

Editado por

Célia A. Zorzo Barcelos

Universidade Federal de Uberlândia - UFU
Uberlândia, MG, Brasil

Eliana X.L. de Andrade

Universidade Estadual Paulista - UNESP
São José do Rio Preto, SP, Brasil

Maurílio Boaventura

Universidade Estadual Paulista - UNESP
São José do Rio Preto, SP, Brasil



Sociedade Brasileira de Matemática Aplicada e Computacional

2012

A Sociedade Brasileira de Matemática Aplicada e Computacional - SBMAC publica, desde as primeiras edições do evento, monografias dos cursos que são ministrados nos CNMAC.

Para a comemoração dos 25 anos da SBMAC, que ocorreu durante o XXVI CNMAC em 2003, foi criada a série **Notas em Matemática Aplicada** para publicar as monografias dos minicursos ministrados nos CNMAC, o que permaneceu até o XXXIII CNMAC em 2010.

A partir de 2011, a série passa a publicar, também, livros nas áreas de interesse da SBMAC. Os autores que submeterem textos à série Notas em Matemática Aplicada devem estar cientes de que poderão ser convidados a ministrarem minicursos nos eventos patrocinados pela SBMAC, em especial nos CNMAC, sobre assunto a que se refere o texto.

O livro deve ser preparado em **Latex (compatível com o Miktex versão 2.7)**, as figuras em **eps** e deve ter entre **80 e 150 páginas**. O texto deve ser redigido de forma clara, acompanhado de uma excelente revisão bibliográfica e de **exercícios de verificação de aprendizagem** ao final de cada capítulo.

Veja todos os títulos publicados nesta série na página
<http://www.sbmac.org.br/notas.php>



Sociedade Brasileira de Matemática Aplicada e Computacional

2012

MÉTODOS PARA PROBLEMAS INVERSOS DE GRANDE PORTE

Fermín S. Viloche Bazán
fermin@mtm.ufsc.br

Leonardo Silveira Borges
lsbplsb@yahoo.com

Departamento de Matemática
Centro de Ciências Físicas e Matemáticas da UFSC - CFM
Universidade Federal de Santa Catarina
Florianópolis, SC



Sociedade Brasileira de Matemática Aplicada e Computacional

São Carlos - SP, Brasil
2012

Coordenação Editorial: Sandra Mara Cardoso Malta

Coordenação Editorial da Série: Eliana Xavier Linhares de Andrade

Editora: SBMAC

Capa: Matheus Botossi Trindade

Patrocínio: SBMAC

Copyright ©2012 by Fermín S. Viloche Bazán e Leonardo S. Borges. Direitos reservados, 2012 pela SBMAC. A publicação nesta série não impede o autor de publicar parte ou a totalidade da obra por outra editora, em qualquer meio, desde que faça citação à edição original.

Catálogo elaborado pela Biblioteca do IBILCE/UNESP
Bibliotecária: Maria Luiza Fernandes Jardim Froner

Bazán, Fermín S. V.

Métodos para Problemas inversos de Grande Porte - São Carlos,
SP: SBMAC, 2012, 102 p.; 20,5cm - (Notas em Matemática
Aplicada ; v. 39)

e-ISBN 978-85-8215-000-9

1. Problemas Inversos.

I. Bazán, Fermín S. V. II. Borges, Leonardo S. III. Título. IV. Série

CDD - 51

Esta é uma republicação em formato de e-book do livro original do mesmo título publicado em 2009 nesta mesma série pela SBMAC.

Agradecimentos

Às pessoas que, no desenvolvimento deste trabalho, contribuíram de diferentes formas para a sua realização, em especial aos Professores Marcio R. Fernandes, Juliano B. Francisco e Jáuber C. Oliveira por terem lido e dado sugestões que melhoraram o texto.

Ao CNPq pelo apoio financeiro.

Conteúdo

Prefácio	9
1 Generalidades sobre problemas inversos	11
1.1 Introdução	11
1.2 Problemas inversos e sua relação com problemas mal-postos .	13
1.3 Alguns exemplos	14
1.4 Entendendo as dificuldades	15
1.5 Método de regularização	17
1.5.1 Método de quase soluções	18
1.5.2 Método de regularização de Tikhonov	18
1.6 Exercícios	19
2 Problemas discretos mal-postos e Regularização de Tikhonov	21
2.1 Problemas discretos mal-postos e a SVD	21
2.2 Regularização de Tikhonov	24
2.3 Condição Discreta de Picard	30
2.4 Exercícios	33
3 Métodos de Regularização Diretos e Iterativos	35
3.1 Introdução	35
3.2 Métodos para determinação do parâmetro de regularização de Tikhonov	37
3.2.1 GCV e W-GCV	37
3.2.2 Princípio da Discrepância	39
3.2.3 Curva-L	41
3.2.4 Ponto-Fixo (FP)	44
3.2.5 LANC-FP: um algoritmo para problemas discretos mal-postos de grande porte	46
3.3 Regularização iterativa	49
3.3.1 Iteração de Landweber	50

3.3.2	Método dos Gradientes Conjugados	50
3.3.3	LSQR e GMRES	51
3.3.4	Técnica de Reconstrução Algébrica	52
3.4	Exercícios	52
4	Uso de Toolboxes na solução de Problemas Inversos	53
4.1	RegularizationTools: um tutorial	53
4.2	RestoreTools	59
4.3	FP-tools: alguns avanços	63
4.4	LANC-FP em ação	65
4.4.1	Resultados numéricos	71
5	Reconstrução e Super Resolução de Imagens	75
5.1	Reconstrução de Imagem	75
5.1.1	Astronomia	78
5.1.2	Paisagem	82
5.1.3	Ressonância magnética	84
5.2	Super resolução de imagens	86
5.3	Exercícios	90
	Referências	90

Prefácio

A imperativa necessidade de resolução de problemas provenientes de diversas áreas das ciências naturais e/ou engenharia, tem impulsionado nas últimas décadas um novo ramo da Matemática dedicado ao estudo e solução dos chamados problemas inversos. Problemas inversos aparecem em diversos ramos das ciências aplicadas, por exemplo, na interpretação de medições de instrumentos físicos, pertinentes a algum objeto desconhecido, ou no processamento de experimentos correspondentes a fenômenos naturais.

O presente livro tem como objetivo principal apresentar os aspectos teóricos e práticos na análise e resolução de problemas inversos, visando complementar material já publicado em atas/anais de diferentes reuniões científicas no país. Assim, o objetivo é mostrar a origem das dificuldades encontradas na solução de problemas inversos, bem como difundir em âmbito nacional propostas modernas para a resolução de problemas de grande porte, tais como em restauração de imagens e super resolução.

O livro está organizado em cinco capítulos. Os dois primeiros contêm generalidades sobre problemas inversos, incluindo a noção de problema mal-posto e sua relação com problemas inversos, e as idéias básicas da teoria de regularização de Tikhonov para problemas discretos mal-postos. O capítulo três trata da escolha do parâmetro de regularização, enquanto o capítulo quatro trata do uso de pacotes computacionais de uso público que incluem a toolbox RegularizationTools de Hansen [34], a toolbox RestoreTools de Nagy *et. al.* [49], bem como FP-tools, uma proposta desenvolvida recentemente pelos autores [5]. Resultados numéricos usando problemas teste disponíveis na toolbox RegularizationTools tais como *phillips*, *foxgood*, *shaw*, *heat*, etc, também são incluídos. Finalmente, o capítulo 5 descreve problemas inversos de restauração e super resolução que facilmente passam de 50.000 variáveis.

Florianópolis, 30 de abril de 2009.

Fermín S. Viloche Bazán
Leonardo Silveira Borges

Capítulo 1

Generalidades sobre problemas inversos

Apresentamos preliminares que servem como base para a compreensão da origem e dificuldades encontradas frequentemente na resolução de problemas inversos.

1.1 Introdução

Muitos sistemas ou processos físicos das ciências naturais e/ou engenharia expressos através de modelos são a fonte de inúmeros problemas em diversos ramos das ciências aplicadas. O termo *sistema* refere-se a uma entidade (física, econômica, etc) capaz de produzir algum efeito g (*saída/output*) como consequência de um estímulo ou causa f (*entrada/input*), conforme Figura 1.1.

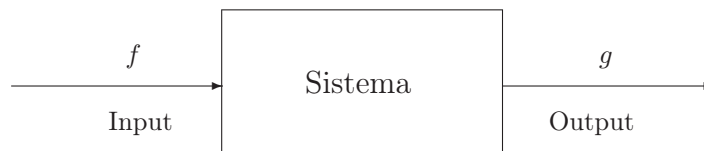


Figura 1.1: Modelagem de um processo físico.

