|u_i^T e| \cong \varepsilon \quad , \quad i = 1, \ldots, m \tag{1.69}$$

o que implica que a condição discreta de Picard não é satisfeita $\forall i$.

Seja x_λ^e a solução para o problema constituindo apenas de erro, então

$$\begin{aligned} \|x_\lambda^e\|_2^2 \cong \sum_{i=1}^{r} \left(\frac{\sigma_i \varepsilon}{\sigma_i^2 + \lambda^2} \right)^2 &\cong \sum_{i=1}^{k} \left(\frac{\varepsilon}{\sigma_i} \right)^2 + \sum_{i=k+1}^{r} \left(\frac{\sigma_i \varepsilon}{\lambda^2} \right)^2 \\ &= \varepsilon^2 \left(\sum_{i=1}^{k} \sigma_i^{-2} + \lambda^{-4} \sum_{i=k+1}^{r} \sigma_i^2 \right) \end{aligned} \tag{1.70}$$

O primeiro somatório é dominado por $\sigma_k^{-2} \cong \lambda^{-2}$ enquanto que o segundo somatório é dominado por $\sigma_{k+1}^2 \cong \lambda^2$ e então obtemos a expressão aproximada

$$\|x_\lambda^e\|_2 \cong c_\lambda \frac{\varepsilon}{\lambda} \tag{1.71}$$

em que c_λ varia lentamente com λ. Logo vemos que $\|x_\lambda^e\|_2$ é uma função crescente e monótona que varia entre 0 (para λ decrescente) até atingir $\|A^\dagger e\|_2 \cong \varepsilon \|A^\dagger\|_F$ para $\lambda = 0$.

A norma do resíduo associado satisfaz

$$\|b - Ax_\lambda^e\|_2^2 \cong \sum_{i=k}^{m} \varepsilon^2 = (m - k)\varepsilon^2. \tag{1.72}$$

Portanto, $\|b - Ax_\lambda^e\|_2 \cong \varepsilon\sqrt{m-k}$ é uma função que varia lentamente com λ de 0 até $\|e\|_2 \cong \varepsilon\sqrt{m}$ e a curva-L é uma curva vertical em $\|b - Ax_\lambda^e\|_2 \cong \|e\|_2$ exceto para valores de λ muito pequenos em que a curva se aproxima do eixo das ordenadas.

Finalmente, considerando a curva-L para $b = b^{\text{exato}} + e$, dependendo do λ, são os coeficientes $u_i^T b^{\text{exato}}$ ou os coeficientes $u_i^T e$ que dominam e a curva-L resultante essencialmente consiste de uma parte da curva-L para b^{exato} e outra parte para 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 1.9

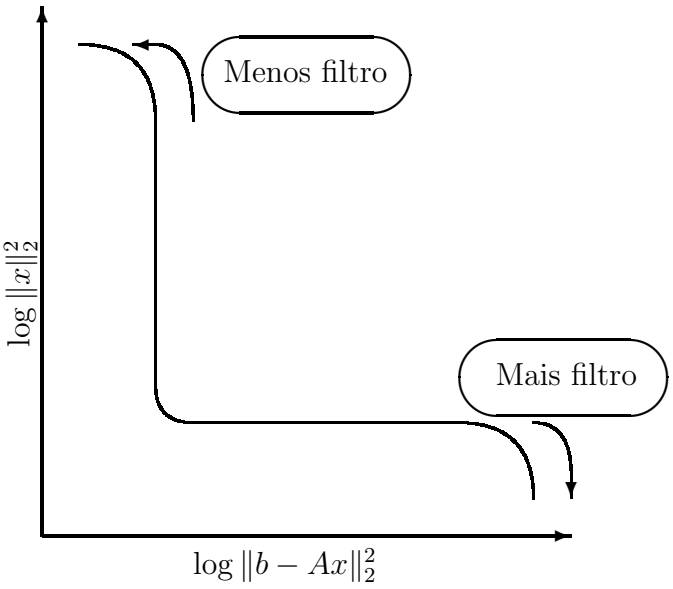

Figura 1.9: Curva-L genérica.

Baseado nessas considerações, Hansen [28] propôs escolher o parâmetro de regularização λ localizado na região de transição entre as duas curvas. Mais precisamente o parâmetro λ deve ser tal que a curvatura em $(\log \|r_\lambda\|_2^2, \log \|x_\lambda\|_2^2)$ seja maximizada.

Então, denotando por $\rho = \|b - Ax_\lambda\|_2$ e $\eta = \|x_\lambda\|_2$, a curvatura num ponto da curva-L é dada por

$$\kappa = \rho\eta \frac{\rho\eta/|\eta'| - \lambda^2 - \lambda^4\eta}{(\rho^2 + \lambda^4\eta^2)^{3/2}} \tag{1.73}$$

em que $|\eta'|$ é uma quantidade dada por

$$|\eta'| = \sum_{i=1}^{r} \frac{2\sigma_i}{(\sigma_i^2 + \lambda^2)^3}(u_i^T b)^2. \tag{1.74}$$

A proposta da curva-L é um procedimento heurístico, pois não há nenhuma argumentação para que o parâmetro de regularização obtido produza uma boa solução.

O critério curva-L como método para a determinação do parâmetro de regularização de Tikhonov apresenta duas limitações. Uma diz respeito à reconstrução de soluções exatas muito suaves, isto é, soluções x^{exato} cujos coeficientes da SVD $|v_i^T x^{\text{exato}}|$ decaem rápido para zero de modo que a solução x^{exato} seja dominada pelas primeiras componentes da SVD. Para situações como estas o parâmetro encontrado pelo critério da curva-L pode ser consideravelmente menor do que o parâmetro ótimo (entendemos por parâmetro ótimo aquele que minimiza o erro relativo $\|x_\lambda - x^{\text{exato}}\|_2/\|x^{\text{exato}}\|_2$). Observa-se que quanto mais suave for a solução, pior será o parâmetro. Detalhes a respeito desta limitação são apontados por Hanke [24].

A outra limitação está relacionada com o comportamento assintótico quando a dimensão n do sistema discretizado aumenta, ou seja, quando tentamos obter uma solução mais precisa da equação integral, usualmente refinamos o intervalo aumentando assim o número de pontos, isto é, aumentamos o número de variáveis. Neste caso o parâmetro de regularização calculado pelo critério da curva-L não se comporta consistentemente com o parâmetro ótimo quando n aumenta. Esta limitação foi apontada por Vogel [50]. Outras informações podem ser encontradas em [29].

Uma das questões com relação à curva-L é conhecer em que regiões ela é convexa e em que regiões ela é côncava. Hansen [29] e Regińska [46] deram os primeiros passos na tentativa de elucidar esta questão, porém, apenas algumas considerações foram desenvolvidas e uma resposta definitiva parece não existir.

Regińska provou que se $\|b_\perp\|_2 = 0$ então a curva-L é côncava para $\lambda \in]0, \sigma_n[\cup]\sigma_1, \infty[$, e se $\|b_\perp\|_2 \neq 0$ então a curva-L é côncava para $\lambda > \sigma_1$ e convexa para $\lambda \in]0, \epsilon[$, em que $\epsilon \cong 0$ e a análise da Regińska nada diz a respeito do comportamento da curva-L no intervalo $[\sigma_n, \sigma_1]$. A pergunta agora é respondida completamente por Bazán e Francisco [3].

Por fim, na figura 1.10 temos dois exemplos para a curva-L. A curva-L da esquerda corresponde ao problema *foxgood* e a curva-L da direita corresponde ao problema *heat*. Ambos foram criados com 2% de erros relativos nos dados e com dimensão $n = 32$.

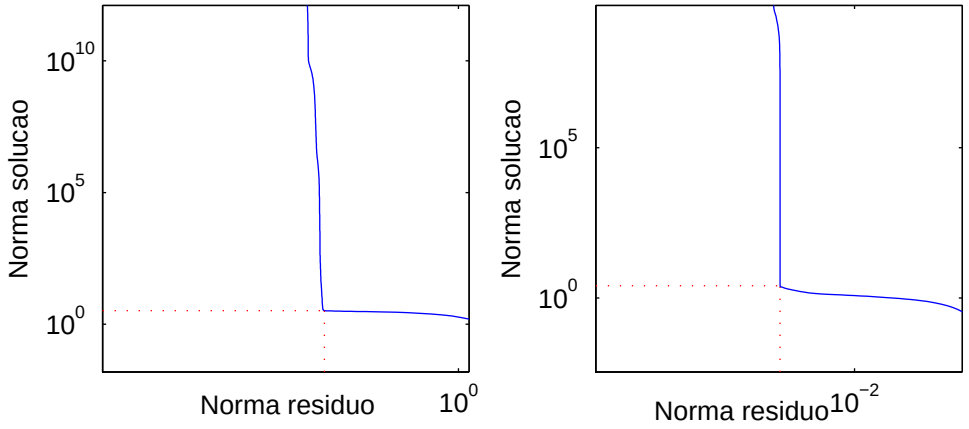

Figura 1.10: Curva-L para os problemas *foxgood* e *heat* com 2% de erros relativos nos dados.

1.5.4 Ponto-Fixo

Recentemente Bazán [2] propôs um algortimo de ponto-fixo para determinar o parâmetro de regularização baseado no trabalho de Regińska [46].

Vimos anteriormente que usando a SVD da matriz A, temos que a norma da solução e a norma do resíduo associado são dados pelas equações (1.64) e (1.65). Vamos definir duas funções: $\mathbf{y}(\lambda)$ para representar a norma da solução e $\mathbf{x}(\lambda)$ para representar a norma do resíduo, por

$$\mathbf{y}(\lambda) \triangleq \sum_{i=1}^{r} \left(f_i \frac{u_i^T b}{\sigma_i} \right)^2 \quad \text{e} \quad \mathbf{x}(\lambda) \triangleq \sum_{i=1}^{r} \left((1 - f_i) u_i^T b \right)^2 + \|b_\perp\|_2^2. \tag{1.75}$$

O símbolo $\triangleq$ indica "por definição". Se denotarmos por $\varpi_i = |u_i^T b|^2$ então podemos escrever as duas equações acima por

$$\mathbf{x}(\lambda) = \sum_{i=1}^{r} \frac{\lambda^4 \varpi_i}{\left(\sigma_i^2 + \lambda^2\right)^2} + \|b_\perp\|_2^2, \tag{1.76}$$

$$\mathbf{y}(\lambda) = \sum_{i=1}^{r} \frac{\sigma_i^2 \varpi_i}{\left(\sigma_i^2 + \lambda^2\right)^2}. \tag{1.77}$$

Derivando com respeito a λ obtemos

$$\mathbf{x}(\lambda)' = 4\lambda^3 \sum_{i=1}^{r} \frac{\sigma_i^2 \varpi_i}{\left(\sigma_i^2 + \lambda^2\right)^3} > 0, \qquad \mathbf{y}(\lambda)' = -4\lambda \sum_{i=1}^{r} \frac{\sigma_i^2 \varpi_i}{\left(\sigma_i^2 + \lambda^2\right)^3} < 0 \tag{1.78}$$

e utilizando as expressões acima encontramos

$$d\mathbf{y}/d\mathbf{x} = -1/\lambda^2 \tag{1.79}$$

o que mostra que $\mathbf{y}(\lambda)$ é uma função de $\mathbf{x}$ monótona e decrescente. Tanto a função $\mathbf{y}(\lambda)$ quanto a função $\mathbf{x}(\lambda)$ são monótonas, uma decrescente e a outra crescente, por este motivo possuem inversa e podemos escrever, por exemplo $\lambda(\mathbf{x})$, e substituindo em $\mathbf{y}(\lambda)$ obtemos que $\mathbf{y}$ é uma função de $\mathbf{x}$.

Regińska, em 1996, 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 $(\log \mathbf{x}(\lambda^*), \log \mathbf{y}(\lambda^*))$ tem inclinação $-1/\mu$, então $\lambda = \lambda^*$ deve ser um minimizador de

$$\Psi_\mu(\lambda) = \mathbf{x}(\lambda)\mathbf{y}(\lambda)^\mu, \quad \mu > 0. \tag{1.80}$$

Derivando (1.80) com respeito a λ temos

$$\Psi'_\mu(\lambda) = \mathbf{y}(\lambda)^\mu \mathbf{y}'(\lambda) \left[\mu \frac{\mathbf{x}(\lambda)}{\mathbf{y}(\lambda)} + \frac{\mathbf{x}'(\lambda)}{\mathbf{y}'(\lambda)}\right] \tag{1.81}$$

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

$$\lambda^{*2} = \mu \frac{\mathbf{x}(\lambda^*)}{\mathbf{y}(\lambda^*)} \Leftrightarrow \lambda^* = \sqrt{\mu} \frac{\sqrt{\mathbf{x}(\lambda^*)}}{\sqrt{\mathbf{y}(\lambda^*)}}. \tag{1.82}$$

Bazán concluiu que se λ^* for um minimizador de $\Psi_\mu(\lambda)$ então λ^* deve ser um ponto fixo de $\phi_\mu : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ definida por

$$\phi_\mu(\lambda) = \sqrt{\mu} \frac{\sqrt{\mathbf{x}(\lambda)}}{\sqrt{\mathbf{y}(\lambda)}}. \tag{1.83}$$

O algoritmo de ponto fixo começa com um chute inicial λ_0, $\mu = 1$ e prossegue com a sequência, para $k = 0, 1, 2, \ldots$

$$\lambda_{k+1} = \phi_\mu(\lambda_k). \tag{1.84}$$

Na figura (1.11) podemos observar a reta $z = \lambda$ e o comportamento da função $\phi_1(\lambda)$ para

o problema teste *shaw* com dimensão $n = 256$ e 1% de ruído nos dados. Para critérios de parada ou ajustes no μ o leitor é recomendado a ler [2].

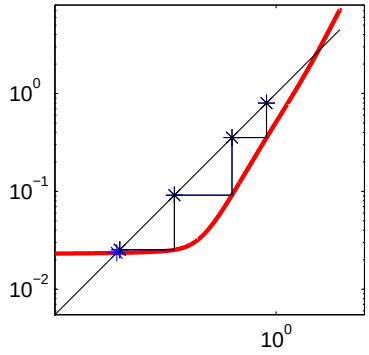

Figura 1.11: Comportamento das iterações da função $\phi_1(\lambda)$ para o problema teste *shaw*.

No próximo capítulo veremos detalhes teóricos a respeito deste método para a determinação do parâmetro de regularização de Tikhonov.

Tanto a TSVD como a regularização de Tikhonov são considerados métodos diretos, ou seja, dado o parâmetro de regularização a solução é obtida. Diferentemente, os métodos iterativos vão construindo a solução à medida que as iteração vão sendo executadas. Na literatura existe uma variedade de métodos iterativos para calcular soluções generalizadas, regularizadas, máxima entropia e de mínimos quadrados para o sistema $Ax = b$. Veremos rapidamente três métodos iterativos.

1.5.5 Regularização iterativa

Para problemas de grande porte em que o cálculo dos valores singulares da matriz A não pode ser realizado em virtude do alto custo computacional, usualmente utilizamos algum método iterativo. A ideia por trás dos métodos iterativos está em projetar o problema original (1.21) em algum subespaço, em particular em subespaços de Krylov. Uma caracterísa comum a muitos métodos iterativos é determinar a iteração de parada uma vez que após atingir o número ótimo de passos, as soluções calculadas tendem a incorporar mais informações do ruído do que gostaríamos, do mesmo modo como visto com a TSVD.

Dentre vários métodos podemos citar as iterações de Landweber [16], os Gradientes Conjugados [30] e a técnica de Reconstrução Algébrica [16] que veremos a seguir. Outras

técnicas podem ser encontradas em [8, 30].

Iteração de Landweber

Aqui, a sequência $x^{(k)}$, $k = 0, 1, \ldots$, é calculada por

$$x^{(k+1)} = x^{(k)} + \gamma^2 A^T (b - Ax^{(k)}) \tag{1.85}$$

em que $\gamma > 0$ é um parâmetro real e usualmente $x^{(0)} = 0$.

Usando a SVD da matriz A podemos escrever

$$x^{(k)} = \sum_{i=1}^{p} c_i^{(k)} v_i \tag{1.86}$$

em que $c_i^{(k)}$ satisfaz a recursão

$$c_i^{(k+1)} = \left(1 - \gamma^2 \sigma_i^2\right) c_i^{(k)} + \gamma^2 \sigma_i < b, u_i > . \tag{1.87}$$

De $c_i^{(0)} = 0$ temos

$$c_i^{(k)} = F_k(\sigma_i)\sigma_i^{-1} < b, u_i > \tag{1.88}$$

$$F_k(\sigma) = 1 - \left(1 - \gamma^2 \sigma^2\right)^{(k)} \tag{1.89}$$

então,

$$x^{(k)} = \sum_{i=1}^{p} F_k(\sigma_i)\sigma_i^{-1} < b, u_i > v_i. \tag{1.90}$$

As iterações convergem para $A^{\dagger}b$ se $\gamma\sigma_1 < 1$ e a taxa de convergência da contribuição de v_i depende do tamanho de σ_i. Para σ_i grande, a convergência é rápida, mas para pequenos σ_i a convergência é lenta. Então nas primeiras iterações as contribuições dos maiores valores singulares são bem representadas, enquanto que a contribuição dos menores valores singulares aparecem apenas após muitas iterações.

Isto mostra que parando as iterações após um número finito de passos tem o mesmo efeito da regularização, ou seja, muitos passos destroem a precisão das soluções e poucos passos podem não ter incorporado o suficiente para uma boa solução.

Método dos Gradientes Conjugados

Como vimos, as iterações de Landweber calculam primeiro as partes que estão associadas aos maiores valores singulares. O algoritmo CG (Conjugate Gradients) tem uma propriedade similar porém mais favorável.

O método CG minimiza $\|b - Ax\|_2$ da seguinte maneira. Dado $x^{(0)}$, chute inicial, definimos $d^{(0)} = b - Qx^{(0)}$ em que $Q = A^T A$, então $x^{(k)}, d^{(k)}$ são calculados recursivamentes por

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} d^{(k)} \ , \ \alpha^{(k)} = -\frac{< g^{(k)}, d^{(k)} >}{< d^{(k)}, Qd^{(k)} >} \ , \ g^{(k)} = -A^T b + Qx^{(k)} \tag{1.91}$$

$$d^{(k+1)} = -g^{(k+1)} + \beta^{(k)} d^{(k)} \ , \ \beta^{(k)} = \frac{< g^{(k+1)}, Qd^{(k)} >}{< d^{(k)}, Qd^{(k)} >}. \tag{1.92}$$

Notemos que o método CG forma explicitamente a matriz $A^T A$, logo não devemos usar este método se existirem valores singulares do tamanho $10^{-t/2}$ em que t é o número de digitos usados no computador.

Técnica de Reconstrução Algébrica

No método ART (Algebraic Reconstruction Technique) escrevemos $Ax = b$ como

$$< a_j, x >= b_j \quad , \quad j = 1, \ldots, m \quad , \quad \|a_j\| = 1 \tag{1.93}$$

em que os vetores $a_j \in \mathbb{R}^n$ são as linhas da matriz A. Vamos definir P_j como sendo o j-ésimo projetor ortogonal nestes hiperplanos, isto é,

$$P_j x = x + (b_j - < a_j, x >) a_j \tag{1.94}$$

e considerar um parâmetro de relaxação $0 < w < 2$

$$P_j^w = (1 - w)I + wP_j \quad , \quad P^w = P_m^w \cdots P_1^w. \tag{1.95}$$

Então, as iterações ART são dadas por $x^{(k+1)} = P^w x^{(k)}$. Para $w = 1$ obtemos um método conhecido como o método de Kaczmarz que simplesmente projeta ortogonalmente

nos hiperplanos definindo $Ax = b$.

A análise de convergência pode ser feita da mesma maneira da utilizada na análise numérica do método SOR, pois o ART é essencialmente o método SOR aplicado ao sistema $A^T Ax = A^T b$.

Capítulo 2

Algoritmo de Ponto-Fixo

Sabendo que o método de Tikhonov necessita de um bom parâmetro de regularização para ser bem sucedido, vimos no capítulo anterior que existem várias técnicas que tentam realizar esta tarefa. Neste capítulo o algoritmo de ponto-fixo será tratado e mais detalhes uma vez que é um método recente.

Regińska [46] provou que o minimizador do funcional

$$\Psi_\mu(\lambda) = \mathbf{x}(\lambda)\mathbf{y}(\lambda)^\mu, \quad \mu > 0, \tag{2.1}$$

está relacionado com o ponto de máxima curvatura da curva-L e Bazán [2] mostrou que o minimizador deve ser um ponto-fixo da função $\phi_\mu(\lambda)$ dada por

$$\phi_\mu(\lambda) = \sqrt{\mu}\frac{\sqrt{\mathbf{x}(\lambda)}}{\sqrt{\mathbf{y}(\lambda)}} \quad , \quad \mu > 0. \tag{2.2}$$

Antes de procedermos com detalhes teóricos a respeito deste algoritmo de ponto-fixo, veremos apenas como se comportam a curva-L e o ponto-fixo para determinar o parâmetro de regularização. Na figura 2.1 e na tabela 2.1 temos uma comparação entre o algoritmo de ponto-fixo e a curva-L para os problemas *shaw* e *heat*, ambos com dimensão $n = 32$ mas com erros diferentes, *shaw* com 1% de erro nos dados e *heat* com 5%. Na parte superior temos as curvas para *shaw*, na esquerda a função $\phi_1(\lambda)$ com os parâmetros de regularização encontrados pelas duas técnicas (Ponto-Fixo e Curva-L) e na direita temos a curva-L com os parâmetros encontrados. Neste exemplo os parâmetros praticamente

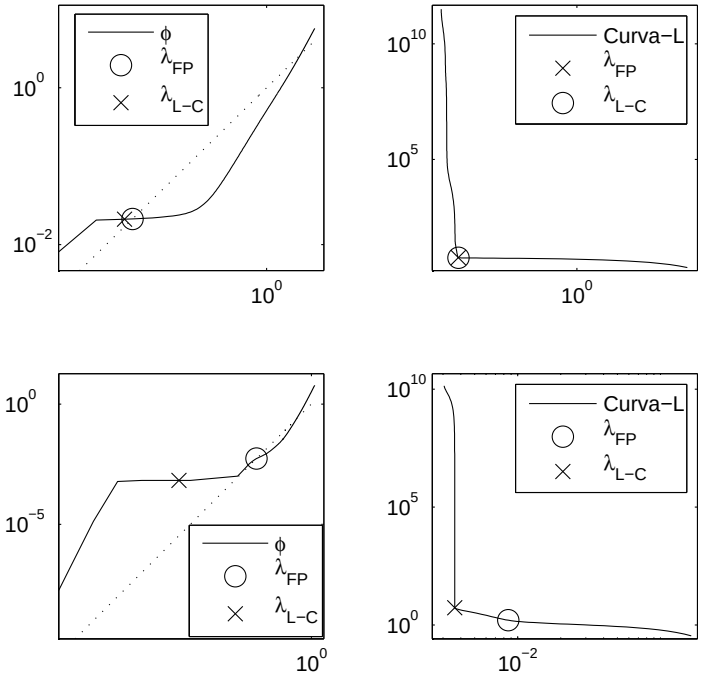

Figura 2.1: Esquerda: função $\phi_1(\lambda)$ para os problemas *shaw* (superior) e *heat* (inferior). Direita: curva-L para *shaw* (superior) e *heat* (inferior).

	shaw		heat	
	FP	C-L	FP	C-L
$\|x\|_2$	5.7037	5.7298	1.5901	5.3872
E	0.0702	0.0791	0.5294	3.7385
λ	0.0212	0.0169	0.0054	3.4105e-06

Tabela 2.1: Curva-L e FP

coincidem (ver tabela 2.1), o que implica que o erro relativo na solução e a norma da solução não difere muito. Na parte inferior temos as curvas para o problema *heat* e neste caso as técnicas obtiveram resultados diferentes e o algoritmo de ponto-fixo obteve resultados mais farováveis.

A existência de ponto-fixo da função $\phi_\mu(\lambda)$ depende do parâmetro μ e na grande maioria dos problemas $\mu = 1$ é o suficiente para que exista ponto-fixo. Um exemplo de problema em que isso não ocorre é o problema *helio*. Esse problema envolve um sistema cuja matriz A tem dimensão 212×100 e a matrix L aproxima o operador segunda derivada.

Na figura 2.2 podemos verificar que quando usamos $\mu = 1$ a função $\phi_1(\lambda)$ está total-

mente por cima da reta $z = \lambda$, isso significa que não temos ponto-fixo e μ precisa ser reajustado. Quando escolhemos, por exemplo, $\mu = 0,3$ a curva passa a ter pontos-fixos e neste caso o algoritmo encontraria o λ marcado com um $\times$. Para detalhes da escolha do parâmetro μ ver [2].