A conexão entre o processo físico e sua modelagem matemática é usual-

mente descrita por meio de equações do tipo

$$\mathcal{A}(f) = g \quad (1.1.1)$$

onde $\mathcal{A} : \mathcal{F} \mapsto \mathcal{G}$ é um operador (diferencial, integral, etc), possivelmente não linear, $f \in \mathcal{F}$ é a função “entrada”, $g \in \mathcal{G}$ é a função “dados”, com \mathcal{F} e \mathcal{G} sendo espaços normados, chamados espaço de soluções e espaço de dados respectivamente. O problema de calcular a resposta (ou saída) de um sistema, dado o modelo e a função de entrada, é relativamente simples de ser resolvido e é conhecido como *problema direto*. Já o *problema inverso*, consiste em determinar a *causa* (ou entrada) desconhecida a partir de efeitos desejados ou observados. Em outras palavras, sendo conhecida a função g e o operador \mathcal{A} , o problema inverso corresponde à inversão do operador, caso exista. Nestas notas trataremos apenas de problemas inversos lineares.

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

$$\|g_\delta - g\| \leq \delta, \quad (1.1.2)$$

onde δ depende da precisão do instrumento e a escolha da norma é feita levando em conta considerações intimamente relacionadas com o tipo de processo físico.

O processo de leitura de sinais/medições de processos através de aparelhos (ou algum dispositivo físico) pode ser descrito pela Figura 1.1 onde para cada sinal de entrada f , uma função g é registrada na saída do aparelho. Em casos simples as funções f e g podem ser relacionadas pelo seguinte modelo linear invariante no tempo

$$\mathcal{A}f \doteq \int_c^t \Xi(t-s)f(s)ds = g(s), \quad (1.1.3)$$

onde c é uma constante apropriada e, Ξ , chamada *função de transição*, caracteriza a operação do aparelho [24]. Teoricamente $\Xi(t)$ é a função registrada pelo aparelho quando o sinal de entrada é a função generalizada delta de Dirac. Assim, o problema de interpretação de leitura de sinais através de aparelhos reduz-se à resolução da equação (1.1.3) considerando dados como em (1.1.2).

1.2 Problemas inversos e sua relação com problemas mal-postos

No processo de resolução de um problema inverso, deve ser feita uma análise prévia com relação a questões de existência, unicidade e estabilidade de soluções. Tal análise é de vital importância pois, se a função g_δ não pertence ao espaço imagem do operador, então (1.1.1) não tem solução. Outra possibilidade é que \mathcal{A} seja não injetor (o qual significa que \mathcal{A}^{-1} não existe), caso em que podem existir muitas soluções. Por outro lado, precisamos saber se a solução depende continuamente dos dados. A estabilidade é necessária se desejamos assegurar que pequenas variações nos dados produzem pequenas mudanças na solução. As considerações que acabamos de apresentar são o ponto central da discussão e análise de problemas mal-postos. A noção de problema bem-posto surgiu no início do século passado quando o matemático francês J. S. Hadamard estudava problemas de valor na fronteira, com a idéia que nenhum fenômeno natural poderia ser refletido por algum modelo que conduza a um problema mal-posto; posteriormente comprovou-se a falsidade de tais idéias.

Dizemos que o problema (1.1.1) é bem-posto no sentido de Hadamard, se as seguintes condições são satisfeitas:

1. *Existência*: Para cada $g \in \mathcal{G}$ existe $f \in \mathcal{F}$ tal que $\mathcal{A}(f) = g$.
2. *Unicidade*: Para cada $g \in \mathcal{G}$ a equação (1.1.1) tem uma única solução.
3. *Estabilidade*: A solução f depende continuamente dos dados de entrada, isto é, o operador \mathcal{A}^{-1} é contínuo.

Se alguma das condições falha o problema é chamado mal-posto. É importante destacar que a dependência contínua da solução com respeito aos dados é uma condição necessária mas não suficiente. Por isso, no caso de um problema linear bem-posto, a propagação do erro relativo na solução com respeito ao erro relativo nos dados é controlada pelo número de condição do operador [45]: $\kappa(\mathcal{A}) = \|\mathcal{A}\| \|\mathcal{A}^{-1}\|$. Especificamente, sabe-se que se resolvemos o problema perturbado $\mathcal{A}(f + \delta f) = g + \delta g$ temos que

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

onde δg é a variação nos dados e δf é a variação na solução. Se $\kappa(\mathcal{A})$ é pequeno, diz-se que o problema é *bem-condicionado* e a solução é estável com respeito a pequenas variações nos dados. Pelo contrário, se $\kappa(\mathcal{A})$ é grande, diz-se que o problema é *mal-condicionado* e o problema é instável.

Claramente, o conceito de ser bem ou mal condicionado é menos preciso que o conceito de ser bem ou mal posto.

1.3 Alguns exemplos

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

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

$$\mathcal{A}f \doteq \int_a^b K(s, t)f(t)ds = g(s), \quad c \leq s \leq d, \quad (1.3.4)$$

onde a existência e a unicidade de soluções geralmente não é garantida. Por exemplo, se \mathcal{A} é operador integral definido pela função $K(s, t)$ contínua em $[a, b] \times [c, d]$ e $f \in L_2[a, b]$, então $\mathcal{A}f \in C[s, d]$. Consequentemente, a equação (1.1.1) não tem solução em $L^2(a, b)$ se g não for contínua. Por outro lado, a equação:

$$\int_{-\pi}^{\pi} s \sin(t)f(t)dt = g(s), \quad -\pi \leq s \leq \pi$$

tem infinitas soluções pois o núcleo do operador, $\mathcal{N}(\mathcal{A})$, tem dimensão infinita. De fato, basta notar que a família infinita de funções definidas por $f_n(t) = \sin(nt)$, $n = 2, 3, \dots$, pertencem ao núcleo do operador. Para piorar, ainda que as situações acima não aconteçam, salvo o caso da função K ser da forma $K(s, t) = K_1(s)K_2(t)$ (também conhecido como o caso do núcleo degenerado), o problema sempre é instável, já que da análise funcional sabe-se que o operador inverso \mathcal{A}^{-1} é descontínuo. A título de ilustração de problemas inversos modelados pela equação integral acima, seguem dois exemplos.

Exemplo 1: (Problema de Diferenciação)

O cálculo da integral definida de uma função (numérica ou analiticamente) é um problema simples de resolver. Considerando este problema como problema direto, o problema inverso é calcular a derivada. Matematicamente, dado $f : [0, 1] \mapsto \mathbb{R}$, o problema direto é calcular

$$g(t) = \int_0^t f(s)ds, \quad t \in [0, 1],$$

enquanto no problema inverso, dada uma função $g : [0, 1] \mapsto \mathbb{R}$ diferenciável, o objetivo é determinar $f \doteq g'$ a partir da equação integral acima. Em situações reais a função g é conhecida aproximadamente (por exemplo em forma de tabela) e não há como garantir que a função g_δ seja continuamente diferenciável, ou seja, não há como garantir que o problema inverso tenha solução. Uma conclusão análoga é obtida quando procura-se a derivada n -ésima de f : neste caso $f = g^{(n)}$ é solução da equação integral

$$\int_0^t \frac{(t-s)^{(n-1)}}{(n-1)!} f(s) ds = g(t).$$

Exemplo 2: (Equação do Calor para tempo anterior)

A distribuição de temperatura U numa barra estendida ao longo do intervalo $[0, \pi]$ satisfazendo

$$\begin{cases} U_t = U_{ss}, & 0 \leq s \leq \pi, \\ U(0, t) = U(\pi, t) = 0, & t > 0 \\ U(s, 0) = f(s) & 0 \leq s \leq \pi \end{cases}$$

pode ser obtida pelo método de separação de variáveis. De fato, para $t > 0$ temos

$$U(s, t) = \frac{1}{2\pi} \sum_{n=1}^{\infty} \exp^{-n^2 t} a_n \sin(ns), \quad a_n = \int_0^\pi f(r) \sin(nr) dr.$$

Se U é conhecida no tempo $t = T$, digamos $U(s, T) = g(s)$, o problema é determinar a distribuição da temperatura inicial $U(s, 0) = f(s)$. Para tal, evidentemente devemos resolver a equação integral

$$\int_0^\pi K(s, r) f(r) dr = g(s), \quad 0 \leq s \leq \pi,$$

onde

$$K(s, r) = \frac{1}{2\pi} \sum_{n=1}^{\infty} \exp^{-n^2 T} a_n \sin(ns) \sin(nr).$$

Como o núcleo $K(s, r)$ não é degenerado, o problema inverso é mal posto.

1.4 Entendendo as dificuldades

O efeito da falta de continuidade do operador \mathcal{A}^{-1} (quando ele existe) na resolução de problemas inversos é entendido melhor quando o operador \mathcal{A} é compacto. Considere, por exemplo, o operador compacto linear

$\mathcal{A} : L^2(\Omega) \rightarrow L^2(\Omega)$ definido por

$$\mathcal{A}f = \int_{\Omega} K(\cdot, t) f(t) dt, \quad (1.4.5)$$

onde $L^2(\Omega)$ é o espaço de Hilbert de funções de quadrado integrável num domínio Ω . Neste caso, existem sistemas ortogonais $\{u_j^c\}$, $\{v_j^c\}$, em \mathcal{F} e \mathcal{G} respectivamente, e $\{\sigma_j^c\}$ com $\sigma_j^c > 0$, $j = 0, 1, 2, \dots$, tais que $\{u_j^c\}$ é uma base para $\overline{\text{Im}(\mathcal{A})} = \mathcal{N}(K^*)^\perp$ e $\{v_j^c\}$ é uma base para $\overline{\text{Im}(\mathcal{A}^*)} = \mathcal{N}(K)^\perp$; satisfazendo-se que

$$\lim_{j \rightarrow \infty} \sigma_j^c = 0, \quad (1.4.6)$$

$$\mathcal{A}f = \sum_{j=1}^{\infty} \sigma_j^c \langle u_j^c, f \rangle u_j^c \quad \forall f \in \mathcal{F}. \quad (1.4.7)$$

Os resultados acima são consequências imediatas do teorema espectral aplicado ao operador $\mathcal{A}^* \mathcal{A}$ [24]. Os números σ_j^c são chamados *valores singulares* de \mathcal{A} e o conjunto $\{\sigma_j^c, u_j^c, v_j^c\}$ é um *sistema singular* para \mathcal{A} .

Notando que toda função $g \in \text{Im}(\mathcal{A})$ pode ser escrita numa série convergente

$$g = \sum_{j=1}^{\infty} \langle u_j^c, g \rangle u_j^c, \quad (1.4.8)$$

baseado em (1.4.7) prova-se o seguinte teorema que fornece uma condição necessária e suficiente para a existência de solução da equação (1.1.1) (no contexto de operadores compactos).

Teorema 1.1 (Condição de Picard). *Uma função g dada por (1.4.8) pertence a $\text{Im}(\mathcal{A})$ se, e somente se,*

$$\sum_{j=1}^{\infty} \left(\frac{\langle u_j^c, g \rangle}{\sigma_j^c} \right)^2 < \infty. \quad (1.4.9)$$

As condições do teorema exigem uma certa regularidade da função g para (1.1.1) ter solução. Em particular, o teorema exige que a sequência de produtos internos $\{\langle u_j^c, g \rangle\}$ decresça suficientemente rápido para conter o crescimento dos fatores $1/\sigma_j^c$, que crescem sem controle devido ao comportamento dos valores singulares. Consequentemente, se $g \in \text{Im}(K)$ e existe \mathcal{A}^{-1} , então

$$f = \mathcal{A}^{-1}g = \sum_{j=1}^{\infty} \frac{\langle u_j^c, g \rangle}{\sigma_j^c} v_j^c. \quad (1.4.10)$$

A expressão acima para a solução mostra que pequenas mudanças em g podem produzir grandes variações na solução, já que $1/\sigma_j^c \rightarrow \infty$ quando j cresce. O assunto será revisto mais adiante no contexto de problemas discretos mal-postos.

Se $g \notin \text{Im}(\mathcal{A})$, ainda podemos aproximar a função g pela função g_k obtida pelo truncamento da expansão (1.4.8) em k termos

$$g_k = \sum_{j=1}^k \langle u_j^c, g \rangle u_j^c. \quad (1.4.11)$$

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

$$\mathcal{A}^{-1}g_k = \sum_{j=1}^k \frac{\langle u_j^c, g \rangle}{\sigma_j^c} v_j^c. \quad (1.4.12)$$

Disso segue que se $k \rightarrow \infty$ então $g_k \rightarrow \mathcal{Q}g$, onde \mathcal{Q} é o operador projeção ortogonal sobre $\overline{\text{Im}(\mathcal{A})}$, mas

$$\|\mathcal{A}^{-1}g_k\| \rightarrow \infty, \quad k \rightarrow \infty. \quad (1.4.13)$$

Devido às considerações acima é que equações integrais de primeira espécie são difíceis de se resolver. Mais detalhes podem ser encontrados em [25, 28] e livros de análise funcional.

1.5 Método de regularização

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

A idéia padrão de muitos métodos de regularização é construir uma família de operadores $R_\lambda : \mathcal{G} \mapsto \mathcal{F}$ limitados chamados *operadores regularizantes* que dependem de um parâmetro $\lambda > 0$ chamado *parâmetro de regularização*, que aproxime pontualmente o operador \mathcal{A}^{-1} (caso ele exista) ou operador pseudo-inversa \mathcal{A}^\dagger , ou seja

$$\lim_{\lambda \rightarrow 0} R_\lambda(g) = \mathcal{A}^{-1}(g).$$

Maiores detalhes sobre teoria de regularização podem ser encontradas em [2, 20, 25].

De modo geral, estudam-se métodos de regularização considerando: a) Problemas mal-postos como consequência de não existência ou ambiguidade na solução e, b) Problemas mal-postos como consequência de instabilidade. Dentre essas possibilidades, a segunda é a mais delicada. Nesse caso, os métodos de regularização visam a restauração da continuidade do operador inverso girando em torno das seguintes idéias:

- Restrição das soluções admissíveis por informação a priori. Por exemplo, por imposição de limitantes na solução.
- Construção de soluções estáveis por métodos variacionais.
- Análise de um problema via associação com um problema bem-posto [44].

Na prática, a manutenção da estabilidade exige informação a priori. Informação a priori geralmente é organizada em subconjuntos do espaço de soluções através de exigências tais como

- Requerimentos de monotonicidade : $f'(t) \geq 0$, $f'(t) \leq 0$,
- Requerimentos de monotonicidade e/ou convexidade : $f''(t) \geq 0$,
- Requerimentos de regularidade: $\int_a^b |f'(t)| dt \leq M$, $M > 0$.

Encerramos o capítulo apresentando dois métodos variacionais usados muito frequentemente na resolução de problemas mal-postos

1.5.1 Método de quase soluções

Se \mathcal{S} denota um suconjunto de possíveis soluções, então um procedimento para construir soluções aproximadas estáveis para problemas mal-postos, considera a resolução do problema reformulado:

$$\mathcal{P} : \text{Determinar } f \in \mathcal{S} \subset \mathcal{F} \text{ tal que } \|g_\delta - \mathcal{A}f\| \leq \delta.$$

Dizemos que f_δ é quase solução para (1.1.1), com g_δ em lugar de g , se

$$\|\mathcal{A}f_\delta - g_\delta\| = \inf\{\|\mathcal{A}f - g_\delta\|, f \in \mathcal{S}\}.$$

Se \mathcal{A} é limitado e \mathcal{S} é compacto e convexo, sabe-se que existe uma única quase solução f_δ que depende continuamente de g_δ , e se g pertence ao espaço imagem de \mathcal{A} , a quase solução associada coincide com a solução clássica [47, teor. 2.9 pag. 27]. Observar que o conceito de quase solução está intimamente relacionado com a noção de de soluções de quadrados mínimos.

1.5.2 Método de regularização de Tikhonov

Talvez o método devido a A. Tikhonov [58] seja um dos mais populares métodos de regularização utilizados na resolução de inúmeros problemas

nos mais diversos ramos das ciências aplicadas. De forma breve, o método consiste em tomar como soluções aproximadas para (1.1.1), as funções f_λ que minimizam o funcional

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

para um valor adequado de λ chamado parâmetro de regularização, onde $\Upsilon(f)$ é um funcional contínuo chamado funcional estabilizador, satisfazendo condições apropriadas.

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

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

e o operador regularizante é

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

Na formulação original, A. Tikhonov propôs

$$\Upsilon(f) = \int_a^b [v(s)f(s)^2 + w(s)f'(s)^2] ds$$

com v, w funções de peso positivas e \mathcal{F} um espaço de Sobolev. Detalhes a respeito da teoria da regularização de Tikhonov fogem ao escopo destas notas, mais informações podem ser encontradas em [25].

1.6 Exercícios

1. Mostre que para cada número real c , $f(t) = ct^2$ é uma solução da equação

$$\int_0^s (3s - 4t)f(t)dt = 0.$$

2. Seja $K(s, t)$ definido por

$$K(s, t) = \sum_{j=1}^n S_j(s)T_j(t).$$

Mostre que a equação integral de primeira espécie (1.3.4) não tem solução se a função g não pertence ao espaço gerado pelas funções S_j .

3. Considere o operador linear $\mathcal{A} : L_2[0, 1] \rightarrow L_2[0, 1]$, definido por

$$\mathcal{A}f(t) = \int_0^t f(s)ds, \quad t \in [0, 1].$$

Prove que $\|\mathcal{A}\| = 1$ e que o operador é compacto. Sug.: use o Teorema de Arzela-Ascoli.