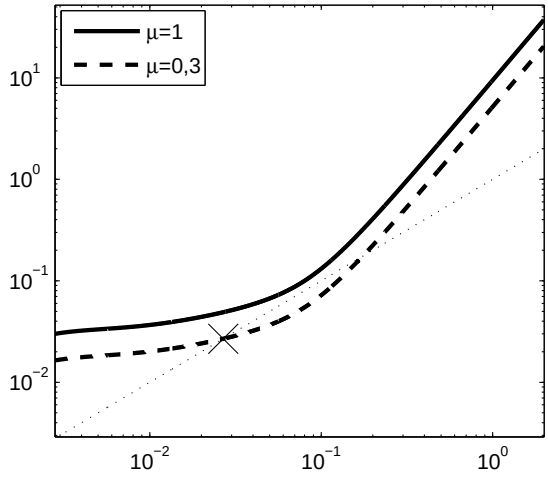

Figura 2.2: Funções $\phi_1(\lambda)$ e $\phi_{0,3}(\lambda)$ para o problema *helio* com 5% de erro nos dados.

Com intuito de responder a questão sobre existência de ponto-fixo veremos agora algumas propriedades da função $\phi_\mu(\lambda)$ definidia pela equação (2.2).

Lema 2.1. *Assuma que* $\mu = 1$. *Seja* $\underline{\sigma} = \min_i \sigma_i$, $\overline{\sigma} = \max_i \sigma_i$. *Se* $\|b_\perp\|_2 = 0$ *e* $0 \leqslant \lambda \leqslant \underline{\sigma}$, *então*

$$0 \leqslant \phi_1(\lambda) \leqslant \lambda. \tag{2.3}$$

Além disso, se $\lambda \geqslant \overline{\sigma}$, *então independentemente de* $\|b_\perp\|_2$ *vale*

$$\phi_1(\lambda) \geqslant \lambda. \tag{2.4}$$

Demonstração. Como $\lambda \leqslant \underline{\sigma}$ temos que $\lambda^4 \varpi_i \leqslant \underline{\sigma}^4 \varpi_i$. Dividir ambos os lados por $(\sigma_i^2 + \lambda^2)^2$ não muda a desigualdade e tomando o somatório de $i = 1$ até r temos

$$\mathbf{x}(\lambda) = \sum_{i=1}^{r} \frac{\lambda^4 \varpi_i}{(\sigma_i^2 + \lambda^2)^2} \leqslant \underline{\sigma}^4 \sum_{i=1}^{r} \frac{\varpi_i}{(\sigma_i^2 + \lambda^2)^2}. \tag{2.5}$$

Analogamente, como $\sigma_i^2\varpi_i \geqslant \underline{\sigma}^2\varpi_i$ e tomando o somatório de $i = 1$ até r, temos

$$\mathbf{y}(\lambda) = \sum_{i=1}^{r} \frac{\sigma_i^2\varpi_i}{\left(\sigma_i^2 + \lambda^2\right)^2} \geqslant \sum_{i=1}^{r} \frac{\underline{\sigma}^2\varpi_i}{\left(\sigma_i^2 + \lambda^2\right)^2}. \tag{2.6}$$

Logo,

$$\phi_1(\lambda)^2 = \frac{\mathbf{x}(\lambda)}{\mathbf{y}(\lambda)} \leqslant \frac{\underline{\sigma}^4 \sum_{i=1}^{r} \frac{\varpi_i}{\left(\sigma_i^2 + \lambda^2\right)^2}}{\underline{\sigma}^2 \sum_{i=1}^{r} \frac{\varpi_i}{\left(\sigma_i^2 + \lambda^2\right)^2}} = \underline{\sigma}^2 \Rightarrow \phi_1(\lambda) \leqslant \underline{\sigma}. \tag{2.7}$$

Agora para $\hat{\lambda} > 0$ com $0 \leqslant \lambda \leqslant \hat{\lambda} \leqslant \underline{\sigma}$ temos

$$\mathbf{x}(\lambda) \leqslant \hat{\lambda}^4 \sum_{i=1}^{r} \frac{\varpi_i}{\left(\sigma_i^2 + \lambda^2\right)^2}. \tag{2.8}$$

Com esta desigualdade e com (2.6) temos $\phi_1(\lambda)^2 \leqslant \hat{\lambda}^4/\underline{\sigma}^2 \leqslant \hat{\lambda}^2$ o que prova a primeira desigualdade. Para provar a segunda parte vamos supor que não vale $\phi_1(\lambda) \geqslant \lambda, \forall\lambda \geqslant \overline{\sigma}$, então existe um $\lambda_0 \geqslant \overline{\sigma}$ tal que $\phi_1(\lambda_0) < \lambda_0$. Então

$$\lambda_0^2 > \frac{\lambda_0^4 \sum_{i=1}^{r} \frac{\varpi_i}{\left(\sigma_i^2 + \lambda_0^2\right)^2} + \|b_\perp\|_2^2}{\sum_{i=1}^{r} \frac{\sigma_i^2\varpi_i}{\left(\sigma_i^2 + \lambda_0^2\right)^2}} \geqslant \frac{\lambda_0^4 \sum_{i=1}^{r} \frac{\varpi_i}{\left(\sigma_i^2 + \lambda_0^2\right)^2} + \|b_\perp\|_2^2}{\overline{\sigma}^2 \sum_{i=1}^{r} \frac{\varpi_i}{\left(\sigma_i^2 + \lambda_0^2\right)^2}} \geqslant \frac{\lambda_0^4 \sum_{i=1}^{r} \frac{\varpi_i}{\left(\sigma_i^2 + \lambda_0^2\right)^2}}{\overline{\sigma}^2 \sum_{i=1}^{r} \frac{\varpi_i}{\left(\sigma_i^2 + \lambda_0^2\right)^2}}. \tag{2.9}$$

Portanto,

$$\lambda_0^2 > \frac{\lambda_0^4}{\overline{\sigma}^2} \Rightarrow 1 > \frac{\lambda_0^2}{\overline{\sigma}^2} \Rightarrow \overline{\sigma}^2 > \lambda_0^2 \tag{2.10}$$

o que é um contradição. Conclusão, $\phi_1(\lambda) \geqslant \lambda$ se $\lambda \geqslant \overline{\sigma}$.

□

O lema 2.1 é útil para localizar pontos-fixos da função $\phi_1(\lambda)$ (quando eles existem) e classificar pontos extremos da função $\Psi_\mu(\lambda)$. Este é o assunto do próximo teorema.

Teorema 2.1. *Assuma que $\mu = 1$. Seja I_1 e I_2 intervalos abertos tal que $\phi_1(\lambda) < \lambda$, $\forall\lambda \in I_1$ e $\phi_1(\lambda) > \lambda$, $\forall\lambda \in I_2$. Então $\Psi_\mu(\lambda)$ é crescente em I_1 e decrescente em I_2. Além disso, vale:*

(a) Se $\|b_\perp\|_2 = 0$ e $\varpi_i \neq 0$, $i = 1, \ldots, r$, existe um $\check{\lambda} \in]\underline{\sigma}, \overline{\sigma}[$ tal que $\Psi_\mu(\lambda)$ tem um

máximo local em $\check{\lambda}$. *E, se* $\check{\lambda}$ *é o ponto-fixo de* $\phi_1(\lambda)$ *mais próximo de zero e* $\Psi_\mu(\lambda)$ *tem um mínimo local em* λ^*, *então* $\check{\lambda} < \lambda^*$ *e* $\Psi_\mu(\lambda)$ *tem outro máximo local em* $]\lambda^*, \overline{\sigma}[$.

(b) Se $\|b_\perp\|_2 \neq 0$ *e* $\Psi_\mu(\lambda)$ *tem um mínimo em* λ^*, *então existe um parâmetro* $\check{\lambda}$ *em* $]\lambda^*, \overline{\sigma}[$ *no qual* $\Psi_\mu(\lambda)$ *tem um máximo local.*

(c) Seja λ^* *um ponto-fixo de* $\phi_1(\lambda)$ *e seja* P *o ponto na curva-L na escala log-log associado a* λ^*. *Então a curva-L é convexa numa vizinhança de* P *se, e somente se,* λ^* *minimiza localmente* $\Psi_\mu(\lambda)$, *e ela é côncava numa vizinhança de* P *se, e somente se,* λ^* *localmente maximiza* $\Psi_\mu(\lambda)$.

Demonstração. Usando o fato de que $d\mathbf{y}(\lambda)/d\mathbf{x}(\lambda) = -1/\lambda^2$ podemos escrever $\Psi'_\mu(\lambda)$ como

$$\Psi'_\mu(\lambda) = \mathbf{y}(\lambda)^\mu \mathbf{y}'(\lambda) \left(\phi_1(\lambda)^2 - \lambda^2\right). \tag{2.11}$$

Mas como $\mathbf{y}(\lambda) > 0$ e $\mathbf{y}'(\lambda) < 0$ concluímos que $\Psi'_\mu(\lambda) > 0$ sempre que $\lambda \in I_1$ e $\Psi_\mu(\lambda)$ é uma função crescente para $\lambda \in I_1$. Com um raciocínio análogo concluímos que $\Psi_\mu(\lambda)$ é uma função decrescente para $\lambda \in I_2$.

Para provar o item (a) primeiro notemos que se $\|b_\perp\|_2 = 0$ então $\underline{\sigma}$ e $\overline{\sigma}$ não podem ser pontos-fixos, não no contexto de problemas discretos que estamos tratando, pois se, por exemplo, $\overline{\sigma}$ for um ponto-fixo, concluiremos que isso só pode ser possível se $\varpi_i = \overline{\sigma}$, $i = 1, \ldots, r$ e nos problemas em questão isso não acontece uma vez que os valores singulares decaem para zero sem saltos. Então, pelo fato da função $\Psi_\mu(\lambda)$ ser contínua, estar por baixo da reta $z = \lambda$ para $\lambda \leqslant \underline{\sigma}$ e por cima da reta $z = \lambda$ para $\lambda \geqslant \overline{\sigma}$ segue que $\phi_1(\lambda)$ tem pelo menos um ponto-fixo no intervalo $]\underline{\sigma}, \overline{\sigma}[$. A outra parte do item (a) segue imediato das considerações feitas até aqui e do lema 2.1.

Para provar o item (b), primeiro, usando o mesmo argumento anterior, $\overline{\sigma}$ não pode ser ponto-fixo. Usando o lema 2.1 e o fato de que $\mathbf{y}(\lambda) \to 0$ quando $\lambda \to \infty$, segue que $\Psi_\mu(\lambda)$ é uma função decrescente para $\lambda > \overline{\sigma}$ e $\Psi_\mu(\lambda) \to 0$ quando $\lambda \to \infty$. Isto nos mostra que se existirem extremos locais da função $\Psi_\mu(\lambda)$, esses devem estar no intervalo $]0, \overline{\sigma}[$. Logo, pelas observações feitas o item (b) é uma consequência imediata.

Finalmente o item (c). A primeira parte está demonstrada em [46]. Para a segunda parte definimos $u(\lambda) = \log(\mathbf{x}(\lambda))$ e $v(\lambda) = \log(\mathbf{y}(\lambda))$. Então

$$\frac{du}{dv} = -\frac{\phi_1(\lambda)^2}{\lambda^2} \quad \text{e} \quad \frac{d^2v}{du^2}\frac{du}{d\lambda} = -\frac{2}{\lambda^2}\phi_1(\lambda)\left[\phi_1'(\lambda) - \frac{\phi_1(\lambda)}{\lambda}\right]. \tag{2.12}$$

Usando as condições sobre a segunda derivada de uma função sobre convexidade e concavidade segue que $\phi_1'(\lambda^*) > 1$ em uma vizinhança de λ^* se, e somente se, λ^* maximiza $\Psi_\mu(\lambda)$. Deste resultado e usando a equação (2.12) segue (c).

□

Iremos agora discutir a existência de mínimos locais para $\Psi_\mu(\lambda)$. Regińska [46] foi a primeira a estabelecer certas condições para isto, em que a existência de minimizador dentro de $]0, \infty[$ existe sob condições muito particulares da matriz A. O corolário a seguir é uma generalização do resultado em [46] não necessitando de nenhuma condição especial na matriz A e abrange o caso $\|b_\perp\|_2 \neq 0$

Corolário 2.1. *Se $\|b_\perp\|_2 \neq 0$ então existe um $\mu^* > 0$ tal que $\Psi_{\mu^*}(\lambda)$ tem um mínimo local em $]0, \overline{\gamma}[$.*

Demonstração. Vamos assumir que $\mu = 1$. Se $\phi_1(\bar{\lambda}) < \bar{\lambda}$ para algum $\bar{\lambda} \in]0, \overline{\sigma}[$, pela continuidade de $\phi_1(\lambda)$ e por $\phi_\mu(0) > 0$ segue que $(\phi_1(\lambda) - \lambda)$ muda de sinal em $]0, \bar{\lambda}[$. Então existe um λ^* tal que $\phi_1(\lambda^*) = \lambda^*$ com a propriedade de que $\phi_1(\lambda) > \lambda$ para todo λ em algum intervalo a esquerda de λ^* e $\phi_1(\lambda) < \lambda$ para todo λ em algum intervalo a direita de λ^*. Isso significa que $\Psi_\mu(\lambda)$ é uma função decrescente em algum intervalo a esquerda de λ^* e $\Psi_\mu(\lambda)$ é crescente em algum intervalo a direita de λ^*. Consequentemente, para $\mu = 1$ a função tem um mínimo local em $\lambda^* \in]0, \overline{\sigma}[$.

Vamos supor que $\phi_1(\lambda) > \lambda$ para todo $\lambda > 0$. Seja a curvatura da curva-L na escala log-log no ponto $(\log(\mathbf{x}(\lambda), \log(\mathbf{y}(\lambda))$ ser denotada por $m_L(\lambda)$. Então, como $m_L(\lambda) = -\phi_1(\lambda)^2/\lambda^2$, temos que $\phi_\mu(\lambda)^2 = \lambda^2$ com $\mu = -1/m_L(\lambda)$. Isto mostra que qualquer $\lambda > 0$ pode ser um ponto-fixo de $\phi_\mu(\lambda)$ desde que $\mu = -1/m_L(\lambda)$.

Usando o item (c) do teorema 2.1, basta escolher um λ^* que esteja numa região convexa da curva-L (o que sempre existe próximo de zero) e definir $\mu^* = -1/m_L(\lambda^*)$. Neste caso teremos então $\phi_{\mu^*}(\lambda^*) = \lambda^*$.

□

Outra contribuição importante decorrente da análise do algoritmo de ponto-fixo, devido a Bazán e Francisco [3], foi responder completamente a questão com relação à convexidade e concavidade da curva-L.

Vamos começar com um teorema que diz quando a curva-L na escala log-log é convexa num ponto λ.

Teorema 2.2. *A curva-L é convexa em λ se, e somente se,*

$$\phi_1'(\lambda) < \frac{\phi_1(\lambda)}{\lambda}. \tag{2.13}$$

Demonstração. Seja $v = \log(\mathbf{y}(\lambda))$, $u = \log(\mathbf{x}(\lambda))$ e $m_L(\lambda)$ a inclinação da reta tangente num ponto (u, v) da curva-L. Derivando com respeito a λ encontramos

$$\frac{dv}{du} \triangleq m_L(\lambda) = -\frac{\phi_1(\lambda)^2}{\lambda^2} \tag{2.14}$$

O símbolo $\triangleq$ indica "por definição". Tomando a derivada com respeito a λ em ambos os lados da equação acima segue que

$$\begin{aligned} \frac{d^2v}{du^2}\frac{\mathbf{x}'(\lambda)}{\mathbf{x}(\lambda)} &= -\, m_L'(\lambda) && (2.15)\\ &= -\,\frac{2\phi_1(\lambda)}{\lambda}\left[\frac{\lambda\phi_1'(\lambda) - \phi_1(\lambda)}{\lambda^2}\right]. && (2.16) \end{aligned}$$

Disto segue que a curva-L é convexa em $(u(\lambda), v(\lambda))$ se, e somente se, a condição (2.13) vale, pois $\dfrac{\mathbf{x}'(\lambda)}{\mathbf{x}(\lambda)} > 0$.

□

Uma consequência do teorema (2.2) é que os pontos críticos da função $\xi(\lambda) = \phi_1(\lambda)/\lambda$, $\lambda > 0$, determinam as regiões na qual a curva-L é convexa/côncava. Mais precisamente, cada par de zeros consecutivos da equação

$$\xi'(\lambda) = 0 \Leftrightarrow \phi_1'(\lambda) - \frac{\phi_1(\lambda)}{\lambda} = 0 \tag{2.17}$$

determina uma região na qual a curva-L é côncava ou convexa, dependendo do sinal de

$\phi_1'(\lambda) - \dfrac{\phi_1(\lambda)}{\lambda}$ em cada região.

Na figura 2.3 temos as funções $\phi_1'(\lambda)$ junto com a função $\phi_1(\lambda)/\lambda$ na esquerda e, na direita a curva-L associada. Podemos concluir que nos intervalos em que a derivada da função $\phi_1(\lambda)$ está por baixo da função $\phi_1(\lambda)/\lambda$, correspondem com os trechos em que a curva-L apresenta convexidade e quando está por cima corresponde às regiões em que a curva-L é côncava. Notemos que neste caso a curva-L apresenta dois "cantos", isto pode ser uma dificuldade para o critério da curva-L decidir qual o ponto é o mais apropriado.

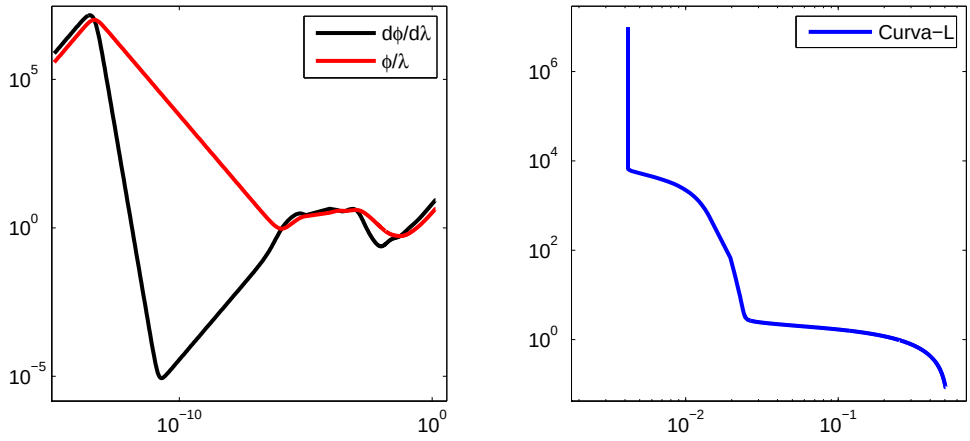

Figura 2.3: Esquerda: funções $\phi_1'(\lambda)$ e $\phi_1(\lambda)/\lambda$. Direita: curva-L. Problema *heat* com 5% de erro no vetor b.

Corolário 2.2. *Assuma que a função $\xi(\lambda)$ têm dois extremos relativos em dois pontos críticos consecutivos de $\xi(\lambda)$, λ_1 e λ_2, com $\lambda_1 < \lambda_2$. Então a curva-L é convexa em $]\lambda_1, \lambda_2[$ se $\xi(\lambda)$ tem um máximo local em λ_1 e a curva-L é côncava em $]\lambda_1, \lambda_2[$ caso contrário.*

Demonstração. Vamos assumir que $\xi(\lambda)$ tenha um máximo local em λ_1 e que $\xi(\lambda)$ seja localmente minimizada em λ_2. Então baseado no fato de que $\xi(\lambda)$ é contínua e diferenciável para $\lambda > 0$, segue que $\xi(\lambda)$ é uma função decrescente em $[\lambda_1, \lambda_2]$, além disso

$$\xi'(\lambda) = \frac{\lambda\phi_1'(\lambda) - \phi_1(\lambda)}{\lambda^2} < 0, \quad \text{para} \quad \lambda_1 < \lambda < \lambda_2. \tag{2.18}$$

Esta desigualdade é equivalente a (2.13), provando a primeira parte do corolário. A segunda parte segue de maneira análoga.

□

O próximo teorema mostra que os pontos-fixos de $\phi_1(\lambda)$, quando existem, fornecem informações que podem ser utilizadas para determinar aproximadamente as regiões em que a curva-L é côncava/convexa.

Teorema 2.3. *Assuma que λ^* é um ponto-fixo de $\phi_1(\lambda)$ e $\phi_1''(\lambda^*) \neq 0$. Então a curva-L é convexa em λ^* se, e somente se, $-\lambda^* \dfrac{\mathbf{y}'(\lambda^*)}{\mathbf{y}(\lambda^*)} < 1$, e quando isso acontece, vale $\lambda^* < \dfrac{\sqrt{3}}{3}\sigma_1$, além disso existe um $\lambda_* \in]\lambda^*, \sigma_1[$ que é um ponto-fixo de $\phi_1(\lambda)$ se, e somente se, a curva-L é côncava em λ_*.*

Demonstração. Derivando com respeito a λ ambos os lados da equação $\phi_1(\lambda)^2 = \dfrac{\mathbf{x}(\lambda)}{\mathbf{y}(\lambda)}$ resulta em

$$2\phi_1(\lambda)\phi_1'(\lambda) = \frac{\mathbf{y}(\lambda)\mathbf{x}'(\lambda) - \mathbf{x}(\lambda)\mathbf{y}'(\lambda)}{\mathbf{y}(\lambda)^2} \tag{2.19}$$

$$= -\frac{\lambda^2\mathbf{y}(\lambda) + \mathbf{x}(\lambda)}{\mathbf{y}(\lambda)^2}\mathbf{y}'(\lambda), \tag{2.20}$$

para chegar nesta expressão usamos o fato de que $\mathbf{x}'(\lambda) = -\lambda^2\mathbf{y}'(\lambda)$. Então temos que

$$\phi_1'(\lambda^*) = -\frac{\lambda^{*2} + \phi_1(\lambda^*)^2}{2\phi_1(\lambda^*)}\frac{\mathbf{y}'(\lambda^*)}{\mathbf{y}(\lambda^*)} = -\frac{\lambda^*\mathbf{y}'(\lambda^*)}{\mathbf{y}(\lambda^*)}. \tag{2.21}$$

Como, pelo teorema 2.2, a convexidade da curva-L é garantida quando $\phi_1'(\lambda) < 1$ segue que

$$-\frac{\lambda^*\mathbf{y}'(\lambda^*)}{\mathbf{y}(\lambda^*)} < 1. \tag{2.22}$$

Vamos agora provar que λ^* não percente ao intervalo $\left[\dfrac{\sqrt{3}}{3}\sigma_1, \infty\right[$ e para isso usaremos o teorema 2.2 que nos diz que uma condição suficiente e necessária para a curva-L ser côncava em λ_* é que $\phi_1'(\lambda_*) > 1$.

Supondo por absurdo que $\lambda^* \geqslant \dfrac{\sqrt{3}}{3}\sigma_1$. Então

$$-\lambda^*\mathbf{y}'(\lambda^*) - \mathbf{y}(\lambda^*) = \sum_{i=1}^{r} \frac{(3\lambda^{*2} - \sigma_i^2)\sigma_i^2\varpi_i}{\left(\sigma_i^2 + \lambda^{*2}\right)^3} > 0 \Leftrightarrow -\frac{\lambda^*\mathbf{y}'(\lambda^*)}{\mathbf{y}(\lambda^*)} > 1. \tag{2.23}$$

Para provar a última parte devemos lembrar que os pontos-fixos da função $\phi_1(\lambda)$, quando existem, pertencem ao intervalo $]0, \sigma_1[$ e que para $\lambda > \sigma_1$ sempre temos $\phi_1(\lambda) \geqslant \lambda$.

Isso implica que a função $\lambda - \phi_1(\lambda)$ muda de sinal no intervalo $]\lambda^*, \sigma_1]$ e então $\phi_1(\lambda_*) = \lambda_*$. Para completar a prova basta vermos que a condição $\phi_1'(\lambda_*) > 1$ é satisfeita numa vizinhança de λ_*.

□

Este teorema nos diz que para casos em que existe mais de um ponto-fixo convexo (ver figura 2.3) o ideal é escolher o maior ponto-fixo convexo e que este não será maior do que $\sigma_1/\sqrt{3}$. Caso o chute inicial seja muito pequeno, as iterações de ponto-fixo

$$\lambda_{k+1} = \phi_\mu(\lambda_k) \tag{2.24}$$

convergirão para um ponto-fixo convexo associado ao ponto de máxima curvatura da curva-L o que resulta numa solução de baixa qualidade.

Capítulo 3

Projeção Sub-Espaço de Krylov

Neste capítulo descreveremos com alguns detalhes o processo de bidiagonalização de Lanczos que transforma a matriz A numa matriz bidiagonal inferior B. O algoritmo LSQR desenvolvido por Paige e Saunders [44, 45] será explicado com detalhes pois esta é uma das bases do algoritmo proposto neste trabalho.

O algoritmo de Lanczos pode ser considerado um caso particular do método de Arnoldi para o cálculo de autovalores de matrizes grandes e esparssas. A idéia do algoritmo de Lanczos é transformar uma matriz A numa matriz tridiagonal e simétrica. Outras aplicações envolvem encontrar a solução de sistemas lineares simétricos.