4. Seja g_δ uma aproximação para $g \in C[0, 1]$ tal que $|g(t) - g_\delta(t)| < \epsilon$, $\forall t \in [0, 1]$. Se $\tau \pm h \in (0, 1)$, a derivada $g'(\tau)$ pode ser aproximada por

$$dg_{\epsilon,h}(\tau) = \frac{g_\delta(\tau + h) - g_\delta(\tau - h)}{2h}.$$

Se $|g'(t)| \leq M \quad \forall t \in [0, 1]$, prove que $|g'(\tau) - dg_{\epsilon,h}(\tau)| \leq hM + \frac{\epsilon}{h}$, e que se h é escolhido adequadamente, a melhor limitante satisfaz

$$|g'(\tau) - dg_{\epsilon,h}(\tau)| \leq 2M^{1/2}\epsilon^{1/2}.$$

5. Seja $\alpha > 0$ e n um inteiro positivo. Mostre que $f(s) = \alpha e^n \sin(ns)$, $s \in \mathbb{R}$, é uma solução da equação integral

$$\int_{-\infty}^{\infty} \frac{f(s)}{1 + (t - s)^2} ds = g(t), \quad t \in \mathbb{R}.$$

Sug.: use o teorema de resíduos.

6. Seja $\mathcal{A} : \mathcal{F} \rightarrow \mathcal{G}$ um operador linear limitado. Defina a curva parametrizada

$$L : \{(u(\lambda), v(\lambda)) = (\|\mathcal{A}f_\lambda - g_\delta\|^2, \|f_\lambda\|^2), \quad \lambda > 0\},$$

onde f_λ é uma solução regularizada construída pelo método de Tikhonov com $\mathcal{L} = I$.

- i) Determine $u'(\lambda)$, $u''(\lambda)$, $v'(\lambda)$ e $v''(\lambda)$. Conclua que u é uma função crescente e que v é decrescente.
 - ii) Defina $w(s) = u(1/s)$, $s > 0$. Demonstre que w é estritamente convexa.
 - iii) Determine a curvatura da curva L .
7. Seja \mathcal{A} um operador linear compacto. Prove que para todo $\lambda > 0$,

$$(\mathcal{A}^* \mathcal{A} + \lambda^2 I)^{-1} \mathcal{A}^* = \mathcal{A}^* (\mathcal{A} \mathcal{A}^* + \lambda^2 I)^{-1}.$$

Use este resultado para provar que a solução regularizada $f_\lambda \in \mathcal{N}(\mathcal{A})^\perp$.

Capítulo 2

Problemas discretos mal-postos e Regularização de Tikhonov

O objetivo deste capítulo é apresentar a versão discreta de problemas inversos envolvendo equações integrais, enfatizando dificuldades e mostrando como usar o método de Tikhonov no contexto do problema discretizado.

2.1 Problemas discretos mal-postos e a SVD

O problema de resolver uma equação integral de primeira espécie

$$\int_a^b K(s, t)f(t)dt = g(s) \quad (c \leq s \leq d) \quad (2.1.1)$$

exige o uso de algum método de discretização. Usando alguma regra de quadratura, por exemplo, obtemos

$$\int_a^b k(s, t)f(t)dt = \sum_{j=1}^n \omega_j K(s, t_j) \tilde{f}(t_j) = g(s), \quad (2.1.2)$$

onde t_1, \dots, t_n são as abscisas da regra de quadratura escolhida, e ω_j os pesos correspondentes. Note que $f(t_j)$ está sendo substituído por $\tilde{f}(t_j)$ porque em geral a regra de quadratura não calcula a integral exatamente. A seguir, usando o método de colocação, isto é, exigindo que a soma iguale

o lado direito em (2.1.2) numa série de pontos s_1, \dots, s_m , obtemos

$$\sum_{j=1}^n \omega_j K(s_i, t_j) \tilde{f}(t_j) = g(s_i), \quad i = 1, \dots, m.$$

As relações acima resultam num sistema de equações lineares da forma

$$\begin{pmatrix} \omega_1 K(s_1, t_1) & \omega_2 K(s_1, t_2) & \cdots & \omega_n K(s_1, t_n) \\ \omega_1 K(s_2, t_1) & \omega_2 K(s_2, t_2) & \cdots & \omega_n K(s_2, t_n) \\ \vdots & \vdots & \ddots & \vdots \\ \omega_1 K(s_m, t_1) & \omega_2 K(s_m, t_2) & \cdots & \omega_n K(s_m, t_n) \end{pmatrix} \begin{pmatrix} \tilde{f}(t_1) \\ \tilde{f}(t_2) \\ \vdots \\ \tilde{f}(t_n) \end{pmatrix} = \begin{pmatrix} g(s_1) \\ g(s_2) \\ \vdots \\ g(s_m) \end{pmatrix}$$

que matricialmente pode ser escrito simplesmente como

$$Ax = b,$$

onde $A \in \mathbb{R}^{m \times n}$ com $a_{ij} = \omega_j K(s_i, t_j)$, $x \in \mathbb{R}^n$ com $x_j = \tilde{f}(t_j)$ e $b \in \mathbb{R}^m$ com $b_i = g(s_i)$.

Como m pode diferir de n , é mais razoável estimar os valores $f(t_j)$ através da resolução do problema de minimização

$$x = \arg \min_{x \in \mathbb{R}^n} \|b - Ax\|_2 \quad (2.1.3)$$

onde a norma usada é a norma euclidiana.

A grande dificuldade encontrada agora é que, pelo fato da matriz se originar da discretização de um problema mal-posto, o problema de minimização torna-se muito sensível a pequenas variações no vetor b e sua solução não é de utilidade prática. A ferramenta mais poderosas para análise e resolução do problema (2.1.3) é a decomposição em valores singulares (SVD) da matriz A .

Teorema 2.1 (SVD). *Para cada $A \in \mathbb{R}^{m \times n}$, existem matrizes ortogonais $U \in \mathbb{R}^{m \times m}$ e $V \in \mathbb{R}^{n \times n}$ tais que*

$$U^T A V = \text{diag}(\sigma_1, \dots, \sigma_p) \in \mathbb{R}^{m \times n}, \quad p = \min\{m, n\} \quad (2.1.4)$$

com valores singulares σ_i , $i = 1, \dots, p$, ordenados de modo não-crescente

$$\sigma_1 \geq \dots \geq \sigma_p \geq 0.$$

Neste caso, os valores singulares da matriz A seguem a tendência dos valores singulares σ_j^c do operador integral (ver (1.4.6)), decaindo para zero sem que haja um salto notório neste decaimento, e o um número de condição

da matriz, $\kappa(A) = \frac{\sigma_1}{\sigma_n}$ [33] seja muito grande. Somando isto ao fato do vetor b ser da forma $b = b^{\text{exato}} + e$, com b^{exato} sendo o vetor sem perturbações desejado e desconhecido, e e um vetor de incertezas, a conclusão é que 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. De fato, usando a SVD segue imediatamente que

$$x_{LS} = \sum_{i=1}^r \frac{u_i^T b}{\sigma_i} v_i. \quad (2.1.5)$$

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

$$x_{LS} = \sum_{i=1}^r \frac{u_i^T b}{\sigma_i} v_i = \sum_{i=1}^r \left(\frac{u_i^T b^{\text{exato}}}{\sigma_i} v_i \right) + \left(\sum_{i=1}^r \frac{u_i^T e}{\sigma_i} v_i \right) \quad (2.1.6)$$

e podemos notar que para valores singulares pequenos os coeficientes $\frac{u_i^T e}{\sigma_i}$ são grandes fazendo com que a segunda parcela da relação acima seja dominante, tornando assim, a solução obtida inútil. Hansen [30] definiu esse tipo de problema como sendo *Problema Discreto Mal-Posto*.

Se a série for truncada em $s < n$, 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 de s deve estabelecer um balanço apropriado entre a quantidade de informação do problema que é capturada e a quantidade de erro que é incluída na solução.

A escolha do índice s satisfazendo esse requerimento é conhecido como o método da SVD Truncada (*Truncated SVD*), em outras palavras, a TSVD é um método de regularização cuja solução é escrita da forma

$$x_s = \sum_{i=1}^s \frac{u_i^T b}{\sigma_i} v_i \quad (2.1.7)$$

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

Para ilustração, na figura 2.1 mostramos quatro soluções para o problema *baart* [34], cuja solução exata é representada pela linha pontilhada. Podemos notar 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 apropriado é $s = 3$ pois, dentre todas as soluções x_s , o erro relativo $\|x_3 - x^{\text{exato}}\|_2 / \|x^{\text{exato}}\|_2$ é o menor.