3.1 Bidiagonalização de Lanczos

Uma matriz bidiagonal pode ser tanto inferior quanto superior. Ao pensarmos em decompor uma matriz $A \in \mathbb{R}^{m\times n}$, $m \geqslant n$, numa matriz bidiagonal devemos levar isso em consideração. Na literatura existem algoritmos para ambas as situações, no entanto, para o algoritmo LSQR o interessante é decompor a matriz A numa matriz bidiagonal inferior.

A idéia da bidiagonalização de Lanczos é decompor a matriz A em

$$A = UBV^T \tag{3.1}$$

em que as matrizes $U \in \mathbb{R}^{m\times(n+1)}$ e $V \in \mathbb{R}^{n\times n}$ tem colunas ortonormais e a matriz $B \in \mathbb{R}^{(n+1)\times n}$ é bidiagonal inferior.

Vamos multiplicar a equação (3.1) pela direita pela matriz V. Além disso, podemos tomar a transposta da matriz A, obtendo a matriz $A^T = VB^TU^T$, e assim multiplicar pela matriz U também pela direita. Deste modo obtemos

$$AV = UB \quad \text{e} \quad A^TU = VB^T. \tag{3.2}$$

O algoritmo é um processo iterativo que no k-ésimo passo produz as matrizes

$$U_{k+1} = [u_1, ..., u_{k+1}] \in \mathbb{R}^{m\times(k+1)}, \qquad V_k = [v_1, ..., v_k] \in \mathbb{R}^{n\times k} \tag{3.3}$$

e

$$B_k = \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \beta_3 & \ddots & \\ & & \ddots & \alpha_k \\ & & & \beta_{k+1} \end{bmatrix} \in \mathbb{R}^{(k+1)\times k}. \tag{3.4}$$

Pelo fato da matriz B ser bidiagonal podemos equacionar as colunas na equação (3.2) e definindo $\beta_1 v_0 \equiv 0$, $\alpha_{k+1}v_{k+1} \equiv 0$, obtemos as seguintes relações

$$A^Tu_j = \beta_j v_{j-1} + \alpha_j v_j, \tag{3.5}$$

$$Av_j = \alpha_j u_j + \beta_{j+1}u_{j+1}, \quad j = 1, ..., k. \tag{3.6}$$

Começando com um vetor $u_1 \in \mathbb{R}^m$, $\|u_1\|_2 = 1$, podemos recursivamente gerar os vetores v_1, u_2, v_2, ..., u_k, v_k, u_{k+1} e os elementos respectivos na matriz B_k, usando, para $j = 1, 2, ..., k$, as fórmulas

$$q_j = A^Tu_j - \beta_j v_{j-1}, \quad \alpha_j = \|q_j\|_2, \quad v_j = q_j/\alpha_j \tag{3.7}$$

$$p_j = Av_j - \alpha_j u_j, \quad \beta_{j+1} = \|p_j\|_2, \quad u_{j+1} = p_j/\beta_{j+1} \tag{3.8}$$

Uma propriedade importante sobre os u_j e v_j que decorre das equações (3.7) e (3.8) é dada na seguinte proposição.

Proposição 3.1. *Para u_j e v_j definidos por (3.7) e (3.8) vale $u_j \in \hat{\mathcal{K}}_j(AA^T, u_1)$ e $v_j \in$*

$\check{\mathcal{K}}_j(A^TA, A^Tu_1)$ *em que*

$$\hat{\mathcal{K}}_j(AA^T, u_1) = span\{u_1, (AA^T)u_1, \ldots, (AA^T)^{j-1}u_1\} \tag{3.9}$$

$$\check{\mathcal{K}}_j(A^TA, A^Tu_1) = span\{A^Tu_1, (A^TA)A^Tu_1, \ldots, (A^TA)^{j-1}A^Tu_1\} \tag{3.10}$$

são os subspaços de Krylov associados às matrizes AA^T *e* A^TA *respectivamente.*

Se escolhermos como vetor inicial u_1 o vetor $b/\|b\|_2$ teremos, pela proposição 3.1, que os vetores u_j, $j = 1, 2, \ldots$, e v_j, $j = 1, 2, \ldots$ pertencerão aos espaços $\hat{\mathcal{K}}_j(AA^T, b)$ e $\check{\mathcal{K}}_j(A^TA, A^Tb)$ respectivamente, além disso

$$\beta_1 u_1 = b, \qquad \alpha_1 v_1 = A^T u_1 \tag{3.11}$$

e usando as relações de recorrência (3.7) e (3.8) sabemos que no passo k da bidiagonalização temos matrizes U_{k+1}, V_k e B_k satisfazendo

$$\beta_1 U_{k+1} e_1 = b, \tag{3.12}$$

$$AV_k = U_{k+1}B_k, \qquad A^TU_{k+1} = V_kB_k^T + \alpha_{k+1}v_{k+1}e_{k+1}^T. \tag{3.13}$$

Veremos que desta maneira podemos calcular soluções aproximadas x_k, $k = 1, 2, \ldots$ para o problema de mínimos quadrados (1.21).

3.2 Melhor aproximação no subespaço de Krylov

O processo de bidiagonalização descrito acima pode ser utilizado para calcular uma sequência de soluções aproximadas x_k, $k \geqslant 1$, para o problema de mínimos quadrados

$$x = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \|b - Ax\|_2^2, \tag{3.14}$$

com $x_k \in \check{\mathcal{K}}_k(A^TA, A^Tb)$. Como $\check{\mathcal{K}}_k(A^TA, A^Tb) = \operatorname{span}(V_k)$, podemos escrever

$$x_k = V_k y_k \tag{3.15}$$

para algum $y_k \in \mathbb{R}^k$.

Vejamos agora o resíduo para um vetor $x_k \in \check{\mathcal{K}}_k(A^TA, A^Tb)$.

Proposição 3.2. *Se $x_k \in \check{\mathcal{K}}_k$ então o resíduo associado r_k é dado por*

$$r_k = \beta_1 U_{k+1} e_1 - U_{k+1} B_k y_k. \tag{3.16}$$

Demonstração. Usando o fato de que $x_k = V_k y_k$ e as equações (3.11) e (3.13) temos

$$r_k = b - Ax_k = \beta_1 u_1 - AV_k y_k = \beta_1 u_1 - U_{k+1} B_k y_k = \beta_1 U_{k+1} e_1 - U_{k+1} B_k y_k.$$

□

Tomando a norma em ambos os lados na equação (3.16) e usando o fato de que a matriz U_{k+1} tem colunas ortonormais segue que

$$\|r_k\|_2 = \|\beta_1 U_{k+1} e_1 - U_{k+1} B_k y_k\|_2 = \|\beta_1 e_1 - B_k y_k\|_2. \tag{3.17}$$

Usando as equações (3.15), (3.17) e considerando que estamos procurando uma solução x_k no subespaço $\check{\mathcal{K}}_k$, ou seja,

$$x_k = \underset{x \in \check{\mathcal{K}}_k}{\operatorname{argmin}} \|b - Ax\|_2^2, \tag{3.18}$$

temos um novo problema que é encontrar a solução $y_k \in \mathbb{R}^k$ tal que

$$y_k = \underset{y \in \mathbb{R}^k}{\operatorname{argmin}} \|\beta_1 e_1 - B_k y\|_2^2. \tag{3.19}$$

Como resolver o problema (3.19) será o assunto da próxima seção.

3.3 O algoritmo LSQR

Uma maneira eficiente de resolver o problema (3.19) é através da decomposição QR da matriz B_k:

$$Q_k B_k = \bar{R}_k = \begin{pmatrix} R_k \\ 0 \end{pmatrix} \in \mathbb{R}^{k+1\times k} \quad \text{e} \quad Q_k(\beta_1 e_1) = \begin{pmatrix} f_k \\ \bar{\phi}_{k+1} \end{pmatrix} \in \mathbb{R}^{k+1} \tag{3.20}$$

em que $R_k \in \mathbb{R}^{k\times k}$ é uma matriz bidiagonal superior,

$$R_k = \begin{pmatrix} \rho_1 & \theta_1 & & & \\ & \rho_2 & \theta_2 & & \\ & & \ddots & \ddots & \\ & & & \rho_{k-1} & \theta_{k-1} \\ & & & & \rho_k \end{pmatrix} \quad \text{e} \quad f_k = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{k-1} \\ \phi_k \end{pmatrix}. \tag{3.21}$$

A matriz Q_k pode ser facilmente obtida através de produtos de rotações de Givens, $Q_k = G_{k,k+1}G_{k-1,k}\cdots G_{1,2}$, escolhidas de modo a eliminar os elementos $\beta_2, ..., \beta_{k+1}$ da matriz B_k. O vetor solução $y_k \in \mathbb{R}^k$ é obtido de

$$R_k y_k = f_k. \tag{3.22}$$

A seguinte proposição mostra uma maneira de calcular a norma do resíduo.

Proposição 3.3. *Se y_k for solução de (3.19) e dado por (3.22), então o resíduo associado é dado por*

$$r_k = Q_k^T \begin{pmatrix} 0 \\ \bar{\phi}_{k+1} \end{pmatrix}. \tag{3.23}$$

Demonstração. Usando o fato de que $I = Q_k^T Q_k$, $B_k = Q_k^T \bar{R}_k$ e $y_k = R_k^{-1} f_k$ temos

$$\begin{aligned} r_k = \beta_1 e_1 - B_k y_k &= Q_k^T Q_k(\beta_1 e_1) - Q_k^T \bar{R}_k R_k^{-1} f_k \\ &= Q_k^T \begin{pmatrix} f_k \\ \bar{\phi}_{k+1} \end{pmatrix} - Q_k^T \begin{pmatrix} I_k \\ 0 \end{pmatrix} f_k \\ &= Q_k^T \begin{pmatrix} 0 \\ \bar{\phi}_{k+1} \end{pmatrix}. \end{aligned} \tag{3.24}$$

□

Por esta proposição podemos facilmente obter a norma para o resíduo, que é

$$\|r_k\|_2 = |\bar{\phi}_{k+1}|. \tag{3.25}$$

A fatoração QR da matriz B_k não precisa ser calculada desde o começo em cada iteração. De modo a facilitar o entendimento iremos mostrar a fatoração QR da matriz B_3 e em seguida a da matriz B_4. A seguir faremos uma comparação entre elas e estabeleceremos uma relação de recorrência.

Começamos com a matriz B_3 e aplicamos uma rotação de Givens, $G_{1,2}$, de modo a eliminar o elemento β_2. Para facilitar a notação escreveremos c e s para $\cos\theta$ e $\operatorname{sen}\theta$, respectivamente, e os espaços em branco são elementos nulos.

$$\begin{bmatrix} c_1 & s_1 & & \\ -s_1 & c_1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 & & \\ \beta_2 & \alpha_2 & \\ & \beta_3 & \alpha_3 \\ & & \beta_4 \end{bmatrix} = \begin{bmatrix} \rho_1 & \theta_1 & \\ 0 & \bar{\rho}_2 & \\ & \beta_3 & \alpha_3 \\ & & \beta_4 \end{bmatrix}.$$

O próximo passo é aplicar uma nova rotação, $G_{2,3}$, para zerar o elemento β_3,

$$\begin{bmatrix} 1 & & & \\ & c_2 & s_2 & \\ & -s_2 & c_2 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} \rho_1 & \theta_1 & \\ 0 & \bar{\rho}_2 & \\ & \beta_3 & \alpha_3 \\ & & \beta_4 \end{bmatrix} = \begin{bmatrix} \rho_1 & \theta_1 & \\ 0 & \rho_2 & \theta_2 \\ & 0 & \bar{\rho}_3 \\ & & \beta_4 \end{bmatrix}.$$

Por fim, a última rotação, $G_{3,4}$, para eliminar β_4,

$$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & c_3 & s_3 \\ & & -s_3 & c_3 \end{bmatrix} \begin{bmatrix} \rho_1 & \theta_1 & & \\ 0 & \rho_2 & \theta_2 & \\ & 0 & \bar{\rho}_3 & \\ & & & \beta_4 \end{bmatrix} = \begin{bmatrix} \rho_1 & \theta_1 & & \\ 0 & \rho_2 & \theta_2 & \\ & 0 & \rho_3 & \\ & & 0 & \end{bmatrix} = \bar{R}_3.$$

Deste modo a matriz Q_3 será o produto das três rotações

$$Q_3 = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & c_3 & s_3 \\ & & -s_3 & c_3 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & c_2 & s_2 & \\ & -s_2 & c_2 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} c_1 & s_1 & & \\ -s_1 & c_1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}.$$

Faremos agora todo o procedimento para a matriz B_4. Então, começando novamente com uma rotação para zerar β_2, denotaremos por $\tilde{G}_{1,2}$,

$$\begin{bmatrix} c_1 & s_1 & & & \\ -s_1 & c_1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 & & & & \\ \beta_2 & \alpha_2 & & & \\ & \beta_3 & \alpha_3 & & \\ & & \beta_4 & \alpha_4 & \\ & & & \beta_5 & \end{bmatrix} = \begin{bmatrix} \rho_1 & \theta_1 & & & \\ 0 & \bar{\rho}_2 & & & \\ & \beta_3 & \alpha_3 & & \\ & & \beta_4 & \alpha_4 & \\ & & & \beta_5 & \end{bmatrix}.$$

Em seguida as rotações $\tilde{G}_{2,3}$ e $\tilde{G}_{3,4}$,

$$\begin{bmatrix} 1 & & & & \\ & c_2 & s_2 & & \\ & -s_2 & c_2 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} \rho_1 & \theta_1 & & & \\ 0 & \bar{\rho}_2 & & & \\ & \beta_3 & \alpha_3 & & \\ & & \beta_4 & \alpha_4 & \\ & & & \beta_5 & \end{bmatrix} = \begin{bmatrix} \rho_1 & \theta_1 & & & \\ 0 & \rho_2 & \theta_2 & & \\ & 0 & \bar{\rho}_3 & & \\ & & \beta_4 & \alpha_4 & \\ & & & \beta_5 & \end{bmatrix}$$

e

$$\begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & c_3 & s_3 & \\ & & -s_3 & c_3 & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} \rho_1 & \theta_1 & & & \\ 0 & \rho_2 & \theta_2 & & \\ & 0 & \bar{\rho}_3 & & \\ & & \beta_4 & \alpha_4 & \\ & & & \beta_5 & \end{bmatrix} = \begin{bmatrix} \rho_1 & \theta_1 & & & \\ 0 & \rho_2 & \theta_2 & & \\ & 0 & \rho_3 & \theta_3 & \\ & & 0 & \bar{\rho}_4 & \\ & & & \beta_5 & \end{bmatrix}.$$

Por fim a última rotação, $\tilde{G}_{4,5}$, de modo a elimitar β_5,

$$\begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & c_4 & s_4 \\ & & & -s_4 & c_4 \end{bmatrix} \begin{bmatrix} \rho_1 & \theta_1 & & & \\ 0 & \rho_2 & \theta_2 & & \\ & 0 & \rho_3 & \theta_3 & \\ & & 0 & \bar{\rho}_4 & \\ & & & & \beta_5 \end{bmatrix} = \begin{bmatrix} \rho_1 & \theta_1 & & & \\ 0 & \rho_2 & \theta_2 & & \\ & 0 & \rho_3 & \theta_3 & \\ & & 0 & \rho_4 & \\ & & & 0 & \end{bmatrix} = \bar{R}_4.$$

e assim a matriz Q_4 será o produto das quatro rotações aplicadas.

Analisando as matrizes de rotações $G_{1,2}$ e $\tilde{G}_{1,2}$ podemos perceber que, usando a notação do MatLab, $\tilde{G}_{1,2}(1{:}4,1{:}4) = G_{1,2}$, ou seja, a primeira matriz de rotação foi recalculada e isso se repetirá para as matrizes B_5, B_6,.... Não apenas esta rotação tem esta característica, como a segunda e a terceira também. De uma maneira mais geral, removendo a última coluna e a última linha de $\tilde{G}_{k,k+1}$ teremos as rotações $G_{k,k+1}$.

Na matriz B_3 quando aplicamos $G_{1,2}$ alteramos apenas as duas primeiras linhas da matriz e as demais linhas foram mantidas iguais. Na matriz B_4 após a primeira rotação isso se repete, isto é, alteramos as duas primeiras linhas e as demais não foram alteradas. Além disso, os elementos resultantes foram os mesmo, isto é, ρ_1, θ_1 e $\bar{\rho}_2$ são os mesmos nas matrizes B_3 e B_4. Podemos concluir disto que ao passarmos da matriz de B_3 para a matriz B_4, ou seja, da iteração $k = 3$ para a iteração $k = 4$, a primeira rotação não precisa ser refeita. O mesmo ocorrendo com a segunda e a terceira rotação.

Pelo fato da matriz B_4 ter mais uma coluna e consequentemente mais uma linha, a terceira rotação obrigatoriamente afetará este incremento e notemos que só a terceira e a quarta rotação alteram os valores da quarta coluna, ou seja, uma vez que temos a decomposição QR da matriz B_3 e progredimos para calcular a decomposição QR da matriz B_4, precisamos apenas "guardar" a matriz $G_{2,3}$, calcular a rotação $\tilde{G}_{3,4}$ e aplicarmos essas duas rotações na última coluna da matriz B_4 pois as duas primeiras rotações não alteram os resultados da última coluna.

O mesmo procedimento pode ser feito com o vetor $Q_k(\beta_1 e_1)$. Quando $k = 3$ temos

$$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & c_3 & s_3 \\ & & -s_3 & c_3 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & c_2 & s_2 & \\ & -s_2 & c_2 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} c_1 & s_1 & & \\ -s_1 & c_1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{pmatrix} \beta_1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \bar{\phi}_4 \end{pmatrix}$$

e quando passarmos para $k = 4$ teremos algo muito semelhante, porém com uma linha a mais no vetor $\beta_1 e_1$

$$\begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & c_4 & s_4 \\ & & & -s_4 & c_4 \end{bmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \bar{\phi}_4 \\ 0 \end{pmatrix} = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \bar{\phi}_5 \end{pmatrix}.$$

Com base neste exemplo faremos agora uma análise para o caso genérico, ou seja, como aproveitar a QR da matriz B_{k-1} para calcular a QR da matriz B_k.

Vamos assumir que a fatoração QR da matriz B_{k-1} tenha sido calculada. Mostraremos abaixo apenas os últimos três elementos da última coluna da matriz B_k e os dois últimos elementos do vetor $Q_k(\beta_1 e_1)$. Na próxima iteração a k-ésima coluna é adicionada e uma rotação é determinada de tal modo que

$$G_{k,k+1} G_{k-1,k} \begin{pmatrix} 0 \\ \alpha_k \\ \beta_{k+1} \end{pmatrix} = \begin{pmatrix} \theta_{k-1} \\ \rho_k \\ 0 \end{pmatrix} \quad \text{e} \quad G_{k,k+1} \begin{pmatrix} \bar{\phi}_k \\ 0 \end{pmatrix} = \begin{pmatrix} \phi_k \\ \bar{\phi}_{k+1} \end{pmatrix}. \tag{3.26}$$

Note que as rotações $G_{k-2,k-1}$,...,$G_{1,2}$ não afetam a coluna k como visto no exemplo acima.

Para entendermos melhor esta parte, faremos primeiro o produto da matriz $G_{k-1,k}$ pela k-ésima coluna

$$\begin{bmatrix} c_{k-1} & s_{k-1} & \\ -s_{k-1} & c_{k-1} & \\ & & 1 \end{bmatrix} \begin{pmatrix} 0 \\ \alpha_k \\ \beta_{k+1} \end{pmatrix} = \begin{pmatrix} s_{k-1}\alpha_k \\ c_{k-1}\alpha_k \\ \beta_{k+1} \end{pmatrix} = \begin{pmatrix} \theta_{k-1} \\ \bar{\rho}_k \\ \beta_{k+1} \end{pmatrix}, \tag{3.27}$$

disso resulta que $\theta_{k-1} = s_{k-1}\alpha_k$. Aplicando agora no vetor resultante a rotação $G_{k,k+1}$ temos

$$\begin{bmatrix} 1 & & \\ & c_k & s_k \\ & -s_k & c_k \end{bmatrix} \begin{pmatrix} \theta_{k-1} \\ \bar{\rho}_k \\ \beta_{k+1} \end{pmatrix} = \begin{pmatrix} \theta_{k-1} \\ \rho_k \\ 0 \end{pmatrix} \tag{3.28}$$

e para o vetor $G_{k,k+1} \begin{pmatrix} \bar{\phi}_k \\ 0 \end{pmatrix}$ temos

$$\begin{bmatrix} c_k & s_k \\ -s_k & c_k \end{bmatrix} \begin{pmatrix} \bar{\phi}_k \\ 0 \end{pmatrix} = \begin{pmatrix} \phi_k \\ \bar{\phi}_{k+1} \end{pmatrix} \tag{3.29}$$

o que resulta

$$\phi_k = c_k\bar{\phi}_k \quad \text{e} \quad \bar{\phi}_{k+1} = -s_k\bar{\phi}_k. \tag{3.30}$$

Como o resíduo é dado por $\|r_k\|_2 = |\bar{\phi}_{k+1}|$ (ver equação (3.25)) segue da equação (3.30) que o resíduo pode ser escrito como

$$\|r_k\|_2 = |\bar{\phi}_{k+1}| = |s_k\bar{\phi}_k| = \cdots = |s_k s_{k-1} \ldots s_1\beta_1|. \tag{3.31}$$

De modo a formarmos o vetor x_k por $x_k = V_k y_k$ como definido anteriormente, será necessário armazenar os vetores $v_1, ..., v_k$. Isso pode ser evitado como mostrado por Paige e Saunders [44]. Eles determinaram uma relação de recorrência para calcular x_k a partir de x_{k-1}. Subtituindo na equação (3.15) y_k por $R_k^{-1}f_k$ obtemos

$$x_k = V_k y_k = V_k(R_k^{-1}f_k) = (V_k R_k^{-1})f_k = Z_k f_k. \tag{3.32}$$

Aqui Z_k satisfaz o sistema triangular inferior $R_k^T Z_k^T = V_k^T$, logo as colunas de $Z_k = (z_1, z_2, ..., z_k)$ podem ser encontradas sucessivamente por substituição direta. Definindo $z_0 = x_0 = 0$, usando a matriz R_k e identificando as últimas colunas em $Z_k R_k = V_k$ temos

$$z_k = \frac{1}{\rho_k}(v_k - \theta_{k-1}z_{k-1}) \quad \text{e} \quad x_k = x_{k-1} + \phi_k z_k, \tag{3.33}$$

uma pequena otimização pode ser feita usando o vetor $w_k = \rho_k z_k$.

Antes de apresentarmos o algoritmo, vamos mostrar que a norma do resíduo é uma função decrescente e a norma da solução é uma função crescente.

Proposição 3.4. *Se x_k, $k = 1, 2, \ldots$ forem as soluções iteradas para o algoritmo LSQR e r_k, $k = 1, 2, \ldots$ forem os resíduos associados, então*

$$\|r_{k+1}\|_2 \leqslant \|r_k\|_2, \quad k = 1, \ldots, n-1, \tag{3.34}$$

$$\|x_{k+1}\|_2 \geqslant \|x_k\|_2, \quad k = 1, \ldots, n-1. \tag{3.35}$$

Demonstração. Para mostrarmos que o resíduo é decrescente basta observamos que da equação (3.31) temos

$$\|r_{k+1}\|_2 = |\bar{\phi}_{k+2}| = |s_{k+1}\bar{\phi}_{k+1}| \leqslant |\bar{\phi}_{k+1}| = \|r_k\|_2. \tag{3.36}$$

Para demonstrarmos que a norma da solução $\|x_k\|_2$ é uma função crescente notemos o seguinte. A matriz R_k pode ser reduzida a uma matriz bidiagonal inferior $\bar{L}_k$ através de uma fatoração ortogonal

$$R_k \bar{Q}_k^T = \bar{L}_k, \tag{3.37}$$

em que $\bar{Q}_k$ é um produto de rotações de Givens. Definindo $\bar{z}_k$ pelo sistema

$$\bar{L}_k \bar{z}_k = f_k, \tag{3.38}$$

segue que $x_k = (V_k R_k^{-1}) f_k = (V_k \bar{Q}_k^T)\bar{z}_k = \bar{W}_k \bar{z}_k$. Então, usando o fato de que $V_k^T V_k = I$, obtemos

$$\|x_k\|_2 = \|\bar{z}_k\|_2. \tag{3.39}$$

Notemos que a parte superior de $\bar{L}_k, \bar{Q}_k, \bar{W}_k$ e $\bar{z}_k$ não muda depois da iteração k. Então da iteração k para a iteração $k+1$ o vetor $\bar{z}_k$ apenas adquire mais um elemento, ou seja,

$$\|x_k\|_2 = \|\bar{z}_k\|_2 \leqslant \|\bar{z}_{k+1}\|_2 = \|x_{k+1}\|_2. \tag{3.40}$$

□

Segue agora um esquema do algoritmo LSQR.