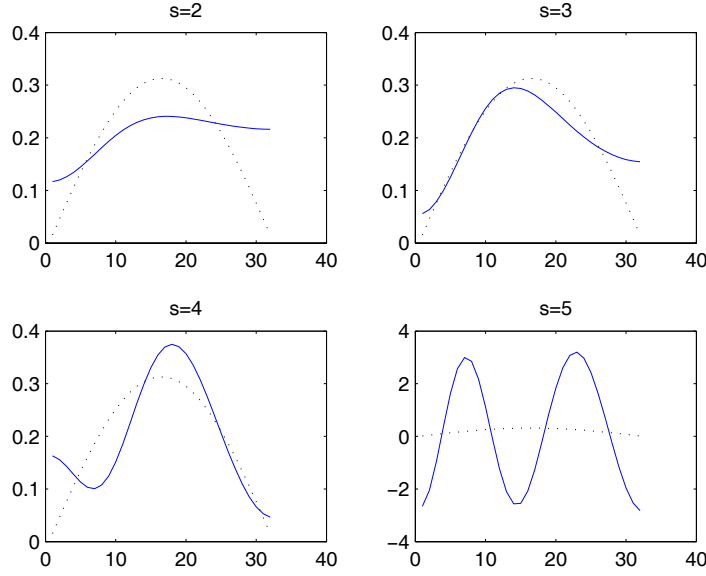


Figura 2.1: Solução x_s para $s = 2, 3, 4, 5$ para o problema *baart* com dimensão $n = 32$ e erro relativo de 1% nos dados (vetor b).

2.2 Regularização de Tikhonov

Para o problema discretizado (2.1.3), a regularização de Tikhonov substitui o problema

$$x = \operatorname{argmin}_{x \in \mathbb{R}^n} \|b - Ax\|_2^2 \quad (2.2.8)$$

por

$$x_\lambda = \operatorname{argmin}_{x \in \mathbb{R}^n} \{\|b - Ax\|_2^2 + \lambda^2 \|L(x - x_0)\|_2^2\} \quad (2.2.9)$$

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 a matriz identidade, i.e., $L = I$, ou uma aproximação discreta de algum operador diferencial definido, por exemplo, pela primeira ou segunda derivada. O vetor x_0 é uma aproximação inicial para a solução caso esteja disponível, caso contrário define-se $x_0 = 0$.

Talvez o primeiro autor a descrever um esquema equivalente à regularização de Tikhonov tenha sido James Riley em 1955 [55] que propôs resolver o sistema $(A + \alpha I)x = b$ em que α é uma constante positiva pequena. Ri-

ley também sugeriu um esquema de iteração que hoje é conhecido como a regularização de Tikhonov iterada.

Em 1962, Phillips escreveu um artigo [50] 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 (2.2.9). A ideia é tratar (2.2.9) como um problema de mínimos quadrados

$$x_\lambda = \operatorname{argmin}_{x \in \mathbb{R}^n} \left\| \begin{pmatrix} A \\ \lambda L \end{pmatrix} x - \begin{pmatrix} b \\ \lambda L x_0 \end{pmatrix} \right\|_2^2 \quad (2.2.10)$$

cujas equações normais são

$$(A^T A + \lambda^2 L^T L) x_\lambda = A^T b + \lambda^2 L^T L x_0. \quad (2.2.11)$$

A idéia chave que permite verificar esta afirmação é notar que resolver o problema (2.1.3) é equivalente a resolver o sistema das equações normais

$$A^T A x = A^T b. \quad (2.2.12)$$

As equações em (2.2.11) são chamadas 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 (2.2.11) pode ser escrita, para $x_0 = 0$ (o que mais ocorre), como

$$x_\lambda = (A^T A + \lambda^2 L^T L)^{-1} A^T b. \quad (2.2.13)$$

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

$$x_\lambda = \operatorname{argmin}_{x \in \mathbb{R}^n} \{ \|b - Ax\|_2^2 + \lambda^2 \|L(x - x_0)\|_2^2 \} \quad (2.2.14)$$

em

$$\bar{x}_\lambda = \operatorname{argmin}_{\bar{x} \in \mathbb{R}^n} \{ \|\bar{b} - \bar{A}\bar{x}\|_2^2 + \lambda^2 \|(\bar{x} - \bar{x}_0)\|_2^2 \}. \quad (2.2.15)$$

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

$$L_A^\dagger \equiv \left(I_n - (A(I_n - L^\dagger L))^\dagger A \right) L^\dagger. \quad (2.2.16)$$

Para detalhes a respeito desta inversa generalizada, ver Eldén [18].

Com a utilização de uma matriz $L \neq I$ faz-se necessária a introdução da Decomposição em Valores Singulares Generalizados (GSVD) [33]. A generalização que trabalhamos aqui está intimamente ligada com um problema de autovalor generalizado [33], 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 2.2. (GSVD) *Seja o par matricial (A, L) em que*

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

Então existem matrizes $U \in \mathbb{R}^{m \times n}$, $V \in \mathbb{R}^{p \times p}$ com $U^T U = I_n$, $V^T V = I_p$ 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 $\Sigma_p = \text{diag}(\sigma_1, \dots, \sigma_p)$ e $M_p = \text{diag}(\mu_1, \dots, \mu_p)$. Os coeficientes σ_i e μ_i satisfazem

$$0 \leq \sigma_1 \leq \dots \leq \sigma_p \leq 1 \quad \text{e} \quad 1 \geq \mu_1 \geq \dots \geq \mu_p \geq 0$$

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 2.2 mostramos os valores singulares da matriz A para o problema *shaw* [34] 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.

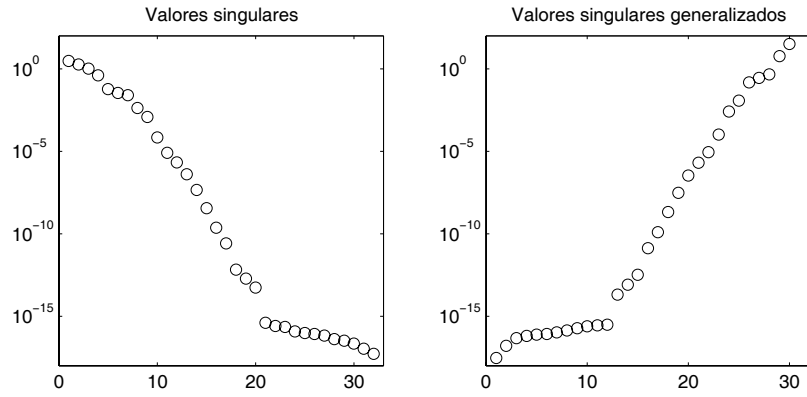


Figura 2.2: Os 32 valores singulares σ_i da matriz A e os 30 valores singulares generalizados γ_i do par matricial (A, L) para o problema *shaw* com L sendo uma aproximação para o operador segunda derivada.

Uma outra característica que a SVD e a GSVD têm em comum é o comportamento oscilatório dos vetores singulares à medida que os valores singulares (generalizados) se aproximam de zero. Conforme mostrado na figura 2.3 (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.

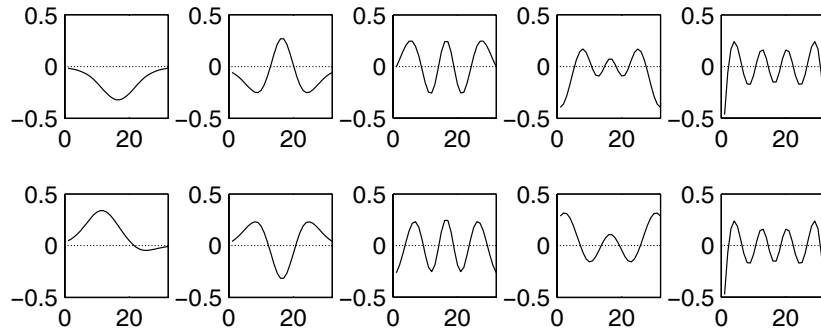


Figura 2.3: Vetores singulares u_i , $i=1,3,5,7,9$ (acima) e vetores singulares generalizados u_i , $i=32,30,28,26,24$ (abaixo).

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

$$(A^T A + \lambda^2 L^T L)x_\lambda = A^T b \quad (2.2.17)$$

pode ser escrita como

$$x_\lambda = \sum_{i=1}^r \frac{\gamma_i^2}{\gamma_i^2 + \lambda^2} \frac{u_i^T b}{\gamma_i} v_i \quad (2.2.18)$$

e truncando o somatório em $s < r$ temos um método de regularização, TGSVD. Mais detalhes podem ser encontrados em [29].

Agora que vimos um pouco sobre a GSVD podemos voltar para a transformação da forma geral para a forma padrão. No caso em que $p \geq n$ 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

$$x_{\mathcal{N}(L)} \equiv (A(I_n - L^\dagger L))^\dagger b. \quad (2.2.19)$$

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

$$L_A^\dagger = X \begin{pmatrix} M^{-1} \\ 0 \end{pmatrix} V^T, \quad x_{\mathcal{N}(L)} = \sum_{i=p+1}^n u_i^T b x_i. \quad (2.2.20)$$

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

$$\bar{A} = AL_A^\dagger, \quad \bar{b} = b - Ax_{\mathcal{N}(L)}, \quad \bar{x}_0 = Lx_0 \quad (2.2.21)$$

enquanto que a transformação da solução é dada por

$$x_\lambda = L_A^\dagger \bar{x}_\lambda + x_{\mathcal{N}(L)}. \quad (2.2.22)$$

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, $U_p = (u_1, \dots, u_p)$, então a expressão

$$AL_A^\dagger = U_p \Sigma M^{-1} V^T \quad (2.2.23)$$

mostra que os valores singulares generalizados γ_i são os valores singulares da matriz AL_A^\dagger , exceto pela ordem reversa. Mais ainda, os vetores u_i e v_i , $i = 1, \dots, p$, são os vetores singulares à esquerda e à 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 $Lx_{N(L)} = 0$) e $Ax - b = \bar{A}\bar{x} - \bar{b}$ leva imediatamente às equações

$$\|Lx\|_2 = \|\bar{x}\|_2, \quad \|Ax - b\|_2 = \|\bar{A}\bar{x} - \bar{b}\|_2. \quad (2.2.24)$$

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 (2.2.11) temos

$$(A^T A + \lambda^2 I)x_\lambda = A^T b \quad (2.2.25)$$

e usando a SVD da matriz A segue que

$$x_\lambda = \sum_{i=1}^r f_i \frac{u_i^T b}{\sigma_i} v_i, \quad r = \text{posto}(A) \quad (2.2.26)$$

em que

$$f_i = \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} \cong \begin{cases} 1 & , \quad \sigma_i \gg \lambda \\ \frac{\sigma_i^2}{\lambda^2} & , \quad \sigma_i \ll \lambda \end{cases} \quad (2.2.27)$$

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

$$r_\lambda = b - Ax_\lambda = \sum_{i=1}^r (1 - f_i) u_i^T b u_i + b_\perp \quad (2.2.28)$$

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 pertence 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

$$x_{LS} = A^\dagger b = V \Sigma^\dagger U^T b. \quad (2.2.29)$$

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

$$x_\lambda = A^\# b = VF\Sigma^\dagger U^T b \quad (2.2.30)$$

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 $i = 1, \dots, s$ e $f_i = 0$ para $i = s + 1, \dots, r$.

2.3 Condição Discreta de Picard

No início deste capítulo comentamos que a matriz A do problema (2.1.3) pode ter os valores singulares indo para zero sem que haja um salto. Na Figura 2.2 vimos que os primeiros 20 valores singulares tem um decaimento quase que constante e que 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 $K(x, y) = \sum_{i=1}^{\infty} \sigma_i^c u_i^c(s) v_i^c(t)$ a expansão em valores singulares do nú-

cleo do operador compacto \mathcal{A} e seja $g(s) = \sum_{i=1}^{\infty} \beta_i u_i(s)$ uma função. Sabemos

que para a equação integral $\mathcal{A}f = g$ ter uma solução de mínimos quadrados de quadrado integrável é necessário e suficiente que g satisfaça a condição de Picard:

$$\sum_{i=1}^{\infty} \left(\frac{\langle u_i^c, g \rangle}{\sigma_i^c} \right)^2 < \infty \quad (2.3.31)$$

em que $\langle \cdot, \cdot \rangle$ denota o produto interno usual no espaço $L^2(a, b)$.

A equação (2.3.31) implica que em certo ponto do somatório os coeficientes de Fourier $\langle u_i^c, g \rangle$ decaem mais rápido que os valores singulares σ_i^c .

Para o caso finito a equação (2.3.31) 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 , $u_i^T b$, ainda representam um papel importante no sucesso da regularização de Tikhonov pois isso determina quão boa a solução regularizada x_λ pode ser.

O resultado central é que se os 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 à 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 $|u_i^T b|$ 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 . Intuitivamente falando, isto significa que para a maior parte dos índices i , os coeficientes de Fourier satisfazem a desigualdade $|u_i^T b| \leq \gamma_i$, veja a Figura 2.4. Maiores informações podem ser encontradas em [30].

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). 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 2.4 temos uma equação de Fredholm [50] (problema *phillips*) 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 (\cdot). 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 (\cdot) fazendo com que a razão $|u_i^T b|/\sigma_i$ seja grande à medida que i cresce.

No entanto, com uma escolha apropriada do parâmetro de regularização

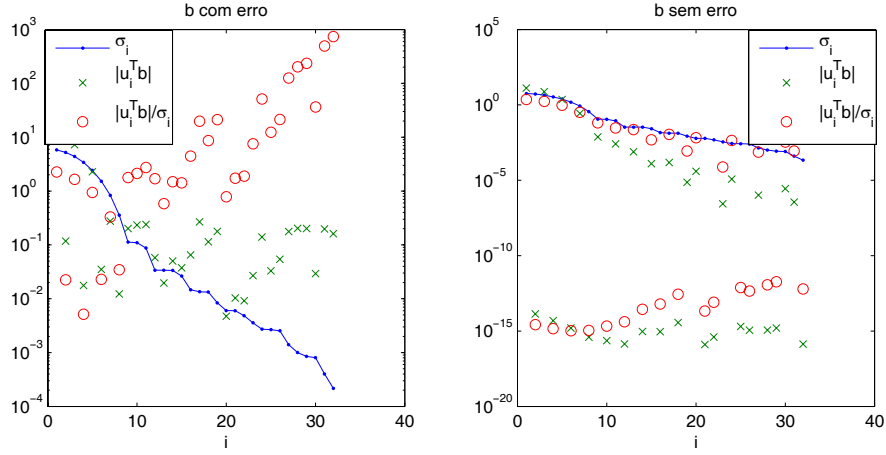


Figura 2.4: Valores singulares da matriz A do problema *phillips*, coeficientes de Fourier e a razão entre eles. Lado esquerdo: vetor b contendo erro relativo de 5%; Lado direito: b livre de erros.

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 2.5. A escolha deste parâmetro será discutida posteriormente.

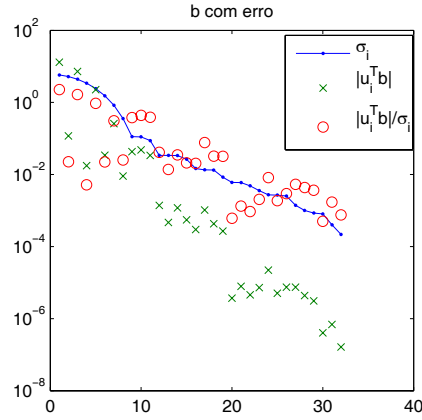


Figura 2.5: Valores singulares para a matriz A do problema *phillips*, coeficientes de Fourier e a razão entre eles considerando a utilização de um parâmetro de regularização λ apropriado.

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* [34], este problema não tem solução de quadrado integrável, logo não satisfaz a CDP como mostrado na figura 2.6, portanto não é possível calcular uma boa solução.

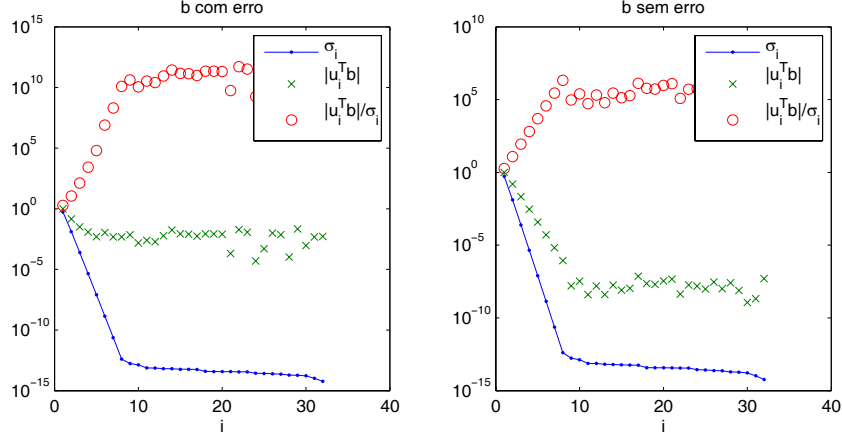


Figura 2.6: Valores singulares para a matriz A do problema *ursell*, coeficientes de Fourier e a razão entre eles considerando o vetor b sem erros e com erro relativo de 5%.