Algoritmo LSQR:

Entrada: A, b

- $x_0 := 0$
- $\beta_1 u_1 := b$
- $\alpha_1 v_1 = A^T u_1$
- $w_1 = v_1$
- $\bar{\phi}_1 = \beta_1$
- $\bar{\rho}_1 = \alpha_1$

Para $i = 1, 2, ...$

- $\beta_{i+1} u_{i+1} = A v_i - \alpha_i u_i$
- $\alpha_{i+1} v_{i+1} = A u_{i+1} - \beta_{i+1} v_i$
- $[c_i, s_i, \rho_i] = givrot(\bar{\rho}_i, \beta_{i+1})$
- $\theta_i = s_i \alpha_{i+1}$
- $\bar{\rho}_{i+1} = c_i \alpha_{i+1}$
- $\phi_i = c_i \bar{\phi}_i$
- $\bar{\phi}_{i+1} = -s_i \bar{\phi}_i$
- $x_i = x_{i-1} + (\phi_i / \rho_i) w_i$
- $w_{i+1} = v_{i+1} - (\theta_i / \rho_i) w_i$

Fim Para

A função *givrot* é uma função que "gera" a matriz G da rotação de Givens tal que

$$\begin{bmatrix} \cos\theta & \operatorname{sen}\theta \\ -\operatorname{sen}\theta & \cos\theta \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \rho \\ 0 \end{pmatrix} \tag{3.41}$$

mas na prática apenas os valores $\cos\theta$, $\operatorname{sen}\theta$ e ρ são retornados.

Uma das dificuldades em se utilizar tal método é que este algoritmo é semi-convergente, ou seja, à medida que as iteração evoluem as soluções iteradas se aproximam da solução exata, mas a partir de dado momento as soluções iteradas passam a se distanciar em vez do algoritmo estabilizar, exatamente como o método TSVD.

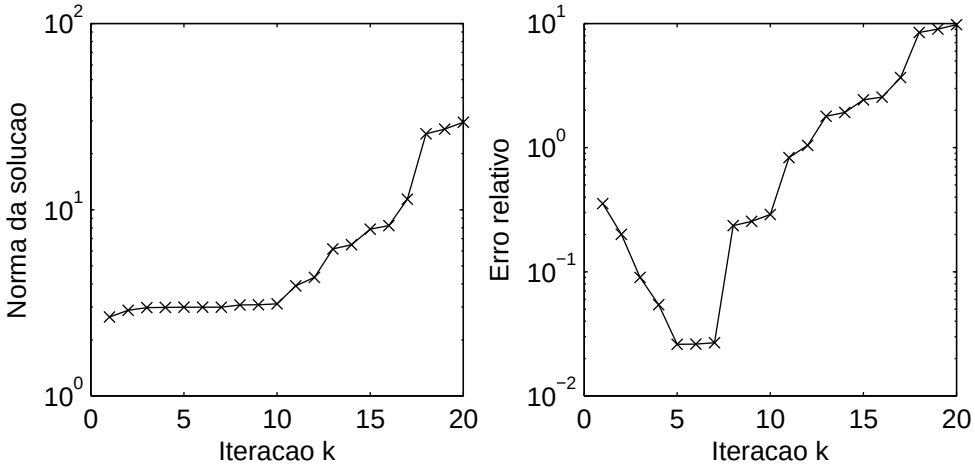

Figura 3.1: Semi-convergência para o algoritmo LSQR.

Este fenômeno pode ser verificado na figura 3.1. Para ilustrar usamos o problema *phillips* com dimensão $n = 32$ e erro relativo nos dados de 1%. Analisando o gráfico da esquerda constatamos que até a iteração $k = 10$ a norma da solução tem um comportamento praticamente estável (no sentido de não variar muito), porém, se o algoritmo continuar as iterações, a norma da solução cresce rapidamente. No gráfico da direita temos os erros relativos para as primeiras 20 soluções iteradas, o menor erro relativo ocorre na iteração $k = 5$ e para este k a norma da solução é praticamente a mesma para $k = 10$ apesar do erro relativo não ser o mais apropriado.

Uma maneira de contornar esta dificuldade é com a utilização de um parâmetro de regularização apropriado e este é o assunto da próxima seção.

3.4 Regularização de Tikhonov e LSQR

Na regularização de Tikhonov temos

$$x_\lambda = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\{\|b - Ax\|_2^2 + \lambda^2\|x\|_2^2\} \tag{3.42}$$

em que λ é o parâmetro de regularização. Como já fora comentado, numericamente o melhor modo de tratar o problema (3.42) é por mínimos quadrados

$$x_\lambda = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}} \left\| \begin{pmatrix} b \\ 0 \end{pmatrix} - \begin{pmatrix} A \\ \lambda I \end{pmatrix} x \right\|_2^2 . \tag{3.43}$$

Como no caso em que não há regularização, após projetarmos o problema original, temos o seguinte problema

$$y_{k,\lambda} = \underset{y\in\mathbb{R}^k}{\operatorname{argmin}} \left\| \begin{pmatrix} \beta_1 e_1 \\ 0 \end{pmatrix} - \begin{pmatrix} B_k \\ \lambda I_k \end{pmatrix} y \right\|_2^2 \tag{3.44}$$

e a solução $x_{k,\lambda}$ é obtida por

$$x_{k,\lambda} = V_k y_{k,\lambda}. \tag{3.45}$$

Podemos aplicar diretamente o algoritmo LSQR à matriz $\bar{A} = \begin{pmatrix} A \\ \lambda I \end{pmatrix}$ e ao vetor $\bar{b} = \begin{pmatrix} b \\ 0 \end{pmatrix}$, porém, utilizando apenas mais uma rotação no algoritmo LSQR podemos resolver o problema de mínimos quadrados acima de maneira eficiente, mais uma vez como mostrado por Paige e Saunders [45].

Vejamos um pequeno exemplo com a matriz B_2. Primeiro aplicamos uma rotação de Givens $G_{1,\lambda}$ nos elementos α_1 e λ com a intenção de zerar λ. Em seguida aplicamos uma nova rotação, $G_{1,2}$, para zerar o elemento β_2. A seguir mais duas rotações para anular o elemento β_3 e o elemento λ. Obviamente que estas rotações influenciam o vetor $[\beta_1 e_1; 0]$, porém de uma maneira muito simples. Abaixo do elemento $\bar{\phi}_3$ aparecem elementos ψ_1, ψ_2. Segue um esquema desta descrição considerando a matriz aumentada $[B_2|\ \beta_1 e_1]$

$$\left[\begin{array}{ccccc} \tilde{c}_1 & & & \tilde{s}_1 & \\ & 1 & & & \\ & & 1 & & \\ -\tilde{s}_1 & & & \tilde{c}_1 & \\ & & & & 1 \end{array}\right] \left[\begin{array}{cc|c} \alpha_1 & & \beta_1 \\ \beta_2 & \alpha_2 & \\ & \beta_3 & \\ \lambda & & \\ & \lambda & \end{array}\right] = \left[\begin{array}{cc|c} \bar{\rho}_1 & & \bar{\phi}_1 \\ \beta_2 & \alpha_2 & \\ & \beta_3 & \\ 0 & & \psi_1 \\ & \lambda & \end{array}\right].$$

Multiplicando agora por uma rotação para zerar o elemento β_2, temos

$$\left[\begin{array}{ccccc} c_1 & s_1 & & & \\ -s_1 & c_1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{array}\right] \left[\begin{array}{cc|c} \bar{\rho}_1 & & \bar{\phi}_1 \\ \beta_2 & \alpha_2 & \\ & \beta_3 & \\ 0 & & \psi_1 \\ & \lambda & \end{array}\right] = \left[\begin{array}{cc|c} \rho_1 & \theta_1 & \phi_1 \\ 0 & \bar{\rho}_2 & \bar{\phi}_2 \\ & \beta_3 & \\ 0 & & \psi_1 \\ & \lambda & \end{array}\right].$$

Prosseguindo temos

$$\left[\begin{array}{ccccc} 1 & & & & \\ & \tilde{c}_2 & & & \tilde{s}_2 \\ & & 1 & & \\ & & & 1 & \\ & -\tilde{s}_2 & & & \tilde{c}_2 \end{array}\right] \left[\begin{array}{cc|c} \rho_1 & \theta_1 & \phi_1 \\ 0 & \bar{\rho}_2 & \bar{\phi}_2 \\ & \beta_3 & \\ 0 & & \psi_1 \\ & \lambda & \end{array}\right] = \left[\begin{array}{cc|c} \rho_1 & \theta_1 & \phi_1 \\ 0 & \bar{\rho}_2 & \bar{\phi}_2 \\ & \beta_3 & \\ 0 & & \psi_1 \\ & 0 & \psi_2 \end{array}\right].$$

Multiplicando agora por uma rotação para zerar o elemento β_2, temos

$$\left[\begin{array}{ccccc} 1 & & & & \\ & c_2 & s_2 & & \\ & -s_2 & c_2 & & \\ & & & 1 & \\ & & & & 1 \end{array}\right] \left[\begin{array}{cc|c} \rho_1 & \theta_1 & \phi_1 \\ 0 & \bar{\rho}_2 & \bar{\phi}_2 \\ & \beta_3 & \\ \hline 0 & & \psi_1 \\ & 0 & \psi_2 \end{array}\right] = \left[\begin{array}{cc|c} \rho_1 & \theta_1 & \phi_1 \\ 0 & \rho_2 & \phi_2 \\ & 0 & \bar{\phi}_3 \\ \hline 0 & & \psi_1 \\ & 0 & \psi_2 \end{array}\right],$$

sendo assim, a matriz Q_2 seria o produto das quatro rotações.

Resumindo, as rotações adicionais afetam o vetor $\beta_1 e_1$ de modo que tenhamos algo como

$$Q_k \begin{pmatrix} \beta_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} f_k \\ \bar{\phi}_{k+1} \\ g_k \end{pmatrix} = \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_k \\ \bar{\phi}_{k+1} \\ \psi_1 \\ \vdots \\ \psi_k \end{pmatrix}. \tag{3.46}$$

Contudo, apesar desta mudança, não há alteração na elaboração do algoritmo no que diz respeito à montagem da solução, isto é, o vetor g_k não influencia na solução $x_{k,\lambda}$. De fato isto acontece pois após a decomposição QR continuamos a ter o sistema $R_k y_{k,\lambda} = f_k$ como na equação (3.22) sendo R_k a matriz bidiagonal superior e o vetor f_k dado da mesma maneira.

Procedendo do mesmo modo como feito na equação (3.24) podemos calcular o resíduo para (3.44).

Proposição 3.5. *Se $y_{k,\lambda}$ for solução para (3.44), então o resíduo associado $\bar{r}_{k,\lambda}$ é dado por*

$$\bar{r}_{k,\lambda} = Q_k^T \begin{pmatrix} 0 \\ \bar{\phi}_{k+1} \\ g_k \end{pmatrix}. \tag{3.47}$$

Demonstração. Novamente, usando a decomposição QR da matriz $\begin{pmatrix} B_k \\ 0 \end{pmatrix}$ temos

$$\begin{aligned}
\bar{r}_{k,\lambda} = \begin{pmatrix} \beta_1 e_1 \\ 0 \end{pmatrix} - \begin{pmatrix} B_k \\ \lambda I \end{pmatrix} y_{k,\lambda} &= Q_k^T Q_k \begin{pmatrix} \beta_1 e_1 \\ 0 \end{pmatrix} - Q_k^T \bar{R}_k R_k^{-1} f_k \\
&= Q_k^T \begin{pmatrix} f_k \\ \bar{\phi}_{k+1} \\ g_k \end{pmatrix} - Q_k^T \begin{pmatrix} I_k \\ 0 \end{pmatrix} f_k \\
&= Q_k^T \begin{pmatrix} 0 \\ \bar{\phi}_{k+1} \\ g_k \end{pmatrix}.
\end{aligned} \tag{3.48}$$

□

Seja $r_{k,\lambda}$ o resíduo correspondente à solução regularizada $x_{k,\lambda}$, isto é, $r_{k,\lambda} = b - Ax_{k,\lambda}$. A proposição acima nos possibilita calcular a norma do resíduo do problema de uma maneira muito simples. De fato, temos que

$$\|\bar{r}_{k,\lambda}\|_2^2 = \bar{\phi}_{k+1}^2 + \|g_k\|_2^2 \tag{3.49}$$

e como $\|\bar{r}_{k,\lambda}\|_2^2 = \|\beta_1 e_1 - B_k y_{k,\lambda}\|_2^2 + \lambda^2 \|y_{k,\lambda}\|_2^2$ segue que

$$\|\beta_1 e_1 - B_k y_{k,\lambda}\|_2^2 = \bar{\phi}_{k+1}^2 + \|g_k\|_2^2 - \lambda^2 \|y_{k,\lambda}\|_2^2 \tag{3.50}$$

que é a norma do resíduo associado a $x_{k,\lambda}$. Isto segue de

$$\begin{aligned}
\bar{\phi}_{k+1}^2 + \|g_k\|_2^2 = \|\bar{r}_{k,\lambda}\|_2^2 &= \left\| \begin{pmatrix} \beta_1 e_1 \\ 0 \end{pmatrix} - \begin{pmatrix} B_k \\ \lambda I \end{pmatrix} y_{k,\lambda} \right\|_2^2 \\
&= \|\beta_1 e_1 - B_k y_{k,\lambda}\|_2^2 + \lambda^2 \|y_{k,\lambda}\|_2^2.
\end{aligned} \tag{3.51}$$

A próxima proposição mostra que, para λ fixo, a norma da solução é uma função crescente e a norma do resíduo é uma função decrescente.

Proposição 3.6. *Dado $\lambda > 0$, então*

$$\|x_{k+1,\lambda}\|_2 \geqslant \|x_{k,\lambda}\|_2, \quad k = 1, \ldots, n-1, \tag{3.52}$$

$$\|r_{k+1,\lambda}\|_2 \leqslant \|r_{k,\lambda}\|_2, \quad k = 1, \ldots, n-1. \tag{3.53}$$

Demonstração. A primeira desigualdade é uma consequência imediata das propriedades monotônicas do algoritmo LSQR. Para provarmos a segunda desigualdade notemos que a solução regularizada $x_{k,\lambda}$ satisfaz

$$x_{k,\lambda} = \underset{x \in \check{\mathcal{K}}_k}{\operatorname{argmin}} \left\{ \|b - Ax\|_2^2 + \lambda^2 \|x\|_2^2 \right\}. \tag{3.54}$$

Agora, como $x_{k,\lambda} \in \check{\mathcal{K}}_{k+1}$ segue que

$$\lambda^2 \|x_{k+1,\lambda}\|_2^2 + \|r_{k+1,\lambda}\|_2^2 \leqslant \lambda^2 \|x_{k,\lambda}\|_2^2 + \|r_{k,\lambda}\|_2^2, \tag{3.55}$$

e usando a equação (3.52) temos

$$\|r_{k+1,\lambda}\|_2^2 \leqslant \lambda^2 \|x_{k+1,\lambda}\|_2^2 - \lambda^2 \|x_{k,\lambda}\|_2^2 + \|r_{k+1,\lambda}\|_2^2 \leqslant \|r_{k,\lambda}\|_2^2, \tag{3.56}$$

o que prova a segunda desigualdade.

□

Assim, a regularização essencialmente acrescenta mais uma rotação uma vez que não precisamos armazenar o vetor g_k.

Como comentado anteriormente, uma desvantagem do método LSQR é o fato deste algoritmo ter a propriedade de ser semi-convergente, na figura 3.2 (esquerda) temos os erros relativos para as 25 primeiras soluções obtidas usando o algoritmo LSQR com reortogonalização para o problema *shaw* com dimensão $n = 32$ e erro relativo nos dados de 4% e na parte da direita temos os erros relativos quando utilizamos alguns parâmetros de regularização. Podemos perceber que quando utilizamos um parâmetro apropriado o método tende a estabilizar a solução, fazendo com que neste caso a escolha da iteração

de parada seja uma tarefa menos importante, isto é, na pior das hipóteses basta executar $k = n$ iterações para obtermos a solução.

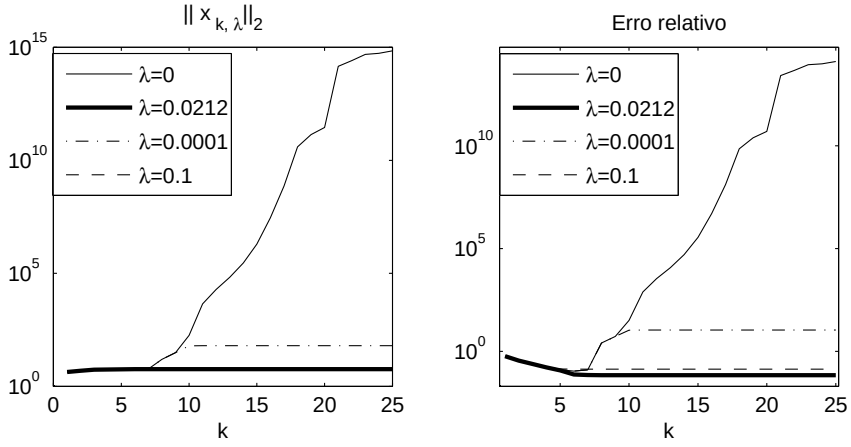

Figura 3.2: Fenômeno de semi-convergência do LSQR e estabilização com parâmetro de regularização apropriado.

Para deixar as coisas mais claras, quando temos $\lambda = 0$ estamos nos referindo ao caso em que não há regularização e neste caso está sendo usado o algoritmo LSQR na sua forma padrão.

Capítulo 4

Lanc-FP: um algoritmo para problemas discretos mal-postos de grande porte

Neste capítulo iremos desenvolver um algoritmo para obter uma solução aproximada do problema (3.42) baseado em considerações a respeito dos algoritmos de ponto-fixo para determinação do parâmetro de Tikhonov e LSQR.

Como a dimensão do subespaço projetado é usualmente muito menor do que a dimensão do problema original, regularizar o problema projetado é bem mais barato do ponto de vista computacional e no final a solução obtida é tão boa quanto à encontrada ao se aplicar um método de regularização diretamente no problema original.

O algoritmo proposto combina a regularização de Tikhonov com o algoritmo de ponto-fixo. Do algoritmo LSQR podemos facilmente obter normas da solução e do resíduo associado, quantidades necessárias para a avaliação da função $\phi_\mu(\lambda)$ do algoritmo de ponto-fixo (1.83). A norma do resíduo calculamos de acordo com a equação (3.50) e a norma da solução por $\|y_{k,\lambda}\|_2$ pois $x_{k,\lambda} = V_k y_{k,\lambda}$ e V_k têm colunas ortonormais.

Vamos introduzir uma sequência de funções $\phi_\mu^{(k)} : \mathbb{R}^+ \to \mathbb{R}$, $k = 2, \ldots, n$, definidas por

$$\phi_\mu^{(k)}(\lambda) = \sqrt{\mu}\frac{\|\beta_1 e_1 - B_k y_{k,\lambda}\|_2}{\|y_{k,\lambda}\|_2}, \;\; \mu > 0. \tag{4.1}$$

em que e_1 denota o vetor canônico $[1, 0, 0, ..., 0]^T$ (em cada iteração $e_1 \in \mathbb{R}^{k+1}$), $\beta_1 = \|b\|_2$,

B_k a matriz bidiagonal definida por (3.4) e $y_{k,\lambda}$ solução de (3.44).

Seja a decomposição em valores singulares da matriz B_k

$$B_k = P_k \begin{pmatrix} \Omega_k \\ 0 \end{pmatrix} Q^T = \sum_{i=1}^{k} \omega_i p_i q_i^T, \tag{4.2}$$

em que as matrizes P_k e Q_k são ortogonais e

$$\omega_1 \geqslant \omega_2 \geqslant \cdots \geqslant \omega_k > 0. \tag{4.3}$$

Então é imediato verificar que

$$\|y_{k,\lambda}\|_2^2 = \beta_1^2 \sum_{i=1}^{k} \frac{\omega_i^2 \xi_{1i}^2}{(\omega_i^2 + \lambda^2)^2}, \tag{4.4}$$

$$\|r_{k,\lambda}\|_2^2 = \beta_1^2 \left(\sum_{i=1}^{k} \frac{\lambda^4 \xi_{1i}^2}{(\omega_i^2 + \lambda^2)^2} + {\delta_0^{(k)}}^2 \right), \tag{4.5}$$

em que $\xi_{1i} = P_k(1,i)$, e $\delta_0^{(k)}$ é a norma-2 da parte incompatível de $\beta_1 e_1$ que está fora do espaço coluna da matriz B_k, $\mathcal{R}(B_k)$, da mesma maneira como foi definido $\|b_\perp\|_2$. Analogamente à função $\phi_1(\lambda)$ temos

$$\phi_\mu^{(k)\prime}(\lambda) > 0, \quad \text{para} \quad \lambda > 0, \tag{4.6}$$

em que $'$ denota a derivada com respeito a λ, isto é, $\phi_\mu^{(k)}(\lambda)$ é estritamente crescente com λ.

Agora, para $\lambda_0^{(k)}$ uma aproximação inicial na iteração k, vamos considerar a sequência

$$\lambda_{j+1}^{(k)} = \phi_1^{(k)}(\lambda_j^{(k)}), \quad j \geqslant 0, \tag{4.7}$$

e assuma que esta sequência converge para um ponto-fixo ${\lambda^{(k)}}^*$ de $\phi_1^{(k)}(\lambda)$; quando isso é verdade e o tamanho da componente de b que está fora do espaço coluna da matriz A é maior do que zero , isto é, $\delta_0^{(n)} = \|b_\perp\|_2 \neq 0$, vamos provar que a função $\phi_1(\lambda)$ do problema original sempre tem um ponto-fixo λ^* que minimiza $\Psi_1(\lambda)$ e que a sequência de pontos-fixos ${\lambda^{(k)}}^*$ converge para λ^* em, no máximo, n passos.

O algoritmo proposto gera uma sequência de valores para o maior ponto-fixo convexo de $\phi_1(\lambda)$, λ^*, usando uma sequência finita de pontos-fixos $\lambda^{(k)^*}$. Isto requer resolver o problema projetado (3.44) para vários valores de λ para um k fixo (mas crescente) e, para isso, usando o algoritmo LSQR. Segue uma breve descrição do algoritmo na tabela 4.1.

LANC-FP
Dados de Entrada: A, b, $p > 1$, $k_{\max}$, ϵ.
Output: Solução regularizada x_{k,λ^*}

1. Aplicar p passos da LDB na matriz A com vetor inicial b e formar a matriz B_p.

2. Definir $k = p$. Calcular o ponto-fixo $\lambda^{(k)^*}$ de $\phi_1^{(k)}$ e definir $\lambda_0 = \lambda^{(k)^*}$, $\lambda_{\text{old}} = \lambda_0$, $k \leftarrow k + 1$.

3. Realizar mais um passo da LDB e calcular o ponto-fixo $\lambda^{(k)^*}$ de $\phi_1^{(k)}$ tomando λ_0 como valor inicial.
Definir $\lambda_{\text{old}} = \lambda_0$, $\lambda_0 = \lambda^{(k)^*}$.

4. **Se** (critério de parada satisfeito) **faça**
$\lambda^* = \lambda_{\text{old}}$
Senão faça
$k \leftarrow k + 1$
Vá para passo **3.**
Fim se

5. Calcular solução regularizada x_{k,λ^*}

Tabela 4.1: Esquema do algoritmo Lanc-FP

Escolhemos como critério de parada quando a mudança relativa entre dois pontos-fixos consecutivos se torna pequena, ou seja,

$$\frac{|\lambda^{(k+1)^*} - \lambda^{(k)^*}|}{|\lambda^{(k)^*}|} < \epsilon_1, \tag{4.8}$$

em que ϵ_1 é um pequeno parâmetro de tolerância. Uma desvantagem do critério de parada (4.8) é que pode tornar o método computacionalmente caro quando a sequência $\lambda^{(k)^*}$ decresce muito devagar. Para contornar esta dificuldade introduzimos um outro critério de parada definido por

$$\frac{|\lambda^{(k+1)^*} - \lambda^{(k)^*}|}{|\lambda^{(0)^*}|} < \epsilon_2, \tag{4.9}$$

em $\lambda^{(0)^*}$ é o ponto-fixo calculado no passo 2 e ϵ_2 um outro parâmetro de tolerância. Este critério de parada é essencialmente o mesmo critério usado por W-GCV em [14]. O

critério de parada usado por Lanc-FP para parar as iterações será aquele que for satisfeito primeiro, ou seja, o algoritmo irá parar as iterações se (4.8) ou (4.9) for satisfeito.

4.1 Análise de convergência

Veremos agora alguns resultados teóricos que suportam nosso algoritmo. O teorema a seguir dá uma condição suficiente para existência de ponto-fixo.

Teorema 4.1. *Assuma $\beta_{k+1} \neq 0$. Uma condição suficiente para que $\phi_1^{(k)}(\lambda)$ tenha um ponto-fixo convexo é que $\dfrac{\left(-\phi_1^{(k)}(\lambda)\right)^2}{\lambda^2} = |m_L^{(k)}(\lambda)| < 1$ em algum intervalo $I \subset (0, \sigma_1(B_k))$.*

Demonstração. Introduza a função $h : \mathbb{R}^+ \to \mathbb{R}$ definida por $h(\lambda) = \|r_{k,\lambda}\|_2^2 - \lambda^2 \|y_{k,\lambda}\|_2^2$. Está claro que $h(\lambda)$ é contínua em $\mathbb{R}^+$ e que $\bar{\lambda}$ é um ponto-fixo de $\phi_1^{(k)}(\lambda)$ se, e somente se, $\bar{\lambda}$ é um zero de $h(\lambda)$. Note que a condição $\beta_{k+1} \neq 0$ implica $\|r_{k,\lambda}\|_2^2 \to \delta_0^{(k)} > 0$ quando $\lambda \to 0^+$, e portanto $h(\lambda) \to \delta_0^{(k)} > 0$ quando $\lambda \to 0^+$. Notemos que, pelo lema 2.1 obtemos

$$\left[\phi_1^{(k)}(\lambda)\right]^2 > \lambda^2 \quad \text{para} \quad \lambda > \omega_1,$$

o que implica $h(\lambda) > 0$ para $\lambda > \omega_1$.

□

O próximo teorema caracteriza as funções $\phi_1^{(k)}(\lambda)$.

Teorema 4.2. *Para todo $\lambda > 0$ vale*

$$\phi_1^{(k+1)}(\lambda) \leqslant \phi_1^{(k)}(\lambda),$$

e também

$$\phi_1(\lambda) \leqslant \phi_1^{(k)}(\lambda), \quad k = 1, \ldots.$$

Demonstração. Suponha que exista um k_0 e um λ_0 tal que

$$\phi_1^{(k_0)}(\lambda_0) < \phi_1^{(k_0+1)}(\lambda_0)$$

pela definição de $\phi_1^{(k)}(\lambda)$ temos

$$\frac{\|r_{k_0,\lambda_0}\|_2}{\|x_{k_0,\lambda_0}\|_2} < \frac{\|r_{k_0+1,\lambda_0}\|_2}{\|x_{k_0+1,\lambda_0}\|_2} \leq \frac{\|r_{k_0,\lambda_0}\|_2}{\|x_{k_0+1,\lambda_0}\|_2}$$

a última desigualdade vale por (3.34). Então temos

$$\frac{1}{\|x_{k_0,\lambda_0}\|_2} < \frac{1}{\|x_{k_0+1,\lambda_0}\|_2} \Rightarrow \|x_{k_0+1,\lambda_0}\|_2 < \|x_{k_0,\lambda_0}\|_2$$

que é uma contradição, veja (3.35).

Se executarmos $k = n$ passos da bidiagonalização de Lanczos, temos $\phi_1(\lambda) = \phi_1^{(n)}(\lambda)$, portanto $\phi_1(\lambda) \leqslant \phi_1^{(k)}(\lambda), \quad k = 1, \ldots.$

□

Uma ilustração do teorema acima segue na figura 4.1. Na esquerda temos a função $\phi_1(\lambda)$ para o problema *heat* com dimensão $n = 256$ e erro relativo nos dados de 1% estando por baixo de todas as demais curvas, e junto as funções $\phi_1^{(3)}(\lambda)$, $\phi_1^{(4)}(\lambda)$, $\phi_1^{(5)}(\lambda)$ e $\phi_1^{(6)}(\lambda)$. Na direita temos um zoom da figura da esquerda para mostrar com mais detalhes o comportamento das curvas $\phi_1^{(k)}(\lambda)$, $k = 3, 4, 5, 6$.

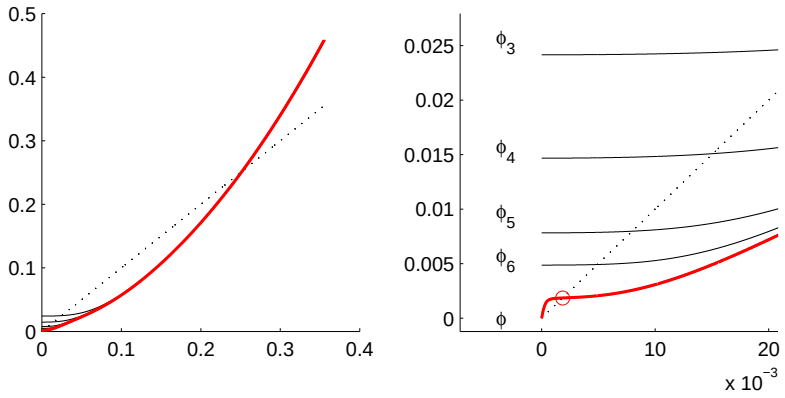

Figura 4.1: Funções $\phi_1(\lambda)$,$\phi_1^{(3)}(\lambda)$, $\phi_1^{(4)}(\lambda)$, $\phi_1^{(5)}(\lambda)$ e $\phi_1^{(6)}(\lambda)$.

Com esse teorema podemos garantir que a sequência de pontos fixos obtidas em cada iteração não irá divergir. Este é o assunto do próximo teorema.

Teorema 4.3. *A sequência de pontos-fixos $\lambda^{(k)^*}$ é não-crescente.*

Demonstração. Seja $\lambda^{(k)^*}$ e $\lambda^{(k+1)^*}$ os pontos fixos de $\phi_1^{(k)}(\lambda)$ e $\phi_1^{(k+1)}(\lambda)$ respectivamente, do teorema 4.2 segue que

$$\lambda^{(k+1)^*} = \phi_1^{(k+1)}(\lambda^{(k+1)^*}) \leqslant \phi_1^{(k)}(\lambda^{(k+1)^*}).$$

Aplicando $\phi_1^{(k)}(\lambda)$ em ambos os lados

$$\phi_1^{(k)}(\lambda^{(k+1)^*}) \leqslant \phi_1^{(k)}(\phi_1^{(k)}(\lambda^{(k+1)^*})).$$

Logo

$$\lambda^{(k+1)^*} \leqslant \phi_1^{(k)}(\phi_1^{(k)}(\lambda^{(k+1)^*})).$$

Aplicando denovo $\phi_1^{(k)}(\lambda)$ em ambos os lados

$$\phi_1^{(k)}(\lambda^{(k+1)^*}) \leqslant \phi_1^{(k)}(\phi_1^{(k)}(\phi_1^{(k)}(\lambda^{(k+1)^*}))).$$

Logo,

$$\lambda^{(k+1)^*} \leqslant \phi_1^{(k)}(\phi_1^{(k)}(\phi_1^{(k)}(\lambda^{(k+1)^*}))).$$

Se continuarmos aplicando $\phi_1^{(k)}(\lambda)$ a sequência irá convergir para $\lambda^{(k)^*}$, i.e.,

$$\lambda^{(k+1)^*} \leqslant \lambda^{(k)^*}.$$

□

O próximo teorema é um dos resultados mais relevantes deste trabalho, pois com ele conseguimos estabilizar o método LSQR de tal maneira que, a partir de certo ponto não existe mudança na solução calculada.

Teorema 4.4. *Assuma que o maior ponto-fixo convexo de $\phi_1(\lambda)$ seja encontrado em k passos da bidiagonalização de Lanczos. isto é, $\lambda^* = \phi_1^{(k)}(\lambda^*)$. Então $x_{k,\lambda^*} = x_{k+1,\lambda^*} = \cdots = x_{n,\lambda^*} \triangleq x_{\lambda^*}$ e como consequência, a norma do erro $\|x^{\text{exato}} - x_{j,\lambda^*}\|_2$ permanece constante para $k \leqslant j \leqslant n$.*

Demonstração. Notemos que de acordo com o algoritmo LSQR temos, no passo k, $x_{k,\lambda^*} =$

$V_k y_{k,\lambda^*}$, e que y_{k,λ^*} é solução do sistema $R_k y = b_k$, com R_k e b_k oriundos da transformação

$$Q_k \left[\begin{array}{c|c} B_k & \beta_1 e_1 \\ \lambda^* I_k & 0 \end{array}\right] = \left[\begin{array}{c|c} R_k & b_k \\ 0 & \bar{\varphi}_{k+1} \\ 0 & c_k \end{array}\right], \quad b_k = \begin{bmatrix} \varphi_1 \\ \vdots \\ \varphi_k \end{bmatrix}, \quad c_k = \begin{bmatrix} \psi_1 \\ \vdots \\ \psi_k \end{bmatrix} \tag{4.10}$$

em que $Q_k \in \mathbb{R}^{(2k+1)\times(2k+1)}$ é um produto de rotações de Givens. É desnecessário mencionar que apenas três novas quantidades são calculadas no passo $k+1$: φ_{k+1} (substituindo $\bar{\varphi}_{k+1}$), $\bar{\varphi}_{k+2}$, e ψ_{k+1}; essas quantidades em b_k e c_k permanecem inalteradas e formam parte de b_{k+1} e c_{k+1}, respectivamente. Também mencionamos que x_{k+1,λ^*} pode ser obtido atualizando a fórmula

$$x_{k+1,\lambda^*} = x_{k,\lambda^*} + \varphi_{k+1} d_{k+1}, \tag{4.11}$$

em que d_{k+1} é a $(k+1)$-ésima coluna de $D_{k+1} = V_{k+1} R_{k+1}^{-1}$. Devemos provar que $\varphi_{k+1} = 0$. De fato, como λ^* é um ponto-fixo de $\phi_1^{(j)}(\lambda)$ para $j = k, \ldots, n$, temos

$$\|\beta_1 e_1 - B_k y_{k,\lambda^*}\|_2 = \lambda^* \|x_{k,\lambda^*}\|_2, \quad \text{e} \quad \|\beta_1 e_1 - B_{k+1} y_{k+1,\lambda^*}\|_2 = \lambda^* \|x_{k+1,\lambda^*}\|_2, \tag{4.12}$$

então

$$\|x_{k+1,\lambda^*}\|_2 \leqslant \|x_{k,\lambda^*}\|_2,$$

em que usamos o fato de que $\|\beta_1 e_1 - B_{k+1} y_{k+1,\lambda^*}\| \leqslant \|\beta_1 e_1 - B_k y_{k,\lambda^*}\|$. Mas como a norma da solução forma uma sequência não-crescente, segue que $\|x_{k+1,\lambda^*}\|_2 = \|x_{k,\lambda^*}\|_2$, e substituindo este resultado em (4.12) nos leva a

$$\|\beta_1 e_1 - B_k y_{k,\lambda^*}\|_2 = \|\beta_1 e_1 - B_{k+1} y_{k+1,\lambda^*}\|_2. \tag{4.13}$$

Usando (4.10) segue que

$$\|\beta_1 e_1 - B_k y_{k,\lambda^*}\|_2^2 = |\bar{\varphi}_{k+1}|^2 + \|c_k\|_2^2 - \lambda^* \|x_{k,\lambda^*}\|^2,$$

e

$$\|\beta_1 e_1 - B_{k+1} y_{k+1,\lambda^*}\|_2^2 = |\bar{\varphi}_{k+2}|^2 + \|c_{k+1}\|_2^2 - \lambda^* \|x_{k+1,\lambda^*}\|^2.$$

Substituindo esses resultados em (4.13) temos

$$|\bar{\varphi}_{k+1}|^2 + \|c_k\|_2^2 = |\bar{\varphi}_{k+2}|^2 + \|c_{k+1}\|_2^2. \tag{4.14}$$

Mas como a norma-2 de

$$Q_k \begin{bmatrix} \beta_1 e_1 \\ 0 \end{bmatrix} = \begin{bmatrix} b_k \\ \bar{\varphi}_{k+1} \\ c_k \end{bmatrix}, \tag{4.15}$$

em (4.10) é igual a β_1 e não depende de k, segue então

$$|\varphi_1|^2 + \cdots + |\varphi_k|^2 + |\bar{\varphi}_{k+1}|^2 + \|c_k\|_2^2 = (|\varphi_1|^2 + \cdots + |\varphi_k|^2) + |\varphi_{k+1}|^2 + |\bar{\varphi}_{k+2}|^2 + \|c_{k+1}\|_2^2.$$

Usando (4.14) nesta última equação concluímos que $\varphi_{k+1} = 0$ como desejado.

□

Na figura 4.2 podemos verificar o teorema 4.4. Utilizamos o problema *heat* com dimensão $n = 1200$ e erro relativo nos dados de 2%. A execução do algoritmo lanc-fp terminou na iteração $k = 20$, por este motivo calculamos as soluções $x_{k,\lambda}$ até $k = 40$. Constatamos que após a iteração $k = 20$ tanto a norma da solução aproximada $x_{k,\lambda}$ como o erro relativo se mantêm constante. Em ambos os gráficos a cruz (×) indica a iteração de parada.

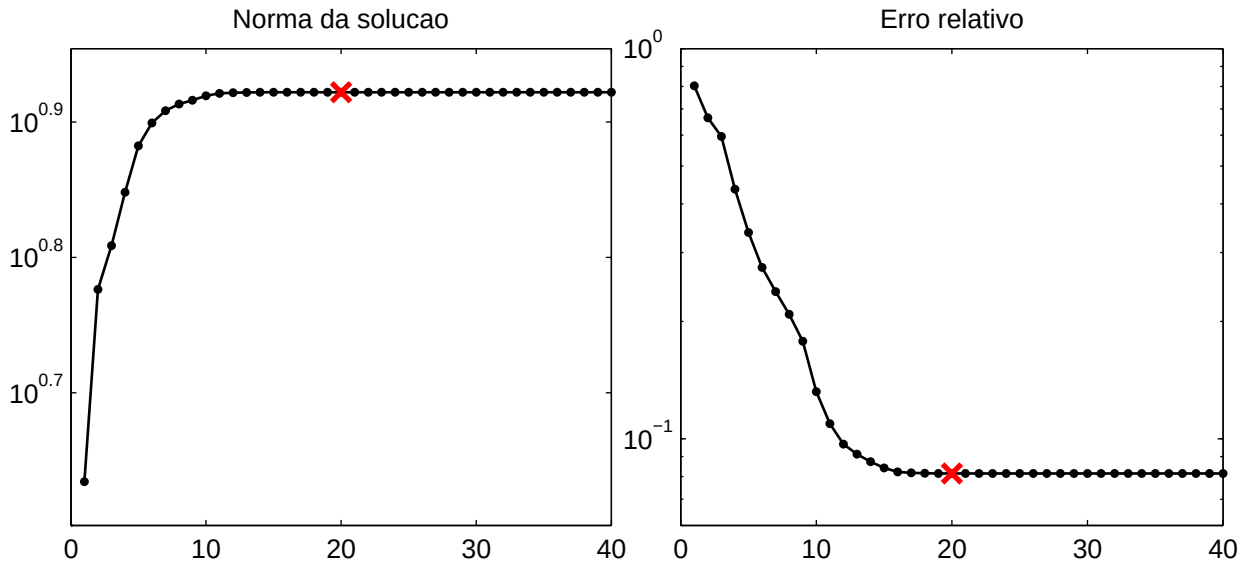

Figura 4.2: Ilustração do teorema 4.4. A cruz (×) indica a iteração de parada.

Analisando a figura constatamos que após a iteração $k = 20$ (marcado com um $\times$), ou seja, após o algoritmo atingir o critério de parada, a norma das soluções $x_{k,\lambda}$ para $k = 20, \ldots, 40$ não sofreram alterações, o mesmo ocorrendo com o erro relativo nas soluções.

Capítulo 5

Resultados Numéricos

Para ilustrar nossa proposta aplicamos o algoritmo a diversos problemas testes encontrados na toolbox RegularizationTools de Hansen [31] e problemas de restauração de imagens, um deles da toolbox RestoreTools [42].

5.1 Equações Integrais

Para avaliar nossa proposta escolhemos, dentre vários, os problemas *foxgood, heat, wing, shaw, baart, deriv2, gravity* e *phillips*. Veremos uma sucinta descrição de cada um deles.

foxgood

Este é um problema modelo que não satisfaz a condição discreta de Picard para valores singulares pequenos. Este problema foi usado inicialmente por Fox e Goodwin e é considerado severamente mal-posto. Mais informações ver em [1].

Podemos conferir na figura 5.1 que a condição discreta de Picard, mesmo para o problema exato, não é satisfeita para os últimos valores singulares. Os círculos ($\circ$) representam os coeficientes $|u_i^T b|/\sigma_i$, as cruzes ($\times$) representam os coeficientes de Fourier $|u_i^T b|$ e os pontos ($-\bullet-$) representam os valores singulares.

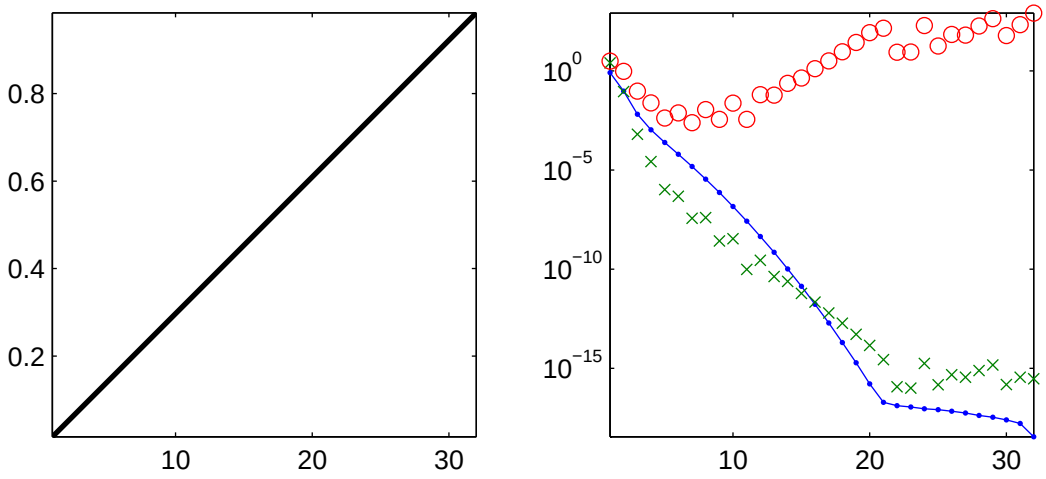

Figura 5.1: Problema: *foxgood*. Solução exata e Condição Discreta de Picard.

heat

Uma equação integral de primeira espécie com $[0, 1]$ sendo o intevalo de integração. O núcleo é $K(x, y) = k(x - y)$ em que

$$k(y) = \frac{y^{-\frac{3}{2}}}{2\kappa\sqrt{\pi}} e^{-\frac{1}{4\kappa^2 y}} \tag{5.1}$$

e κ controla o mal-condicionamento da matriz A do problema discretizado. Mais informações podem ser encontradas em [13, 18].

Na figura 5.2 temos a solução exata na esquerda e na direita os coeficientes para a inspeção visual da condição discreta de Picard. Os círculos ($\circ$) representam os coeficientes $|u_i^T b|/\sigma_i$, as cruzes ($\times$) representam os coeficientes de Fourier $|u_i^T b|$ e os pontos ($-\bullet-$) representam os valores singulares.

wing

Discretização de uma equação integral de Fredholm de primeira espécie com núcleo K e função g dados por

$$K(x, y) = ye^{-xy^2} \tag{5.2}$$

$$g(x) = \frac{e^{-xt_1^2} - e^{-xt_2^2}}{2x} \tag{5.3}$$

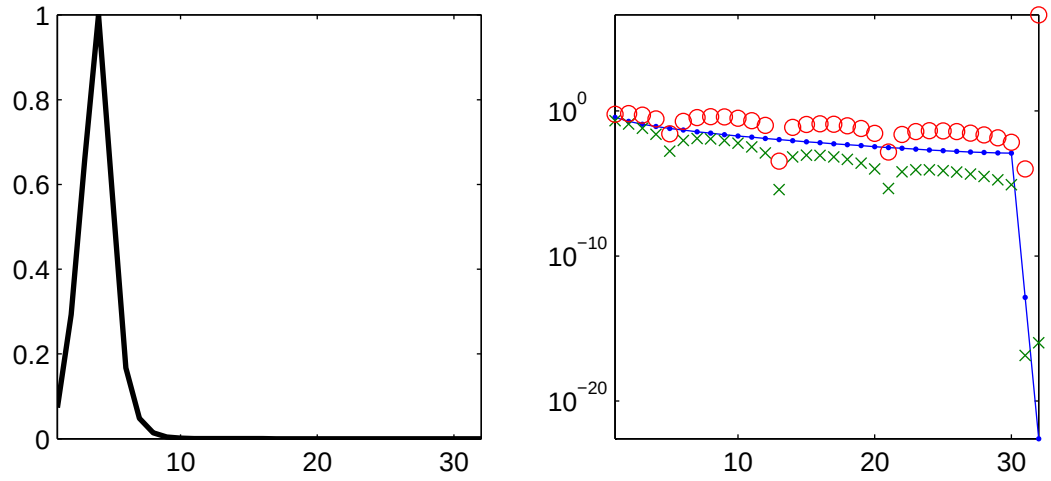

Figura 5.2: Problema: *heat*. Solução exata e Condição Discreta de Picard.

e as constantes t_1 e t_2 satisfazem $t_1 < t_2$. O intervalo de integração é $[0, 1]$. A solução f é dada por

$$f(y) = \begin{cases} 1 \text{ para } t_1 < y < t_2 \\ 0 \text{ para } y \notin (t_1, t_2) \end{cases} \tag{5.4}$$

Na figura 5.3 temos a solução exata, que neste caso é descontínua, na esquerda e na direita os coeficientes para a inspeção visual da condição discreta de Picard. Os círculos ($\circ$) representam os coeficientes $|u_i^T b|/\sigma_i$, as cruzes ($\times$) representam os coeficientes de Fourier $|u_i^T b|$ e os pontos ($-\bullet-$) representam os valores singulares.

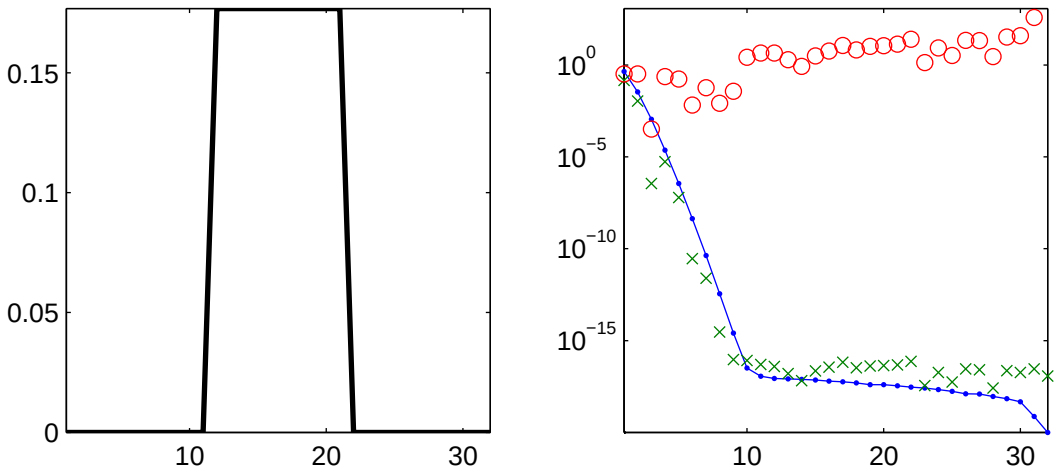

Figura 5.3: Problema: *wing*. Solução exata e Condição Discreta de Picard.

shaw

Modelo de restauração de imagem unidimensional. O sistema linear vem da discretização de uma equação integral de Fredholm de primeira espécie com $[-\pi/2, \pi/2]$ sendo o intervalo de integração. O núcleo K e a solução f são dados por

$$K(x,y) = (\cos(x) + \cos(y)) \left(\frac{\operatorname{sen}(u)}{u} \right)^2 \tag{5.5}$$

$$f(y) = a_1 e^{-c_1(y-t_1)^2} + a_2 e^{-c_2(y-t_2)^2} \tag{5.6}$$

em que $u = \pi(\operatorname{sen}(x) + \operatorname{sen}(y))$ e as constantes definidas por $a_1 = 2$, $c_1 = 6$, $t_1 = 0,8$, $a_2 = 1$, $c_2 = 2$ e $t_2 = -0,5$.

Na figura 5.4 temos a solução exata na esquerda e na direita os coeficientes para a inspeção visual da condição discreta de Picard. Este problema teste é bastante utilizado na literatura para avaliação de algoritmos. Os círculos ($\circ$) representam os coeficientes $|u_i^T b|/\sigma_i$, as cruzes ($\times$) representam os coeficientes de Fourier $|u_i^T b|$ e os pontos ($-\bullet-$) representam os valores singulares.