2.4 Exercícios

1. Uma outra forma de construir aproximações para a solução regularizada

$$f_\lambda = \operatorname{argmin}\{\|\mathcal{A}f - g_\delta\|^2 + \lambda^2\|f\|^2\}$$

é minimizando o funcional de Tikhonov num subespaço de \mathcal{F} de dimensão finita. Seja V_n um subespaço de \mathcal{F} gerado pelos n vetores linearmente independentes $\{v_1, \dots, v_n\}$. Prove que $f_\delta^{(n)} \in V_n$ minimiza $J_\lambda(f, g_\delta)$ sobre V_n se e somente se

$$f_\delta^{(n)} = \sum_{j=1}^n \alpha_j v_j,$$

onde os coeficientes α_j satisfazem o sistema de equações lineares

$$(A + \lambda^2 B)\alpha = \hat{g},$$

com

$$\hat{g} = [< \mathcal{A}v_1, g_\delta >_{\mathcal{G}}, \dots, < \mathcal{A}v_n, g_\delta >_{\mathcal{G}}]^T,$$

$$a_{i,j} = < \mathcal{A}v_i, \mathcal{A}v_j >_{\mathcal{G}}, \quad i, j = 1, \dots, n,$$

e

$$b_{i,j} = < v_i, v_j >_{\mathcal{F}}, \quad i, j = 1, \dots, n.$$

2. Seja $\mathcal{A} : L_2[0, 1] \rightarrow L_2[0, 1]$ tal que

$$(\mathcal{A}f)(s) = \int_0^s f(t) dt.$$

Mostre que a aproximação f_λ para a solução da equação $\mathcal{A}f = g$, construída pelo método de Tikhonov, onde g é diferenciável e $g(0) = 0$, é uma solução do problema de valor de fronteira

$$\begin{cases} \lambda^2 f_\lambda''(t) - f_\lambda'(t) = -g'(t) \\ f_\lambda(1) = f_\lambda'(0) = 0. \end{cases}$$

3. Seja $A = U\Sigma V^T$ a SVD da matriz A $m \times n$, $m \geq n$, e b um vetor $m \times 1$. Defina

$$x_k = \sum_{j=1}^k \frac{u_j^T b}{\sigma_j} v_j, \quad k = 1, \dots, n, \quad \text{e} \quad r_k = b - Ax_k.$$

Mostre que enquanto $\|x_k\|_2$ cresce com k , $\|r_k\|_2$ decresce. O que podemos afirmar do problema $\min_{x \in \mathbb{R}^n} \|b - Ax\|_2$ se $r_k = 0$ para algum k ? Qual o significado do x_k nesse caso?

4. Seja $A \in \mathbb{R}^{m \times n}$ de posto $r < n \leq m$. Defina

$$B(\lambda) = (A^T A + \lambda I)^{-1} A^T, \quad \lambda > 0.$$

Demonstre que $B(\lambda) \rightarrow A^\dagger$ quando $\lambda \rightarrow 0$. Sug.: use a SVD de A e demonstre que $\|B(\lambda) - A^\dagger\|_2 \rightarrow 0$ quando $\lambda \rightarrow 0$.

5. Demonstre que as equações normais generalizadas:

$$(A^T A + \lambda^2 L^T L)x = A^T b$$

tem uma única solução se e somente se $\mathcal{N}(A) \cap \mathcal{N}(L) = \{0\}$. Use a GSVD para verificar que x_λ pode ser escrito por meio da fórmula (2.2.18).

Capítulo 3

Métodos de Regularização Diretos e Iterativos

Veremos neste capítulo algumas das técnicas existentes na literatura para a determinação do parâmetro de regularização de Tikhonov bem como alguns métodos iterativos de regularização.

3.1 Introdução

A resolução numérica de problemas discretos mal-postos do tipo (2.1.3) tem sido motivo de extensas pesquisas ao longo das últimas décadas, pois a solução usual de mínimos quadrados não é de utilidade prática por estar contaminada por ruídos. Para contornar tal dificuldade o método de regularização de Tikhonov é frequentemente utilizado. A forma mais simples da regularização de Tikhonov [58] substitui o problema usual de mínimos quadrados por

$$x_\lambda = \operatorname{argmin}_{x \in \mathbb{R}^n} \{\|b - Ax\|_2^2 + \lambda^2 \|x\|_2^2\} \quad (3.1.1)$$

em que o parâmetro de regularização $\lambda > 0$ deve ser convenientemente escolhido. Na literatura existem diversos métodos para encontrar tal parâmetro. Dentre estes podemos citar a curva-L de Hansen [31, 32], a Validação Cruzada-Generalizada (GCV) de Golub, Heath e Wahba [22], o princípio da discrepância dado por Morozov [48] e recentemente um algoritmo de ponto-fixado dado por Bazán [3]. Outros métodos podem ser encontrados em [15, 29, 41, 42].

O critério da curva-L e a GCV são métodos que têm sido utilizados com sucesso em diversos problemas. Os mesmos são bastantes conhecidos

no meio científico e vem sugerindo novas propostas como a W-GCV por Chung, Nagy e O’Leary [15] e a utilização de *ribbons* [10, 12, 13] 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 [27] e Vogel [59]. 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.

A regularização de Tikhonov pertence à classe dos métodos 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 [33], a TGSVD [29], GMRES [11] e o LSQR de Paige e Saunders [51, 52], este último é analiticamente idêntico ao Gradiente Conjugados [33]. Tais métodos podem ser considerados métodos iterativos, pois vão, iteração por iteração, construindo a solução através de uma sequência de soluções x_k , $k = 1, 2, \dots$, 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 é capturadas, porém, se as iterações persistirem, as novas componentes passam a conter mais contribuições do erro nos dados, os quais podem desestabilizar a 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 [38, 39, 40] que utiliza a curva-L para determinar o parâmetro de regularização em cada iteração do método de projeção LSQR [51, 52]. Outras propostas podem ser encontradas em [8, 12, 15].

3.2 Métodos para determinação do parâmetro de regularização de Tikhonov

3.2.1 GCV e W-GCV

A Validação Cruzada Generalizada, do inglês Generalized Cross-Validation (GCV), desenvolvida por Golub, Heath e Wahba [22], é um método muito popular para a escolha do parâmetro de regularização baseada em considerações estatísticas. A GCV sugere que um bom valor para o parâmetro de regularização deve prever dados no vetor b que estejam faltando ou que foram retirados. Mais precisamente, se um elemento (uma equação do sistema de equações $Ax = b$) arbitrário b_j 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

$$G_{A,b}(\lambda) = \frac{n\|(I - AA_\lambda^\dagger)b\|_2^2}{\left(\text{tr}(I - AA_\lambda^\dagger)\right)^2} \quad (3.2.2)$$

em que $A_\lambda^\dagger = (A^T A + \lambda^2 I)^{-1} A^T$ representa a pseudoinversa da matriz \bar{A} do sistema $\bar{A}x_\lambda = \bar{b}$ em que $\bar{A} = (A^T A + \lambda^2 I)$ e $\bar{b} = A^T b$, 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

$$G_{A,b}(\lambda) = \frac{n \left(\sum_{i=1}^n \left(\frac{\lambda^2 u_i^T b}{\sigma_i^2 + \lambda^2} \right)^2 + \sum_{i=n+1}^m (u_i^T b)^2 \right)}{\left((m - n) + \sum_{i=1}^n \frac{\lambda^2}{\sigma_i^2 + \lambda^2} \right)^2} \quad (3.2.3)$$

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

$$\sum_{i=1, i \neq j}^m (b_i - [Ax]_i)^2 + \lambda^2 \|x\|_2^2. \quad (3.2.4)$$

Seja $E_j = \text{diag}(1, \dots, 1, 0, 1, \dots, 1) \in \mathbb{R}^{m \times m}$ em que o elemento 0 aparece na j -ésima posição. Então a minimização acima é equivalente a

$$x = \underset{x \in \mathbb{R}^n}{\text{argmin}} \left\{ \|E_j(b - Ax)\|_2^2 + \lambda^2 \|x\|_2^2 \right\}. \quad (3.2.5)$$

Em estudos comparativos encontrados na literatura podemos apontar que uma desvantagem do método GCV para a determinação do parâmetro de regularização de Tikhonov é que nem sempre este método funciona, no sentido de que dependendo de como o erro está distribuído nos dados, o parâmetro encontrado pode produzir uma solução não satisfatória.