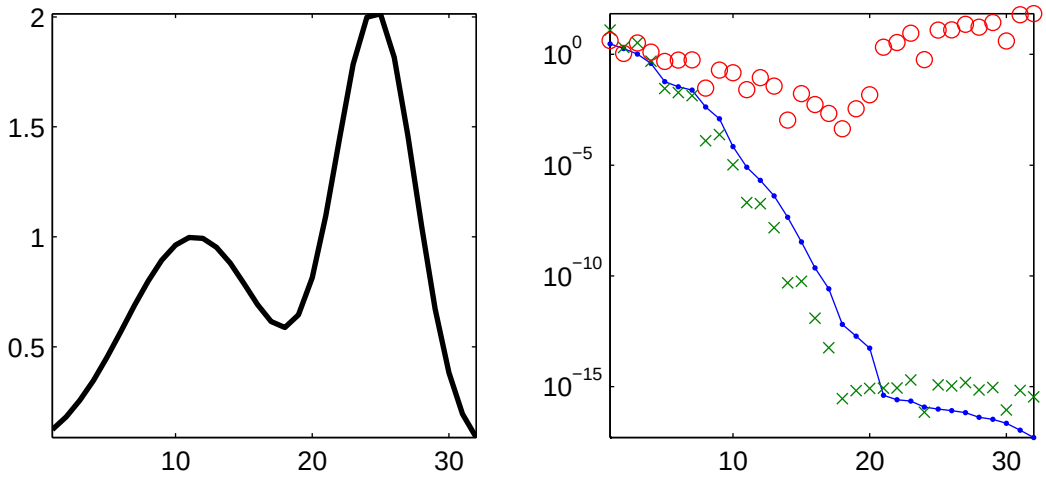

Figura 5.4: Problema: *shaw*. Solução exata e Condição Discreta de Picard.

baart

Outra equação de Fredholm de primeira espécie em que o núcleo K e a função g são dados por

$$K(x, y) = e^{x\cos(y)} \tag{5.7}$$

$$g(x) = 2\frac{\text{senh}(x)}{x} \tag{5.8}$$

com $x \in [0, \pi/2]$ e $y \in [0, \pi]$, e a função f dada por

$$f(y) = \text{sen}\,(y) \tag{5.9}$$

Na figura 5.5 temos a solução exata na esquerda e na direita os coeficientes para a inspeção visual da condição discreta de Picard. A condição discreta de Picard para este problema é satisfeita. Os círculos ($\circ$) representam os coeficientes $|u_i^T b|/\sigma_i$, as cruzes ($\times$) representam os coeficientes de Fourier $|u_i^T b|$ e os pontos ($-\bullet-$) representam os valores singulares.

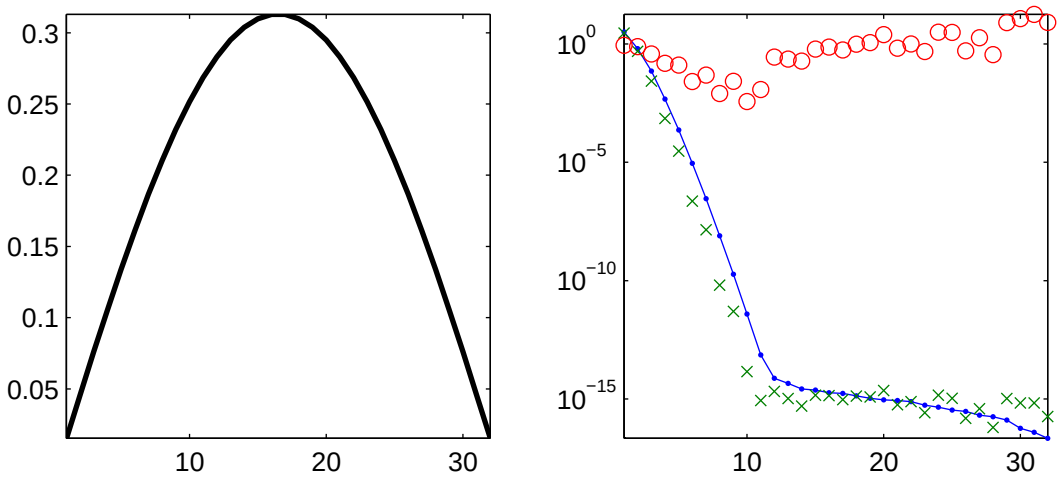

Figura 5.5: Problema: *baart*. Solução exata e Condição Discreta de Picard.

deriv2

O núcleo K desta equação integral de Fredholm de primeira espécie é a função de Green para a segunda derivada

$$K(x,y) = \begin{cases} x(y-1), & x < y \\ y(x-1), & x \geqslant y \end{cases} \tag{5.10}$$

e o intervalo de integração é $[0,1]$. Neste problema dispomos de três exemplos para a função g e a solução f, são eles:

$$\begin{aligned}
&\text{exemplo 1}: \quad g(x) = (x^3 - x)/6, && f(y) = y \\
&\text{exemplo 2}: \quad g(x) = e^x + (1-e)x - 1, && f(y) = e^y \\
&\text{exemplo 3}: \quad g(x) = \begin{cases} (4x^3 - 3x)/24, & x < 0,5 \\ (-4x^3 + 12x^2 - 9x + 1)/24, & x \geqslant 0,5 \end{cases} \\
&\qquad\qquad f(y) = \begin{cases} y, & y < 0,5 \\ 1-y, & y \geqslant 0,5 \end{cases}
\end{aligned}$$

Na figura 5.6 temos a solução exata na esquerda e na direita os coeficientes para a inspeção visual da condição discreta de Picard. Aqui escolhemos o exemplo 3 por apresentar um ponto sem derivada. Os círculos ($\circ$) representam os coeficientes $|u_i^T b|/\sigma_i$, as cruzes ($\times$) representam os coeficientes de Fourier $|u_i^T b|$ e os pontos ($-\bullet-$) representam os valores singulares.

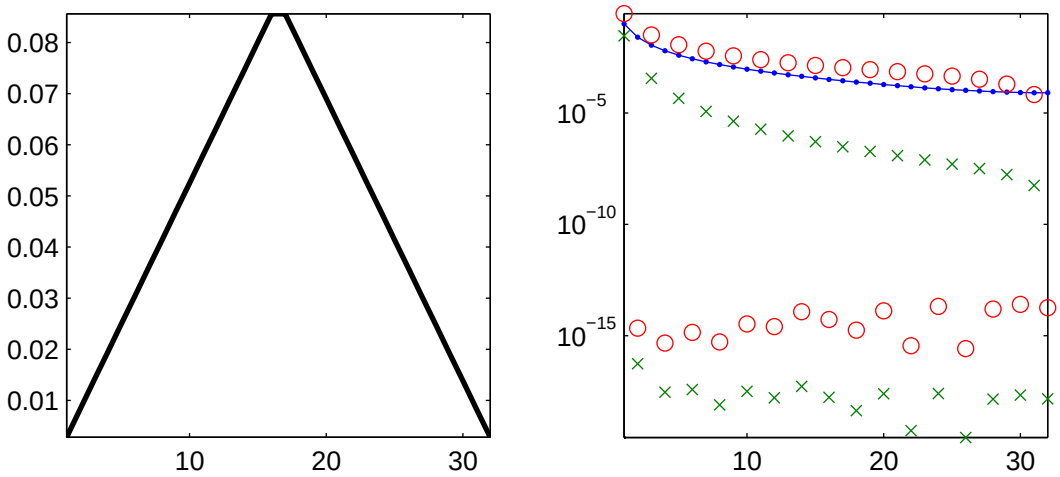

Figura 5.6: Problema: *deriv2*. Solução exata e Condição Discreta de Picard.

gravity

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

$$K(x, y) = d(d^2 + (x - y)^2)^{-3/2} \tag{5.11}$$

e função f é dada por

$$f(y) = \text{sen}\,(\pi y) + \frac{1}{2}\text{sen}\,(2\pi y), \tag{5.12}$$

em que a constante d representa a profundidade no 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 5.7 temos a solução exata para o exemplo em que a solução é uma função contínua por partes, porém a equação (5.12) representa o exemplo 1 que neste caso é uma curva suave. Optamos por escolher o exemplo 2 por este apresentar alguns pontos sem derivada, os demais exemplos não apresentam esta característica. Os círculos ($\circ$) representam os coeficientes $|u_i^T b|/\sigma_i$, as cruzes ($\times$) representam os coeficientes de Fourier $|u_i^T b|$ e os pontos ($-\bullet-$) representam os valores singulares.

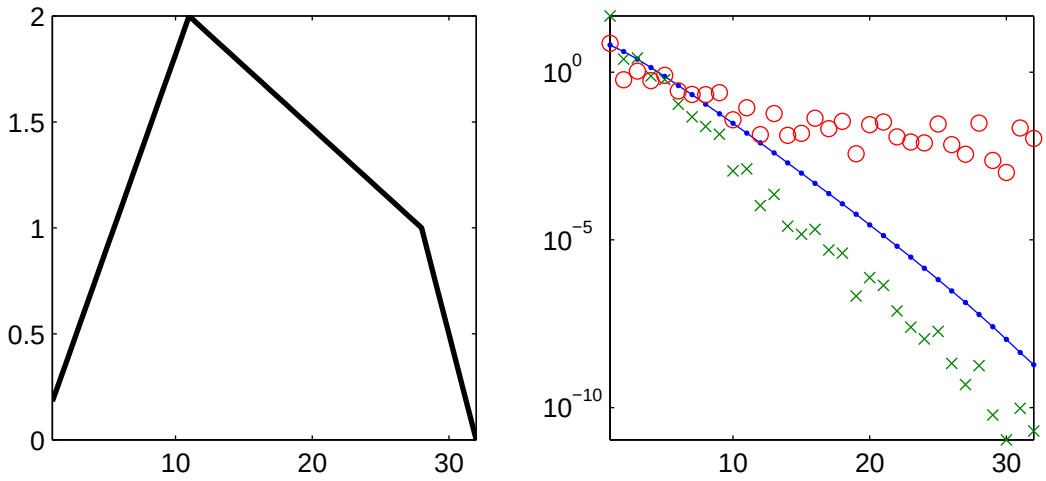

Figura 5.7: Problema: *gravity*. Solução exata e Condição Discreta de Picard.

phillips

Esta é a equação que Phillips [43] estudou. Vamos definir a função

$$\zeta(p) = \begin{cases} 1 + \cos\left(p\frac{\pi}{3}\right), & |p| < 3 \\ 0, & |p| \geqslant 3 \end{cases} \tag{5.13}$$

Então o núcleo K, a solução f e a função g são dados por

$$\begin{aligned} K(x,y) &= \zeta(x-y) \\ f(y) &= \zeta(y) \\ g(x) &= (6-|x|)\left(1+\frac{1}{2}\cos\left(x\frac{\pi}{3}\right)\right) + \frac{9}{2\pi}\text{sen}\left(|x|\frac{\pi}{3}\right) \end{aligned} \tag{5.14}$$

em que $[-6, 6]$ é o intervalo de integração.

Na figura 5.8 temos a solução exata na esquerda e na direita os coeficientes para a inspeção visual da condição discreta de Picard. O problema *phillips* é um dos problemas bem famosos na literatura. Os círculos ($\circ$) representam os coeficientes $|u_i^T b|/\sigma_i$, as cruzes ($\times$) representam os coeficientes de Fourier $|u_i^T b|$ e os pontos ($-\bullet-$) representam os valores singulares.

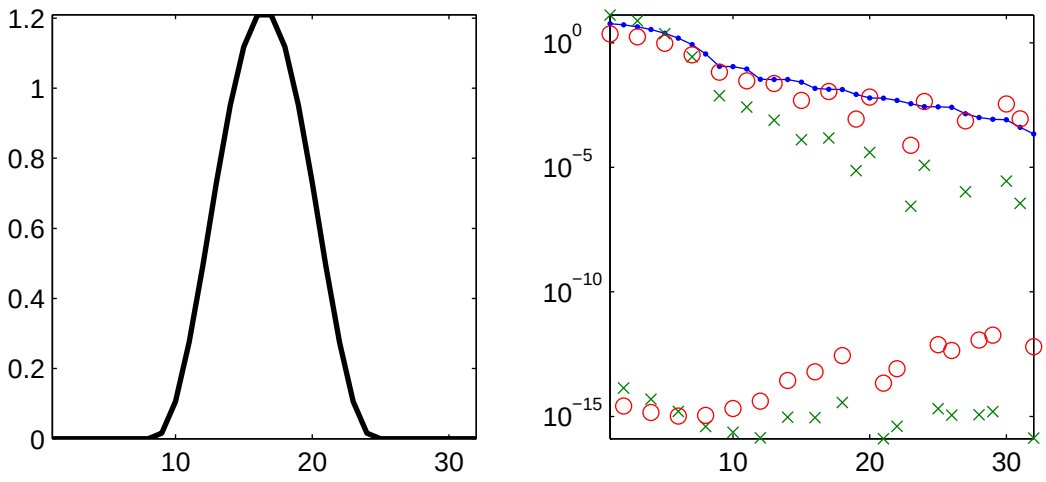

Figura 5.8: Problema: *phillips*. Solução exata e Condição Discreta de Picard.

Com base nos problemas acima descritos, utilizamos erros relativos de diversas intensidades, ou seja, $\|b^{\text{exato}} - b\|_2/\|b^{\text{exato}}\|_2 = t$, em que $t = 0,01$, $t = 0,025$, $t = 0,05$, o que

representa, respectivamente, 1%, 2,5% e 5%, neste caso, $b = b^{\text{exato}} + e$ com e contendo números com distribuição Gaussiana de média zero gerados pela rotina *randn*.

Todos os testes foram realizados com os mesmos níveis de erro com seed=0 e a dimensão do sistema $n = 4096$. Cada problema foi executado 500 vezes para que possamos obter médias para algumas quantidades. Para os problemas em que existe a possibilidade de fornecer parâmetros adicionais como dados de entrada, como *wing* por exemplo, optamos pelos valores padrões de cada problema teste e nas rotinas *deriv*2 e *gravity* utilizamos os exemplos 3 e 2 respectivamente.

O valor da variável ϵ foi definido como 10^{-4} e começamos com $p = 5$. Para os testes definimos o chute inicial como 10^{-4}.

Símbolo	Descrição
$\bar{\lambda}$	Média dos parâmetros encontrados em todas as execuções.
$\lambda_{M,m}$	Parâmetro máximo (resp. mínimo).
$\bar{E}$	Média dos erros relativos $\|x_\lambda - x_{\text{exato}}\|_2 / \|x_{\text{exato}}\|_2$.
$E_{M,m}$	Erro relativo máximo (resp. mínimo).
$\text{STD}_{\lambda,E}$	Desvio padrão dos parâmetros encontrados e dos erros relativos nas soluções, respectivamente.
$k_{M,m}$	Dimensão máxima (resp. mínimo) do subespaco projetado.
$Iter_{M,m}$	Número máximo (resp. mínimo) de avaliações da função $\phi(\lambda)$.

Tabela 5.1: Legenda para os dados colhidos nos testes numéricos realizados.

Vejamos os resultados obtidos quando temos 1% de erro relativo nos dados, isto é,

$$\frac{\|b - b^{\text{exato}}\|_2}{\|b^{\text{exato}}\|_2} = \frac{\|e\|_2}{\|b^{\text{exato}}\|_2} = 0,01.$$

Analisando a tabela 5.2 podemos perceber que o algoritmo proposto tem uma importante característica herdada do algoritmo de ponto-fixo devido a Bazán [2], ou seja, ele é consistente em encontrar o parâmetro de regularização independente de como o erro nos dados está distribuído.

A dimensão do espaço projetado (k_M) foi um resultado interessante uma vez que nenhum dos problemas precisou mais do que 0,6% da dimensão original do problema ($n = 4096$) para atingir um resultado suficientemente bom.

Vejamos o que acontece quando existe um pouco mais de erro no dados, ou seja, quando

Erro: 1%	*foxgood*	*heat*	*wing*	*shaw*	*baart*	*deriv2*	*gravity*	*phillips*
$\bar{\lambda}$	0,0078	0,0019	0,0032	0,0236	0,0238	0,0010	0,0633	0,0509
λ_M	0,0078	0,0019	0,0032	0,0236	0,0238	0,0010	0,0633	0,0511
λ_m	0,0077	0,0019	0,0032	0,0235	0,0236	0,0010	0,0631	0,0507
STD_λ	3,7e-6	2,7e-6	3,2e-6	1,6e-5	3,2e-5	6,7e-7	3,8e-5	7,9e-5
$\bar{E}$	0,0200	0,0524	0,6026	0,0760	0,1657	0,0279	0,0297	0,0228
E_M	0,0501	0,0694	0,6028	0,0944	0,1733	0,0456	0,0505	0,0395
E_m	0,0051	0,0364	0,6024	0,0594	0,1597	0,0162	0,0157	0,0089
STD_E	0,0071	0,0056	7,0e-5	0,0060	0,0023	0,0050	0,0057	0,0040
k_M	7	21	7	8	7	10	10	12
k_m	7	19	7	8	7	7	8	7
$Iter_M$	5	41	5	7	6	10	10	14
$Iter_m$	4	36	5	6	5	4	6	4

Tabela 5.2: Resultados obtidos após 500 execuções e 1% de erro relativo.

temos $\|e\|/\|b^{\text{exato}}\|_2 = 0,025$. Os dados obtidos nas 500 execuções podem ser conferidos na tabela 5.3.

Assim como no caso de 1% de erro relativo nos dados, para o caso de 2,5% temos as mesmas características, ou seja, o desvio padrão no parâmetro de regularização continua pequeno, da ordem de 10^{-4}, 10^{-5}. As dimensões dos espaços projetados diminuiram praticamente para todos os problemas, a mais notória redução foi para o problema *heat* que de 19-21 (min-max) foi para 15-16 (min-max).

Erro: 2.5%	*foxgood*	*heat*	*wing*	*shaw*	*baart*	*deriv2*	*gravity*	*phillips*
$\bar{\lambda}$	0,0195	0,0049	0,0083	0,0595	0,0617	0,0025	0,1585	0,1275
λ_M	0,0195	0,0049	0,0083	0,0596	0,0618	0,0025	0,1587	0,1278
λ_m	0,0194	0,0049	0,0082	0,0593	0,0614	0,0025	0,1579	0,1271
STD_λ	1,8e-5	1,0e-5	2,0e-5	6,0e-5	7,2e-5	1,8e-6	1,2e-4	1,1e-4
$\bar{E}$	0,0310	0,1045	0,6035	0,1294	0,2098	0,0310	0,0434	0,0238
E_M	0,0503	0,1217	0,6040	0,1475	0,2229	0,0503	0,0594	0,0400
E_m	0,0162	0,0848	0,6031	0,1110	0,1924	0,0170	0,0299	0,0101
STD_E	0,0045	0,0061	0,0002	0,0058	0,0046	0,0056	0,0051	0,0032
k_M	7	16	7	7	7	7	9	11
k_m	7	15	7	7	7	7	7	7
$Iter_M$	6	33	7	5	7	5	8	12
$Iter_m$	5	29	6	5	6	4	4	4

Tabela 5.3: Resultados obtidos após 500 execuções e 2,5% de erro relativo.

Finalmente a última tabela com erros de 5% de erros relativos nos dados pode ser conferida na tabela 5.4 a seguir.

Erro: 5%	*foxgood*	*heat*	*wing*	*shaw*	*baart*	*deriv2*	*gravity*	*phillips*
$\bar{\lambda}$	0,0396	0,0105	0,0191	0,1203	0,1273	0,0051	0,3187	0,2562
λ_M	0,0397	0,0106	0,0193	0,1207	0,1282	0,0051	0,3194	0,2569
λ_m	0,0394	0,0104	0,0188	0,1198	0,1264	0,0050	0,3172	0,2551
STD_λ	5,6e-5	3,2e-5	7,4e-5	1,9e-4	2,9e-4	5,1e-6	3,5e-4	3,0e-4
$\bar{E}$	0,0530	0,1875	0,6179	0,1569	0,2843	0,0430	0,0538	0,0268
E_M	0,0670	0,2087	0,6217	0,1699	0,2970	0,0634	0,0696	0,0427
E_m	0,0395	0,1696	0,6148	0,1448	0,2661	0,0254	0,0413	0,0136
STD_E	0,0045	0,0060	0,0011	0,0039	0,0049	0,0059	0,0042	0,0032
k_M	7	13	7	7	7	7	8	10
k_m	7	12	7	7	7	7	7	7
$Iter_M$	6	30	11	6	7	5	7	11
$Iter_m$	6	25	10	5	6	5	4	4

Tabela 5.4: Resultados obtidos após 500 execuções e 5% de erro relativo.

Novamente, o que já era esperado neste ponto, o desvio padrão foi pequeno e para alguns problemas a dimensão k do espaço projetado teve novas reduções.

De modo a comparar nossa proposta com algumas das técnicas descritas no capítulo 1, faremos uma avaliação com alguns dos problemas testes descritos utilizando erros de 1% e dimensão $n = 1024$ de modo que assim possamos calcular a SVD das matrizes para cada problema e aplicar a curva-L, a GCV, o FP e a discrepância.

Nas tabelas 5.5,5.6 e 5.7 temos os resultados obtidos com apenas uma execução para os problemas *foxgood, heat* e *shaw* e para o princípio da discrepância usamos a norma do erro e exata de modo que fosse possível obter o melhor resultado do método.

Desta comparação podemos concluir que o nossa proposta de algoritmo além de obter resultados praticamente idênticos aos do algoritmo de Ponto-Fixo original, precisou projetar os problemas em subespaços de dimensão consideravelmente menores.

foxgood	Curva-L	GCV	FP	DP	Lanc-FP
λ	0,0052	0,0056	0,0077	0,0091	0,0077
$\|x_\lambda\|_2$	18,4793	18,4781	18,4702	18,4652	18,4702
ER (%)	1,2828	1,3304	1,7277	1,9487	1,7277
Iter	-	-	-	-	4

Tabela 5.5: Comparação entre Curva-L, GCV, FP, DP e Lanc-FP para o problema *foxgood*.

heat	Curva-L	GCV	FP	DP	Lanc-FP
λ	0,3214	0,0017	0,0019	0,0027	0,0019
$\|x_\lambda\|_2$	2,2057	7,8197	7,8114	7,7745	7,8114
ER (%)	83,5903	7,1848	7,2056	7,8547	7,2059
Iter	-	-	-	-	17

Tabela 5.6: Comparação entre Curva-L, GCV, FP, DP e Lanc-FP para o problema *heat*.

shaw	Curva-L	GCV	FP	DP	Lanc-FP
λ	0,0179	0,0092	0,0236	0,0189	0,0236
$\|x_\lambda\|_2$	31,7462	31,8778	31,6743	31,7330	31,6743
ER (%)	5,6865	4,1362	7,0446	5,9081	7,0447
Iter	-	-	-	-	7

Tabela 5.7: Comparação entre Curva-L, GCV, FP, DP e Lanc-FP para o problema *shaw*.

5.2 Restauração de Imagem

Quando usamos uma câmera gostaríamos que a imagem gravada fosse exatamente igual a cena real, contudo no momento em que a imagem deve ser gravada existem diversos fatores que podem fazer com que a imagem fique embaçada (*blurred*) devido, por exemplo, à limitação em se representar as cores num sistema computacional ou uma má qualidade das lentes que podem desfocar a imagem. Um exemplo deste fenômeo está representado na figura 5.9 que tem 256×256 pixels.

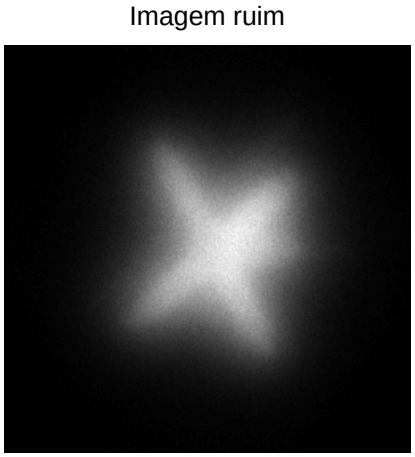

Figura 5.9: Imagem ruim de um satélite.

Uma imagem digital é composta por pontos chamados de pixels, e cada pixel representa uma variável. Uma imagem pequena normalmente tem aproximadamente $256^2 = 65.536$

pixels, ou variáveis, mas imagens não pequenas podem facilmente chegar a 5 ou 10 milhões de pixels.

Neste exemplo a imagem pode ser considerada um vetor b de dados, a imagem original e desconhecida armazenada num vetor x e o operador que criou este efeito uma matriz A, o que acarreta num sistema $Ax = b$ com 65.536 variáveis.

Vamos supor que temos uma imagem totalmente preta exceto por um ponto (pixel) no centro que é branco. No instante em que tirarmos uma foto deste único ponto o processo de embaçar a foto causará um espalhamento daquele ponto na sua vizinhança, como ilustrado na figura 5.10. Este espalhamento é chamado de *point spread function* (PSF).

Figura 5.10: Esquerda: um único ponto de luz, chamado de *point source*. Direita: o ponto de luz espalhado, chamado *point spread function*.

Em algumas situações 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) é dado por

$$p_{ij} = \begin{cases} \dfrac{1}{\pi r^2} & \text{se } (i-k)^2 + (j-l)^2 \leqslant r^2 \\ 0 & \text{caso contrário,} \end{cases} \tag{5.15}$$

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 descrito

como uma função Gaussiana bidimensional [34, 48] e os elementos da PSF são dados por

$$p_{ij} = exp\left(-\frac{1}{2}\begin{bmatrix} i-k \\ j-l \end{bmatrix}^T \begin{bmatrix} s_1^2 & \rho^2 \\ \rho^2 & s_2^2 \end{bmatrix}^{-1} \begin{bmatrix} i-k \\ j-l \end{bmatrix}\right), \tag{5.16}$$

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) da PSF P. Para outras PSF's ver [33] e as referências contidas nele.

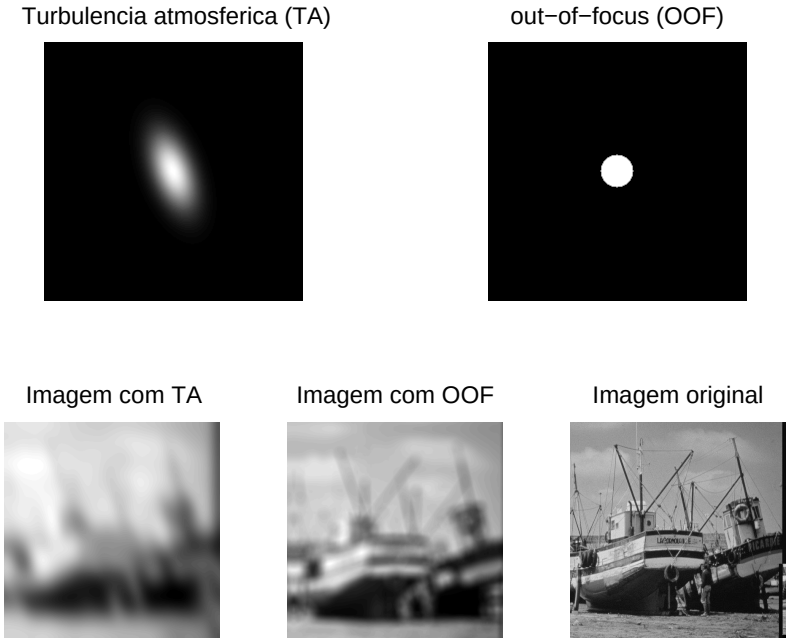

Figura 5.11: Da esquerda para a direita. Parte superior: PSF para turbulência atmosférica e *out-of-focus*. Parte inferior: Imagem embaçada com turbulência atmosférica, imagem fora-de-foco e imagem original.

Diferentes PSF's produzem diferentes matrizes e efeitos na imagem original como podemos conferir na figura 5.11 em que temos a PSF de tamanho 256×256 para a turbulência atmosférica com centro no ponto $(128, 128)$, $s_1 = 25$, $s_2 = 15$ e $\rho = 13$ e a PSF para *out-of-focus* utilizando $r = 16$. Para mais detalhes a respeito de modelagem e outras informações a respeito do efeito *blur*, ver [33, 42].

5.2.1 Astronomia

Assim como a toolbox RegularizationTools do Hansen [31], a toolbox RestoreTools têm várias rotinas para tratar problemas de restauração de imagem. Neste pacote existem alguns problemas testes bem como algoritmos para o tratamento destes problemas, um deles é um outro algoritmo chamado HyBR. Neste algoritmo o usuário pode escolher o método WGCV (Weighted-GCV) [14] como critério de parada.

Para testar nossa proposta com um problema de dimensão maior, escolhemos o problema *satellite*, que é uma imagem de 256×256 em tons de cinza, o que significa que temos um problema com 65.536 variáveis. O método HyBR pode ser usado com ou sem precondicionador.

Executamos a proposta deste trabalho 50 vezes com erros relativos de 1%, 5% e 0,1% nos dados e comparamos os resultados com os obtidos com o algortimo HyBR.

Nas tabelas 5.8, 5.9 e 5.10, os símbolos *prec* e $\perp$ significam, quando aparecem, que foi utilizado precondicionador e reortogonalização no processo de bidiagonalização e a legenda é a mesma encontrada na tabela 5.1 exceto que $Iter_{M,n}$ significa o número máximo (resp. mínimo) de avaliações da função $\phi^{(2)}(\lambda)$ pois escolhemos $p = 2$ para este teste de restauração de imagem.

Vejamos os resultados obtidos quando o erro relativo nos dados são de 1% na tabela 5.8.

Erro: 1%	Lanc-FP$^{\perp}_{prec}$	Lanc-FP$_{prec}$	Lanc-FP$^{\perp}$	Lanc-FP	HyBR$_{prec}$	HyBR
$\bar{\lambda}$	0,0053	0,0053	0,0055	0,0056	0,0188	0,0311
λ_M	0,0054	0,0054	0,0055	0,0056	0,0204	0,0312
λ_m	0,0052	0,0052	0,0055	0,0056	0,0172	0,0310
STD$_\lambda$	5,2e-5	5,1e-5	3,6e-6	6,5e-6	7,9e-4	6,4e-5
$\bar{E}$	0,2984	0,2984	0,2941	0,2963	0,2984	0,3975
E_M	0,3023	0,3023	0,2952	0,2972	0,3023	0,3981
E_m	0,2957	0,2957	0,2923	0,2947	0,2957	0,3968
STD$_E$	0,0015	0,0015	0,0006	0,0006	0,0015	0,0003
k_M	13	13	113	166	11	45
k_m	10	10	110	150	8	45
$Iter_M$	5	3	6	6	-	-
$Iter_m$	5	3	6	6	-	-

Tabela 5.8: Resultados obtidos após 50 execuções e 1% de erro relativo.

A utilização de um precondicionador teve um impacto considerável para acelerar a convergência do algoritmo proposto reduzindo a dimensão do espaço projetado de 166 para meros 13, o que significa uma redução de mais de 92%.

A proposta mostra ser estável no mesmo sentido já descrito, ou seja, o parâmetro de regularização encontrado não difere muito uma vez estando estabelecido o nível de ruído nos dados, independente da utilização de precondicionador ou de reortogonalização.

Comparando os resultados obtidos do algoritmo proposto com os obtidos do algoritmo HyBR, mais especificamete os campos referentes aos erros relativos nas soluções, observamos que nossa proposta é bem competitiva uma vez que os erros relativos de ambos os algortimos ficaram próximos dos 30%. Vejamos o que acontece quando o nível de ruído é elevado para 5%.

Erro:5%	Lanc-FP$^{\perp}_{prec}$	Lanc-FP$_{prec}$	Lanc-FP$^{\perp}$	Lanc-FP	HyBR$_{prec}$	HyBR
$\bar{\lambda}$	0,0277	0,0277	0,0322	0,0322	0,0549	0,0873
λ_M	0,0280	0,0280	0,0323	0,0323	0,0557	0,1553
λ_m	0,0273	0,0273	0,0321	0,0321	0,0532	0,0177
STD$_\lambda$	1,7e-4	1,7e-4	2,4e-5	2,4e-5	5,8e-4	0,0684
$\bar{E}$	0,3773	0,3773	0,3995	0,4002	0,3772	0,4864
E_M	0,3822	0,3822	0,4015	0,4022	0,3820	0,6113
E_m	0,3739	0,3739	0,3976	0,3984	0,3738	0,3554
STD$_E$	0,0018	0,0018	0,0008	0,0008	0,0018	0,1262
k_M	5	5	45	56	3	234
k_m	5	5	45	56	3	8
$Iter_M$	5	3	6	6	-	-
$Iter_m$	5	3	6	6	-	-

Tabela 5.9: Resultados obtidos após 50 execuções e 5% de erro relativo.

Após utilizarmos por 50 vezes concluímos que não há mudança significativa no comportamento do algoritmo, pois mais uma vez o desvio padrão foi da ordem de $10^{-4}, 10^{-5}$ estando de acordo com a característica herdada do algoritmo de ponto-fixo [2]. A grande surpresa neste ponto foi o algoritmo HyBR, sem a utilização de precondicionador, que apresentou um erro relativo máximo de 48% enquanto que os demais não ultrapassaram a casa dos 40%.

Finalmente, veremos os resultados para um caso que temos pouca quantidade de erro, isto é, para erros da ordem de 0,1%. Vejamos os resultados na tabela 5.10

Erro: 0,1%	Lanc-FP$^{\perp}_{prec}$	Lanc-FP$_{prec}$	Lanc-FP$^{\perp}$	Lanc-FP	HyBR$_{prec}$	HyBR
$\bar{\lambda}$	0,0002	0,0026	0,0007	0,0012	0,0103	0,0048
λ_M	0,0002	0,0042	0,0007	0,0013	0,0158	0,0049
λ_m	0,0001	0,0012	0,0007	0,0012	0,0062	0,0047
STD$_\lambda$	2,834e-3	6,784e-2	8,082e-5	2,207e-3	2,219e-1	2,851e-3
$\bar{E}$	0,2820	0,2823	0,3095	0,2999	0,2822	0,2957
E_M	0,2905	0,2911	0,3096	0,3020	0,2911	0,2959
E_m	0,2779	0,2777	0,3094	0,2996	0,2778	0,2948
k_M	54	90	81	103	90	128
k_m	42	66	81	96	56	123
$Iter_M$	3	3	3	3	-	-
$Iter_m$	3	3	3	3	-	-

Tabela 5.10: Resultados obtidos após 50 execuções e 0,1% de erro relativo.

Nenhuma surpresa que o algoritmo proposto apresente resultados similares aos descritos anteriormente.

Nas figuras 5.12 e 5.13 temos a imagem original, a imagem com ruído de 1% e as seis soluções obtidas, uma para cada possibilidade e a solução exata juntamente com o vetor de dados b perturbado.

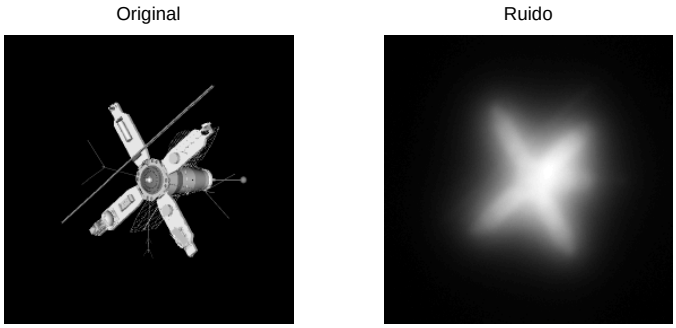

Figura 5.12: Solução exata e vetor de dados b perturbado.

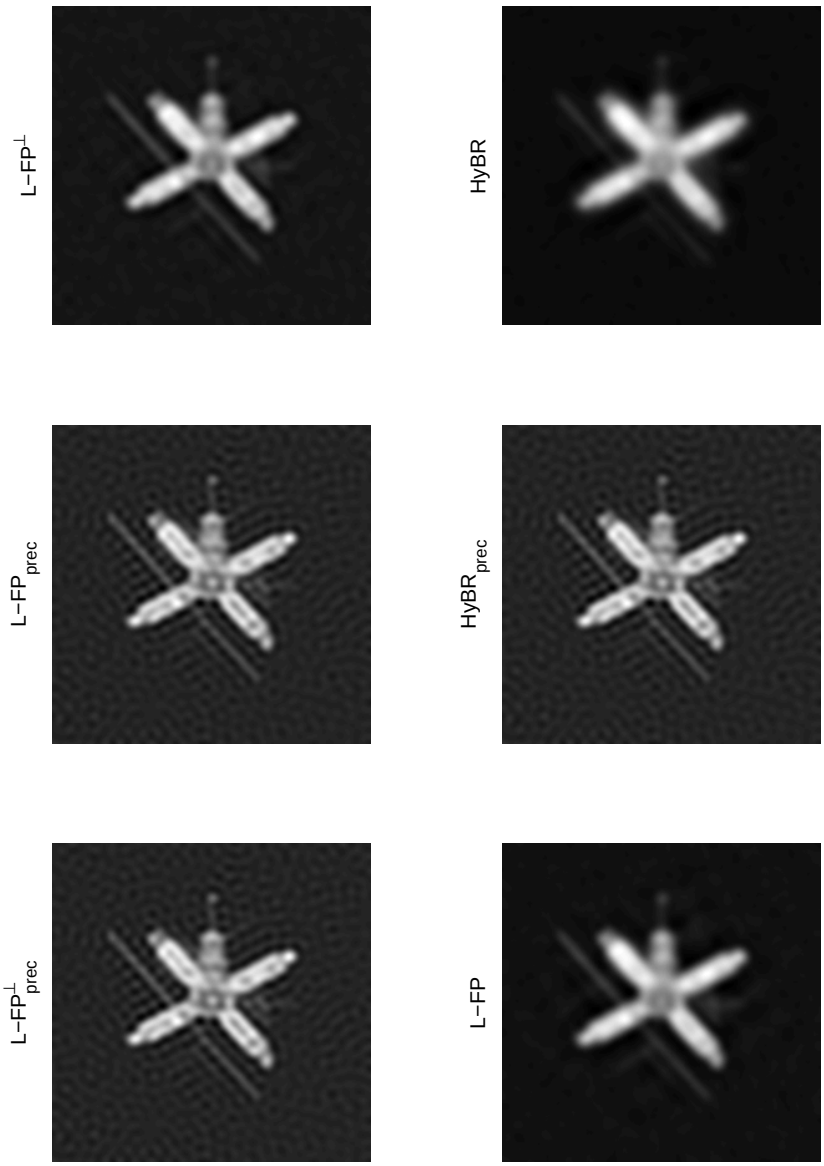

Figura 5.13: Soluções obtidas pelos métodos HyBR e Lanc-FP com e sem precondicionador e/ou reortogonalização.

5.2.2 Paisagem

Podemos obter imagens das mais diversas origens, seja em uma festa de aniversário, em um passeio de férias, ou de uma foto tirada por um satélite que se encontra no espaço. Qualquer que seja a imagem, podemos ter vários tipos de embaçamentos, como por exemplo na figura 5.14 onde podemos conferir uma foto original e duas fotos embaçadas, uma com o *motion blur* (embaçamento causado pelo movimento da câmera) e outra com o *out-of-focus blur* (embaçamento causado pelo fato da imagem estar fora de foco).

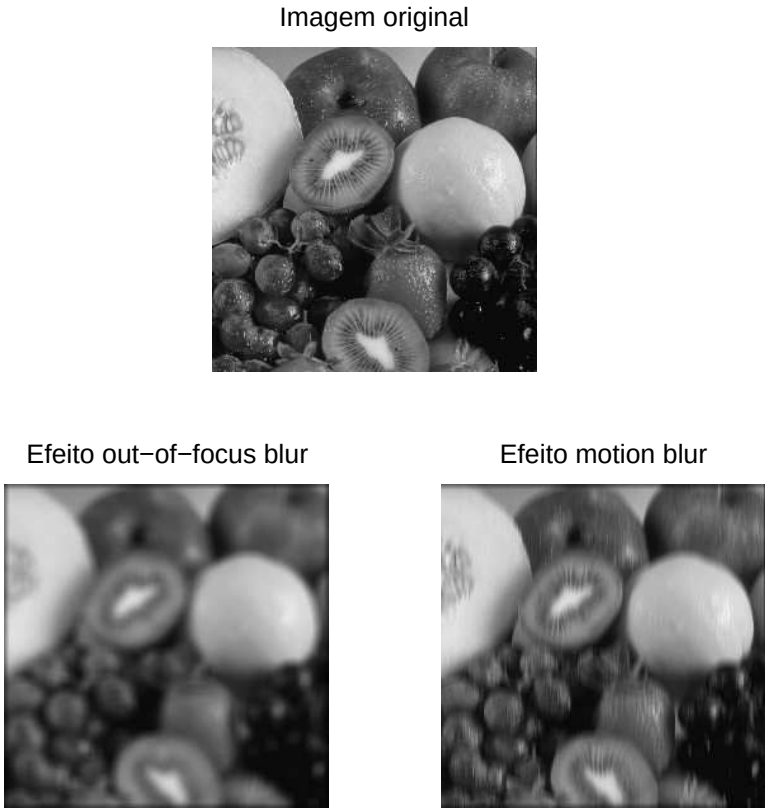

Figura 5.14: Imagem original (acima). Imagens com *out-of-focus blur* (esquerda) e *motion blur* (direita).

A seguir veremos dois exemplos de paisagens, um da ilha de Santa Catarina - Florianópolis - e outro de uma foto da praia da Armação (em Florianópolis), nos quais tentaremos remover o efeito *blur*.

Ilha de Santa Catarina

Para simular um efeito *blur* usamos a rotina *mblur* (que pode ser localizada no site [32]) com dimensão $n = 200$, o que gera um sistema de 40.000 variáveis, com ruído gerado pela rotina *WhiteNoise* localizado na toolbox RestoreTools com 0,1% de intensidade do erro. Na figura 5.15 apresentamos a imagem original que será embaçada. Nas figuras 5.15, 5.16 e 5.17, temos a imagem original, a fora de foco e a reconstruída usando nossa proposta. O erro relativo na solução foi de 2,03%, o parâmetro de regularização encontrado foi $\lambda = 4,1665 \times 10^{-4}$ e foram necessárias 40 iterações para o algoritmo parar.

Figura 5.15: Ilha de Santa Catarina. Imagem original.

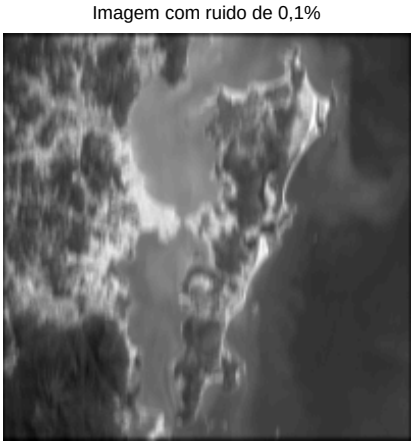

Figura 5.16: Ilha de Santa Catarina com *motion blur* e ruído de 0,1%.

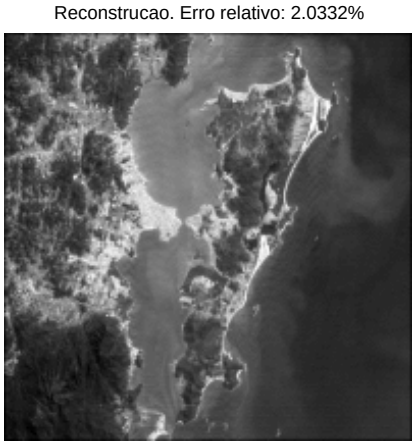

Figura 5.17: Ilha de Santa Catarina. Imagem reconstruída.

Praia da Armação

Uma imagem colorida é tratada como sendo uma imagem em três camadas pelo Matlab. Novamente para simular uma imagem levemente fora de foco usamos a rotina *oblur* com dimensão $n = 256$ o que gera um sistema com 65.536 variáveis.

Para fotos coloridas existem basicamente duas maneiras de tratar este tipo de problema [33]. Um dos modos é separar o problema em três, sendo um para cada camada e o outro modo é formar um sistema com uma dimensão três vezes maior. Mais detalhes podem ser encontrados em [33].

Neste trabalho escolhemos tratar a imagem colorida resolvendo três sistemas de 65.536 variáveis. O ruído foi gerado da mesma maneira utilizando a rotina *WhiteNoise* com intensidade de 0,1%.

Podemos conferir nas figuras 5.18, 5.19 e 5.20, as imagens original, fora de foco e a reconstruída usando nossa proposta. O erro relativo na solução foi de 5,47%. Os parâmetros de regularização encontrados foram $\lambda_1 = 0,0022$, $\lambda_2 = 0,0021$ e $\lambda_3 = 0,0020$ e os três subproblemas precisaram de 17 iterações para atingir o critério de parada.

Figura 5.18: Praia da Armação. Imagem original.

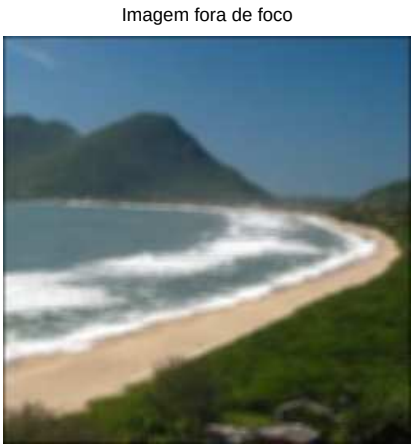

Figura 5.19: Praia da Armação com efeito *out-of-focus blur* e ruído de 0,1%.

Figura 5.20: Praia da Armação. Imagem reconstruída.

5.2.3 Ressonância magnética

Em medicina existe uma grande variedade de problemas de imagem como tomografias e ressônancias magnéticas, ultrasom, raio-X e muitas outras. No Matlab existe uma sequência de 27 imagens que representam um cérebro humano. Cada uma das imagens representa uma fatia do cérebro, todas as imagens podem ser vistas na figura 5.21.

Figura 5.21: As 27 fatias horizontais para o problema de ressonância magnética.

Este problema foi obtido de dados de uma ressonância magnética de um cérebro humano. No Matlab as imagens podem ser trabalhadas em conjunto para formar o desenho tridimensional do crânio. Mais detalhes podem ser encontrados no site [40].

De modo a testarmos o algoritmo Lanc-FP adicionamos erros de 1% nos dados e comparamos os resultados com a rotina HyBR. Na tabela 5.11 temos os erros relativos obtidos em cada uma das fatias. O mais interessante é que nossa proposta obteve resultados melhores em todos os problemas (um para cada imagem do cérebro).

Fatia	E-LancFP	E-HyBR	Fatia	E-LancFP	E-HyBR
1	0,2024	0,2132	15	0,2026	0,2180
2	0,1989	0,2079	16	0,1985	0,2117
3	0,2164	0,2245	17	0,1885	0,2044
4	0,2098	0,2224	18	0,1786	0,1930
5	0,1975	0,2106	19	0,1881	0,2023
6	0,1827	0,1949	20	0,1834	0,1985
7	0,1824	0,1963	21	0,1796	0,1970
8	0,1832	0,1976	22	0,1536	0,1706
9	0,1872	0,2033	23	0,1270	0,1437
10	0,1847	0,1982	24	0,1137	0,1269
11	0,1970	0,2104	25	0,1121	0,1251
12	0,1975	0,2106	26	0,1151	0,1241
13	0,1998	0,2153	27	0,1128	0,1167
14	0,1950	0,2113			

Tabela 5.11: Erros relativos em cada fatia obtidos com Lanc-FP e HyBR.

Na figura 5.22 temos as imagens para a fatia 15. Comparando as imagens e os erros na fatia 15 (ver tabela 5.11) constatamos que o Lanc-FP obteve um resuldado ligeiramente melhor do que o encontrado pela outra proposta.

Com isto podemos constatar que nosso algoritmo é bastante competitivo com relaï£¡ï£¡o a outros encontrados na literatura. De modo que possamos obter soluções mais precisas outras informações próprias de cada problema devem ser incorporadas na sua resolução, fazendo com que desta maneira os erros relativos sejam menores do que os encontrados quando não levamos em consideração o tipo de problema.

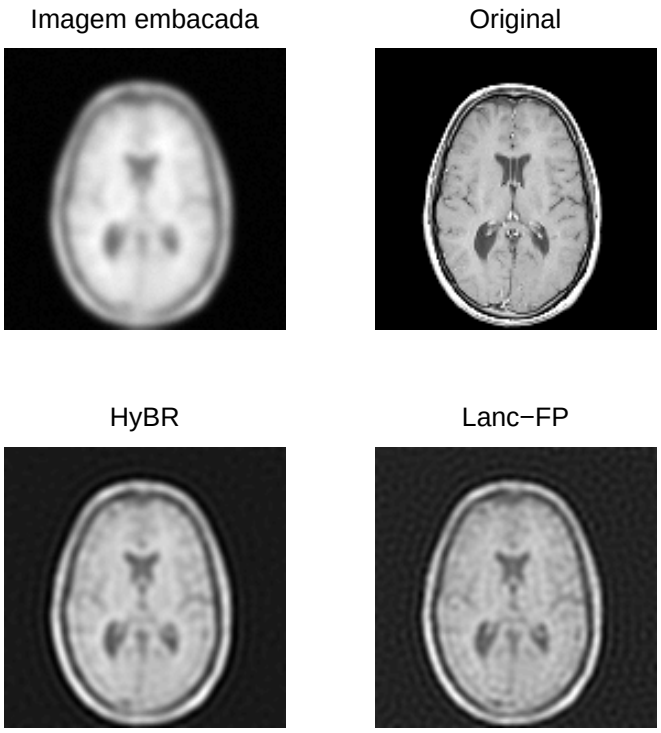

Figura 5.22: Fatia número 15 (dentre 27 disponíveis). A imagem original e duas soluções obtidas.

Conclusão

Apresentamos uma discussão sobre alguns métodos de regularização clássicos e métodos para a escolha do parâmetro de regularização. No capítulo dois realizamos uma apresentação detalhada do algoritmo de ponto-fixo novo na literatura por Bazán [2] e Bazán e Francisco [3]. Uma das grandes dificuldades de muitos métodos é que eles necessitam da SVD da matriz A e que para sistemas com m e n muito grandes a SVD é inviável. Caso a matriz A tenha alguma estrutura como, por exemplo, ser oriunda de um produto de Kronecker ou ser tridiagonal por blocos, a SVD ainda pode ser uma possibilidade apesar de ter que receber um tratamento especial, uma vez que os vetores singulares, especialmente os vetores singulares a esquerda, exigem grande quantidade de memória para serem armazenados, ver, por exemplo, [14].

Sistemas lineares de grande porte usualmente são projetados em algum subespaço de dimensão pequena. Vimos no capítulo três que o algoritmo LSQR projeta o problema de mínimos quadrados e então determina uma solução x_k neste subespaço. A principal desvantagem deste método é que ele apresenta a característica de semi-convergência e determinar a iteração de parada é uma tarefa tão difícil quanto determinar o parâmetro de regularização para Tikhonov.

Um modo de contornar esta dificuldade é regularizar o problema projetado e com um bom parâmetro de regularização fazer com que o método estabilize, ou seja, fazer com que a solução não tenha mudanças significativas a partir de certo ponto. Em relação a este ponto, temos mostrado teorica e numericamente que o algoritmo de ponto-fixo apresenta propriedades interessantes com respeito à convergência e estabilidade, no sentido de calcular consistentemente o parâmetro de regularização no caso do vetor de dados b for contaminado por ruído branco, um fato não visto em relação a curva-L e a GCV (ver [2] para mais detalhes). Isso sugeriu a utilização do método de ponto-fixo para a

determinação do parâmetro de regularização para estabilidar o método LSQR.

Com base nas características dos métodos de ponto-fixo e LSQR sugerimos um algoritmo para problemas discretos mal-postos de grande porte chamado de Lanc-FP. Desenvolvemos a teoria que suporta Lanc-FP e dentre os resultados mais relevantes estão os teoremas 4.2 e 4.3 que tratam da existência de ponto-fixo para os problemas projetados e que a sequência de pontos-fixos obtida é não-crescente. O resultado mais significativo da pesquisa foi o teorema 4.4 o qual mostra que Lanc-FP *realmente estabiliza* as iteradas [4]. O teorema 4.4 também nos diz que uma vez determinado o parâmetro de regularização na iteração k, não há necessidade de continuarmos o processo de bidiagonalização tendo em vista que a solução não muda.

Por fim, no último capítulo apresentamos resultados obtidos em simulações numéricas de diversos problemas. Além disso comparamos os resultados com W-GCV [14] e concluímos que nossa proposta é bastante competitiva e em alguns casos superior. Mesmo em casos em que a dimensão do sistema permite o cálculo dos valores singulares, a utilização do nosso algoritmo, Lanc-FP, continua sendo uma alternativa pois nesse caso o custo computacional é menor se comparado com o cálculo da SVD, pois os métodos da curva-L do Hansen [29] e a GCV do Golub [20] necessitam dos valores singulares e dos vetores singulares associados.

Como sugestões para trabalhos futuros propomos:

1. Incorporação de informações ao problema como

   - Positividade. Problemas como restauração de imagem ou o problema teste *phillips* tem soluções cujas componenetes são todas não-negativas.
   - Descontinuidade, como no exemplo *wing* em que a solução não é contínua.
   - Estender a proposta do ponto-fixo [2, 3] e deste trabalho para o caso em que a matriz L é diferente da matriz identidade, ou seja, $L \neq I$.

2. Análise de erro e determinação de múltiplos parâmetros de regularização, isto é, tratar o problema

$$x = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \left\{ \|b - Ax\|_2^2 + \lambda_1^2 \|L_1 x\|_2^2 + \lambda_2^2 \|L_2 x\|_2^2 + \ldots + \lambda_p^2 \|L_p x\|_2^2 \right\}.$$

Apêndice A

Problema de Autovalor Generalizado

O problema padrão de autovalores consiste em determinar $\lambda \in \mathbb{C}$ tal que a matriz

$$A - \lambda I \tag{A.1}$$

seja singular, isto é,

$$\det(A - \lambda I) = 0 \tag{A.2}$$

em que $A \in \mathbb{C}^{n \times n}$.

Como (A.1) é singular, sabe-se que existe um vetor não nulo $x \in \mathbb{C}^n$ tal que

$$Ax = \lambda x \tag{A.3}$$

e nesse caso x é um autovetor associado ao autovalor λ.

Seja $B \in \mathbb{C}^{n \times n}$ e considere a matriz

$$A - \lambda B \tag{A.4}$$

Os autovalores generalizados de (A.4) são os escalares $\lambda \in \mathbb{C}$ tal que (A.4) seja singular, isto é, $\det(A - \lambda B) = 0$ e denota-se por $\lambda(A, B)$ o conjunto dos autovalores generalizados do par (A, B). Um vetor não nulo $x \in \mathbb{C}^n$ é um autovetor generalizado se

$$Ax = \lambda Bx \tag{A.5}$$

Caso $\deg(p(\lambda)) < n$, isto é, o grau do polinômio $p(\lambda)$ seja menor do que n, em que $p(\lambda) = \det(A - \lambda B)$ então $\lambda = \infty$ é autovalor com multiplicidade $n - \deg(p(\lambda))$.

Uma pergunta que pode ser feita é como os autovalores generalizados de (A.4) estão relacionados com o problema padrão (A.1).

Proposição A.1. *Sejam $A, B \in \mathbb{C}^{n\times n}$ e B não singular. Então os autovalores de $A - \lambda B$ são finitos e iguais aos de AB^{-1} e $B^{-1}A$.*

Demonstração. Como B é não singular, segue que

$$A - \lambda B = AB^{-1}B - \lambda B = (AB^{-1} - \lambda I)B \tag{A.6}$$

Aplicando o determinante em ambos os lados tem-se

$$0 = \det(A - \lambda B) = \det(AB^{-1} - \lambda I)\det(B) \tag{A.7}$$

Como $\det(B) \neq 0$ vale então

$$0 = \det(A - \lambda B) = \det(AB^{-1} - \lambda I) \tag{A.8}$$

Segue análogo para $B^{-1}A$ pois $A - \lambda B = B(B^{-1}A - \lambda I)$.

□

Proposição A.2. *Sejam $A, B \in \mathbb{C}^{n\times n}$ e B singular. Então $A - \lambda B$ tem autovalor infinito com multiplicidade $n - posto(B)$.*

Demonstração. Seja $B = U\Sigma V^H$ a SVD de B. Então

$$p(\lambda) = \det(A - \lambda U\Sigma V^H) = \det(U(U^H AV - \lambda\Sigma)V^H) = \pm\det(U^H AV - \lambda\Sigma) \tag{A.9}$$

Como $posto(B) = posto(\Sigma) = k$, somente k λ's aparecem em $U^H AV - \lambda\Sigma$, logo o grau do polinômio $\det(U^H AV - \lambda\Sigma)$ é k.

□

Proposição A.3. *Sejam $A, B \in \mathbb{C}^{n\times n}$ e A não singular. Então os autovalores de $A-\lambda B$ são os inversos dos autovalores de $A^{-1}B$ e BA^{-1}. Se $\lambda = 0$ for um autovalor de $A^{-1}B$ então $\lambda = \infty$ é autovalor de $A - \lambda B$.*

Demonstração. Como $\det(A) \neq 0$ segue que

$$\begin{aligned} 0 &= \det(A-\lambda B) = \det(A)\det(I-\lambda A^{-1}B) \\ &= \det(A)\det\left(-\lambda\left(A^{-1}B - \frac{1}{\lambda}I\right)\right) \\ &= (-\lambda)^n \det(A)\det\left(A^{-1}B - \frac{1}{\lambda}I\right) \end{aligned} \tag{A.10}$$

Portanto $\frac{1}{\lambda}$ é autovalor de $A^{-1}B$. Análogo para BA^{-1}.

□

Foi mostrado até aqui que o problema de autovalor generalizado $Ax = \lambda Bx$ pode ser reduzido para o problema padrão para alguma das matrizes $AB^{-1}, A^{-1}B, B^{-1}A, A^{-1}B$.

Como a inversão de matriz pode causar mal condicionamento, destruição de estrutura como simetria ou esparsidade, além de outros problemas, deve-se buscar outras alternativas.

Duas matrizes A, B quadradas são semelhantes se existe uma matriz não singular P tal que

$$P^{-1}AP = B \tag{A.11}$$

Nessas condições A e B têm os mesmo autovalores.

Dois pares (A, B) e $(\tilde{A}, \tilde{B})$ são equivalentes se existem matrizes U e V não singulares tal que

$$\tilde{A} = UAV \text{ e } \tilde{B} = UBV \tag{A.12}$$

e são congruentes se existe uma matrix não singular P tal que

$$\tilde{A} = P^H AP \text{ e } \tilde{B} = P^H BP \tag{A.13}$$

Proposição A.4. *Pares equivalentes possuem os mesmo autovalores.*

Demonstração.

$$0 = \det(\tilde{A} - \lambda\tilde{B}) = \det(U)\det(A-\lambda B)\det(V) \tag{A.14}$$

□

Para o problema padrão (A.1) existe a forma de Schur que revela os autovalores da matriz A na decomposição $A = QTQ^H$ em que $Q \in \mathbb{C}^{n\times n}$ é unitária, $T \in \mathbb{C}^{n\times n}$ é triangular superior e os autovalores de A estão na diagonal de T.

Pode-se generalizar a forma de Schur para o par (A, B).

Teorema A.1. *(Forma de Schur Generalizada) Sejam $A, B \in \mathbb{C}^{n\times n}$. Então existem matrizes unitária s $Q, Z \in \mathbb{C}^{n\times n}$ tal que*

$$Q^H AZ = T \text{ e } Q^H BZ = S \tag{A.15}$$

em que T e S são triangulares superiores. Além disso, se para algum k tem-se $t_{kk} = s_{kk} = 0$ então $\lambda(A, B) = \mathbb{C}$. Caso contrário,

$$\lambda(A, B) = \left\{ \frac{t_{kk}}{s_{kk}} \middle/ 1 \leq k \leq n \text{ e } s_{kk} \neq 0 \right\} \tag{A.16}$$

Demonstração. Seja $(B_k)_{k\in\mathbb{N}}$ uma sequência de matrizes não singulares que converge para B. Para cada k seja $Q_k^H(AB_k^{-1})Q_k = R_k$ a forma de Schur de AB^{-1}. Seja Z_k uma matriz unitária tal que $Z_k^H(B_k^{-1}Q_k) = S_k^{-1}$ seja triangular superior. Como R_k e S_k são triangulares superiores segue que

$$Q_k^H AZ_k = R_k S_k \text{ e } Q_k^H B_k Z_k = S_k \tag{A.17}$$

também são.

Pelo teorema de Bolzano-Weierstrass sabe-se que a sequência limitada $(Q_k, Z_k)_{k\in\mathbb{N}}$ admite uma subsequência convergente

$$\lim_{i\to\infty}(Q_{k_i}, Z_{k_i}) = (Q, Z) \tag{A.18}$$

Segue que Q e Z são unitárias e que

$$Q^H AZ = T \text{ e } Q^H BZ = S \tag{A.19}$$

são triangulares superiores.

A parte final do teorema com respeito ao conjunto $\lambda(A, B)$ segue imediatamente do fato

$$\det(A - \lambda B) = \det(QZ^H)\det(T - \lambda S) = \det(QZ^H)\prod_{i=1}^{n}(t_{ii} - \lambda s_{ii}) \tag{A.20}$$

□

Caso A e B sejam reais o teorema pode ser reescrito de modo que Q e Z sejam ortogonais e T quase-triangular.

Em problemas que não há simetria das matrizes A e B é comum reduzir a matriz A para uma Hessenberg superior.

Teorema A.2. *Sejam $A, B \in \mathbb{C}^{n\times n}$. Então existem matrizes unitárias Q e Z tal que*

$$Q^H AZ = H \text{ e } Q^H BZ = S \tag{A.21}$$

em que H é Hessenberg superior e S triangular superior

Caso B seja singular existe uma expressão mais precisa para H e S.

Teorema A.3. *Sejam $A, B \in \mathbb{C}^{n\times n}$ e k a dimensão do espaço nulo de B. Então existem matrizes unitárias Q e Z tal que $Q^H AZ = H$ e $Q^H BZ = S$ em que*

$$H = \begin{bmatrix} H_{11} & H_{12} \\ 0 & H_{22} \end{bmatrix} \text{ e } S = \begin{bmatrix} 0 & S_{12} \\ 0 & S_{22} \end{bmatrix} \tag{A.22}$$

com $H_{11} \in \mathbb{C}^{k\times k}$ triangular superior, H_{22} Hessenberg superior e $S_{22} \in \mathbb{C}^{(n-k)\times(n-k)}$ triangular superior e não singular.

As estruturas das matrizes A e B não foram levadas em consideração até aqui.

Será visto agora o caso em que A e B são simétricas e quando necessário B também terá a hipótese de ser positiva definida.

Teorema A.4. *Sejam $A, B \in \mathbb{R}^{n\times n}$ simétricas e*

$$C(\mu) = \mu A + (1 - \mu)B \ , \ \mu \in \mathbb{R} \tag{A.23}$$

Se existe $\mu \in [0,1]$ *tal que* $C(\mu)$ *é não-negativa definida e*

$$N(C(\mu)) = N(A) \cap N(B) \tag{A.24}$$

então existe uma matriz X *não singular tal que* $X^T AX$ *e* $X^T BX$ *são diagonais.*

Demonstração. Seja $\mu \in [0,1]$ tal que $C(\mu)$ seja não-negativa definida e que satisfaça a equação (A.24) e seja

$$Q_1^T C(\mu) Q_1 = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}, \; D = \text{diag}(d_1, ..., d_k) > 0 \tag{A.25}$$

a forma de Schur de $C(\mu)$ e defina

$$X_1 = Q_1 \text{diag}\left(D^{-1/2}, I_{n-k}\right) \tag{A.26}$$

Se $A_1 = X_1^T AX_1$, $B_1 = X_1^T BX_1$ e $C_1 = X_1^T C(\mu) X_1$ então

$$C_1 = \begin{bmatrix} I_k & 0 \\ 0 & 0 \end{bmatrix} = \mu A_1 + (1-\mu) B_1 \tag{A.27}$$

Como $\text{span}\{e_{k+1}, ..., e_n\} = N(C_1) = N(A_1) \cap N(B_1)$ segue que A_1 e B_1 tem a seguinte estrutura de bloco

$$A_1 = \begin{bmatrix} A_{11} & 0 \\ 0 & 0 \end{bmatrix} \text{ e } B_1 = \begin{bmatrix} B_{11} & 0 \\ 0 & 0 \end{bmatrix} \tag{A.28}$$

com $A_{11}, B_{11} \in \mathbb{R}^{k \times k}$. Mais ainda, $I_k = \mu A_{11} + (1-\mu) B_{11}$.

Suponha agora $\mu \neq 0$. Segue então que se $Z^T B_{11} Z = \text{diag}(b_1, ..., b_k)$ é a forma de Schur de B_{11} e $X = X_1 \text{diag}(Z, I_{n-k})$ então

$$X^T BX = \text{diag}(b_1, ..., b_k, 0, ...0) = D_B \tag{A.29}$$

e

$$X^T AX = \frac{1}{\mu} X^T (C(\mu) - (1-\mu) B) X = \frac{1}{\mu} \left(\begin{bmatrix} I_k & 0 \\ 0 & 0 \end{bmatrix} - (1-\mu) D_B \right) = D_A \tag{A.30}$$

Por outro lado, se $\mu = 0$, seja $Z^T A_{11} Z = \text{diag}(a_1, ..., a_k)$ a forma de Schur de A_{11} e $X = X_1 \text{diag}(Z, I_{n-k})$. Desse modo $X^T AX$ e $X^T BX$ são ambas diagonais.

□

Corolário A.1. *Se $A - \lambda B \in \mathbb{R}^{n \times n}$ é simétrica-definida, isto é, $A = A^T$, $B = B^T$ e B é positiva definida, então existe uma matriz não singuar $X = [x_1, ..., x_n]$ tal que*

$$X^T AX = diag(a_1, ..., a_n) \text{ e } X^T BX = diag(b_1, ..., b_n) \tag{A.31}$$

além disso, $Ax_i = \lambda_i Bx_i$, $i = 1, ..., n$ e $\lambda_i = \dfrac{a_i}{b_i}$.

Bibliografia

[1] BAKER, C. T. H., *The Numerical Treatment of Integral Equations*, Clarendon Press, Oxford, pp. 665, 1977.

[2] BAZÁN, F. S. V., *Fixed-point iterations in determining the Tikhonov regularization parameter*, Inverse Problems, 24, 2008.

[3] BAZÁN, F. S. V. e FRANCISCO, J. B., *Improved Fixed-point algorithm for determining the Tikhonov regularization parameter*, Inverse Problems, 25, 2009.

[4] BAZÁN, F. S. V. e BORGES, L. S., *LANC-FP: an algorithm for large-scale discrete ill-posed problems*, submetido, março 2009.

[5] BAZÁN, F. S. V., *CGLS-GCV: a hybrid algorithm for low-rank-deficient problems*, App. Num. Math. 47, pp. 91-108, 2003.

[6] BELGE, M., KILMER, E. e MILLER, E. L., *Efficient determination of multiple regularization parameters in a generalized L-curve framework*, Inverse Problems 18, pp. 1161-1183, 2002.

[7] BJÖRCK, Å., *A Bidiagonalization Algorithm for Solving Large and Sparse Ill-Posed Systems of Linear Equations*, BIT 38, pp. 659-670, 1988.

[8] BJÖRCK, Å., *Numerical Methods for Least Squares Problems*, SIAM, 1996.

[9] CALVETTI, D., GOLUB, G. H. e REICHEL, L., *Estimation of the L-curve via Lanczos bidiagonalization*, BIT, 39, pp 603-619, 1999.

[10] CALVETTI, D., LEWIS, B. e REICHEL, L., *On the regularizing properties of the GMRES method*, Numer Math, 91, pp. 605-625, 2002.

[11] CALVETTI, D. e REICHEL, L., *Tikhonov Regularization of Large Linear Problems*, BIT 43, pp. 263-283, 2003.

[12] CALVETTI, D., REICHEL, L. e SHUIBI, A., *L-curve and curvature bounds for Tikhonov regularization*, Numerical Algorithms, 35, pp 301-314, 2004.

[13] CARASSO, A. S., *Determining surface temperatures from interior observations*, SIAM J. Appl. Math., 42, pp. 558-574, 1982.

[14] 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.

[15] DEMMEL, J. W., *Applied Numerical Linear Algebra*, SIAM, 1997.

[16] DOLD, A. e ECKMANN, B., *Lecture Notes in Mathematics*, Inverse Problems, 1986.

[17] ELDÉN, L., *A weighted pseudoinverse, generalized singular values, and constrained least squares problems*, BIT, 22, pp. 487-501. 1982.

[18] ELDÉN, L., *The numerical solution of a non-characteristic Cauchy problem for a parabolic equation*, in P. Deuflhard & E. Hairer (Eds.), *Numerical Treatment of Inverse Problems in Differential and Integral Equations*, Birkhäuser, 1983.

[19] GOLUB, G. H., *Numerical Methods for Solving Linear Least Squares Problems*, Numerische Mathematik 7, pp 206-216, 1965.

[20] GOLUB, G. H., HEATH, M. T. e WAHBA, G., *Generalized cross-validation as a method for choosing a good ridge parameter*, Technometrics, 21, pp. 215-223, 1979.

[21] GOLUB, G. H. e LOAN, C. F. V., *Matrix Computations*, Third Edition, The Johns Hopkins University, London, 1996.

[22] GROETSCH, C. W., *The theory of Tikhonov regularization for Fredholm equations of the first kind*, Research Notes in Mathematics, 105, Pitman, Boston, 1984.

[23] GU, M. e EISENSTAT, S. C., *A Stable and Fast Algorithm For Updating the Singular Value Decomposition*, Tech. report YALEU/DCS/RR-966, Department of Computer Science, Yale University, New Haven, 1993.

[24] HANKE, M., *Limitations of the L-curve method in ill-posed problems*, BIT 36, pp 287-301, 1996.

[25] HANKE, M. e HANSEN, P.C., *Regularization methods for large-scale problems*, Surveys Math. Indust., v3, pp. 253-315, 1993.

[26] HANSEN, P. C., *Regularization, GSVD and Truncated GSVD*, BIT 29, pp. 491-504, 1989.

[27] HANSEN, P. C., *The Discrete Picard Condition For Discrete Ill-Posed Problems*, BIT 30, pp. 658-672, 1990.

[28] HANSEN, P. C., *Analysis of discrete ill-posed problems by means of the L-curve*, SIAM Review, 34, pp. 561-580, 1992.

[29] HANSEN, P. C., *The L-curve and its use in the numerical treatment of inverse problems*; invited chapter in P. Johnston (Ed.), Computational Inverse Problems in Electrocardiology, WIT Press, Southampton, pp. 119-142, 2001.

[30] HANSEN, P. C., *Rank-deficient and discrete ill-posed problems*, SIAM Philadelphia, PA, 1998.

[31] HANSEN, P. C., *Regularization Tools: A MATLAB package for analysis and solution of discrete ill-posed problems*, Numer. Algorithms, 6 pp. 1-35, 1994.

[32] HANSEN, P. C., *Regularization Tools*, *http://www2.imm.dtu.dk/∼pch/Regutools/*.

[33] HANSEN, P. C., NAGY, J. G. e O'LEARY, D., *Deblurring Images. Matrices, Spectra and Filtering.*, SIAM, 2006.

[34] JAIN, A. K., *Fundamentals of Digital Image Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1989.

[35] JIANG, M., XIA, L., e SHOU, G., *Combining regularization frameworks for solving the electrocardiography inverse problem*, Communications in Computer and Information Science, Vol. 2, pp. 1210-1219, 2007.

[36] JIANG, M., XIA, L., SHOU, G. e TANG, M., *Combination of the LSQR method and a genetic algorithm for solving the electrocardiography inverse problem*, Physics in Medicine and Biology, 52, pp. 1277-1294, 2007.

[37] JIANG, M., XIA, L., SHOU, G., LIU, F. e CROZIER, S., *Two hybrid regularization frameworks for solving the electrocardiography inverse problem*, Physics in Medicine and Biology, 53, pp. 5151-5164, 2008.

[38] JOHNSTON, P. R. e GULRAJANI, R. M., *An analysis of the zero-crossing method for choosing regularization parameters*, SIAM J. Sci. Comput. Vol 24, No. 2, pp 428-442, 2002.

[39] KILMER, M. E. e O'LEARY, D., *Choosing regularization parameters in iterative methods for ill-posed problems*, SIAM J. Matrix Anal. Appl., Vol. 22, No. 4, pp. 1204-1221, 2001.

[40] Mathworks, *http://www.mathworks.com/products/image/demos.html?file=/products/demos/shipping/images/ipexmri.html*

[41] MOROZOV, V. A., *On the solution of functional equations by the method of regularization*, Soviet Math. Dokl., 7, pp. 414-417, 1966.

[42] NAGY, J. G., PALMER, K. e PERRONE, L., *Iterative methods for image deblurring: A MATLAB object oriented approach*, Numer. Algorithms, 36, pp. 73-93, 2004.

[43] PHILLIPS, D. L., *A technique for the numerical solution of certain integral equations of the first kind*, J. of the ACM, Vol. 9, pp. 84-97, 1962.

[44] PAIGE, C. C. e 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.

[45] PAIGE, C. C. e SAUNDERS, M. A., *LSQR: Sparse linear equations and least squares problems*, ACM Trans. Math. Softw, Vol. 8, No. 2, pp 195-209, 1982.

[46] REGIŃSKA, T., *A regularization parameter in discrete ill-posed problems*, SIAM J. Sci. Comput. Vol. 17, No. 3, pp. 740-749, 1996.

[47] RILEY, J., *Solving systems of linear equations with a positive definite, symmetric, but possibly ill-conditioned matrix*, Math. Tables Aids Comput., pp. 96–101, vol. 9, 1955.

[48] ROGGEMANN, M. C., e WELSH, B., *Imaging Through Turbulence*. CRC Press, Boca Raton, FL, 1996.

[49] TIKHONOV, A. N., *Solution of incorrectly formulated problems and the regularization method*, Soviet Math. Dokl., pp. 1035-1038, Vol. 4, 1963.

[50] VOGEL, C. R., *Non-convergence of the L-curve regularization parameter selection method*, Inverse Problems, Vol. 12, pp. 535-547, 1996.

[51] WATKINS, D. S., *Fundamentals of Matrix Computations*, John Wily & Sons, 1991.