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

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

$$G_{A,b}(\omega, \lambda) = \frac{n\|(I - AA_\lambda^\dagger)b\|_2^2}{\left(\text{tr}(I - \omega AA_\lambda^\dagger)\right)^2}. \quad (3.2.6)$$

Seja $0 < \omega \leq 1$ e $F_j = \text{diag}(1, \dots, 1, \sqrt{1-\omega}, 1, \dots, 1) \in \mathbb{R}^{m \times m}$ em que $\sqrt{1-\omega}$ aparece na j -ésima posição, e, assim como no caso da GCV, procuramos uma solução para o problema de minimização

$$x = \underset{x \in \mathbb{R}^n}{\text{argmin}} \left\{ \|F_j(b - Ax)\|_2^2 + \lambda^2 \|x\|_2^2 \right\}. \quad (3.2.7)$$

Neste caso a j -ésima observação é mantida (com peso $\sqrt{1-\omega}$) se $0 < \omega < 1$, mas desaparece se $\omega = 1$. Uma conclusão similar é obtida se $\omega \geq 1$ e $F_j = \text{diag}(1, \dots, 1, \sqrt{\omega-1}, 1, \dots, 1)$. Usando a SVD da matriz A podemos escrever o denominador em (3.2.6) como

$$\begin{aligned} \text{tr}(I - \omega AA_\lambda^\dagger) &= \sum_{i=1}^n \frac{(1-\omega)\sigma_i^2 + \lambda^2}{\sigma_i^2 + \lambda^2} + (m-n) \\ &= \sum_{i=1}^n (1-\omega)f_i + \sum_{i=1}^n \frac{\lambda^2}{\sigma_i^2 + \lambda^2} + (m-n), \end{aligned} \quad (3.2.8)$$

onde os f_i são os fatores de filtro definidos anteriormente. Então, se $\omega < 1$ estamos adicionando um múltiplo da soma dos fatores de filtro ao traço do termo original, se $\omega > 1$ estamos subtraindo um múltiplo. O gráfico da função GCV também sofre mudanças quando ω deixa de assumir o valor 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 [15] podem ser encontradas informações a respeito de como, de maneira adaptativa, o coeficiente ω pode ser determinado.

Na Figura 3.1 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 $\omega = 1,2$.

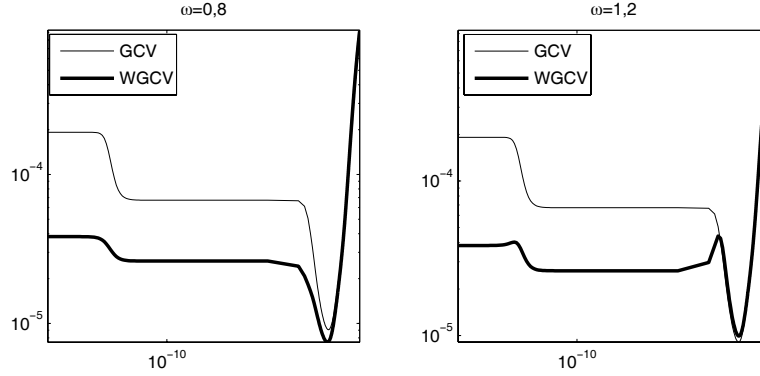


Figura 3.1: Curvas GCV e W-GCV com $\omega = \{0,8 ; 1,2\}$ para o problema *heat* com 4% de erros nos dados.

Comparando as curvas da Figura 3.1 podemos perceber que o valor do parâmetro ω influencia especialmente o comportamento da curva para valores pequenos de λ .

3.2.2 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 [48]. Se o problema mal-posto é consistente no sentido de que $Ax^{exato} = b^{exato}$ 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

$$\|b - Ax_\lambda\|_2 = \delta, \quad \|e\|_2 \leq \delta. \quad (3.2.9)$$

Usando a SVD da matriz A na equação (2.2.28) temos que a norma do resíduo pode ser escrita como

$$\|r_\lambda\|_2^2 = \sum_{i=1}^r ((1 - f_i)u_i^T b)^2 + \|b_\perp\|_2^2, \quad (3.2.10)$$

com b_\perp introduzido em (2.2.28), e podemos perceber que a norma do resíduo é uma função crescente e monótona. Então, resolver (3.2.9) equivale a encontrar a intersecção entre a curva da norma do resíduo e a reta horizontal $z = \delta$.

O princípio da discrepância é um dos poucos métodos existentes na literatura que tem análise de erro, ao contrário dos critérios da GCV e da curva-L

(que será visto mais adiante) que não possuem. 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 o erro e tiver certa estrutura [25].

Da teoria geral de regularização de Tikhonov [25] sabemos que uma limitante superior para o erro na solução x_{λ_e} é

$$\|x^{\text{exato}} - x_{\lambda_e}\|_2 = \mathcal{O}(\|e\|_2) \quad (3.2.11)$$

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

$$\|x^{\text{exato}} - x_{\lambda_e}\|_2 \leq 2\|A^\dagger\|_2\|e\|_2 \quad (3.2.12)$$

A grande dificuldade deste método é que precisamos de uma estimativa para a norma do erro e . Caso essa estimativa seja muito grande, podemos encontrar um parâmetro de regularização muito grande causando os inconvenientes já discutidos. O mesmo acontecendo com uma estimativa menor o que, neste caso, pode ser mais indesejável pelo fato dos erros serem dominantes na construção da solução.

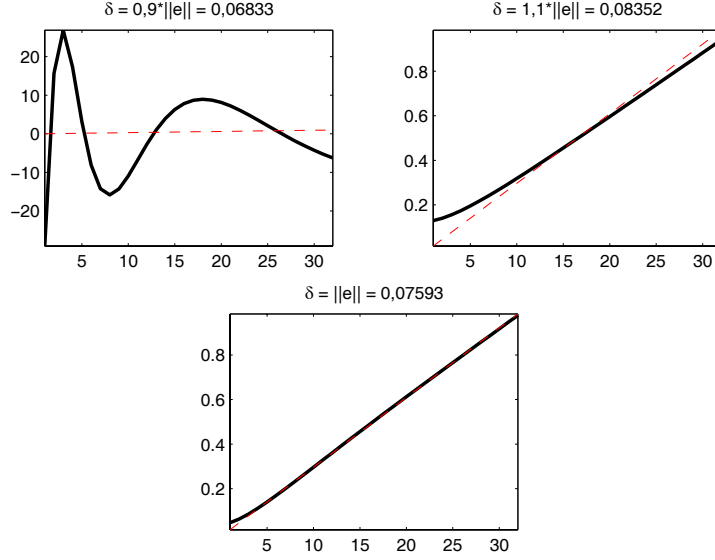


Figura 3.2: Três soluções encontradas pelo princípio da discrepância para diferentes estimativas para a norma do erro e , $\|e\|_2$.

Na Figura 3.2 temos três reconstruções para a solução do problema teste *foxgood* [34] com dimensão 32 e erro relativo nos dados de 3%. No último

gráfico da figura 3.2, 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%.

3.2.3 Curva-L

Usando a SVD da matriz A vimos que a solução e o resíduo para a regularização de Tikhonov são dados pelas equações (2.2.26) e (2.2.28). Com essas expressões as respectivas normas são obtidas por

$$\|x_\lambda\|_2^2 = \sum_{i=1}^r \left(f_i \frac{u_i^T b}{\sigma_i} \right)^2, \quad (3.2.13)$$

$$\|r_\lambda\|_2^2 = \sum_{i=1}^r ((1 - f_i) u_i^T b)^2 + \|b_\perp\|_2^2. \quad (3.2.14)$$

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

$$\mathcal{L}(\lambda) = \{(a, b)/a = \log(\mathbf{x}(\lambda)), b = \log(\mathbf{y}(\lambda))\}. \quad (3.2.15)$$

O método apareceu em 1992 e foi introduzido por Hansen [31]; ele 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 (2.2.27), e que o número de fatores de filtro próximos da unidade seja k . Segue de (3.2.13) que

$$\|x_\lambda^{\text{exato}}\|_2^2 \cong \sum_{i=1}^k (v_i^T x^{\text{exato}})^2 \cong \sum_{i=1}^r (v_i^T x^{\text{exato}})^2 = \|x^{\text{exato}}\|_2^2 \quad (3.2.16)$$

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 (3.2.16) vale enquanto λ não é muito grande.

Podemos perceber que quando $\lambda \rightarrow \infty$ (e $k \rightarrow 0$) temos $x_\lambda^{\text{exato}} \rightarrow 0$. Por outro lado, quando $\lambda \rightarrow 0$ temos $x_\lambda^{\text{exato}} \rightarrow x^{\text{exato}}$.

O resíduo correspondente a x_λ^{exato} satisfaz

$$\|b - Ax_\lambda^{\text{exato}}\|_2^2 \cong \sum_{i=k+1}^r (u_i^T b^{\text{exato}})^2 \quad (3.2.17)$$

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

$$|u_i^T e| \cong \varepsilon, \quad i = 1, \dots, m \quad (3.2.18)$$

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

$$\begin{aligned} \|x_\lambda^e\|_2^2 &\cong \sum_{i=1}^r \left(\frac{\sigma_i \varepsilon}{\sigma_i^2 + \lambda^2} \right)^2 \cong \sum_{i=1}^k \left(\frac{\varepsilon}{\sigma_i} \right)^2 + \sum_{i=k+1}^r \left(\frac{\sigma_i \varepsilon}{\lambda^2} \right)^2 \\ &= \varepsilon^2 \left(\sum_{i=1}^k \sigma_i^{-2} + \lambda^{-4} \sum_{i=k+1}^r \sigma_i^2 \right) \end{aligned} \quad (3.2.19)$$

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

$$\|x_\lambda^e\|_2 \cong c_\lambda \frac{\varepsilon}{\lambda} \quad (3.2.20)$$

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

$$\|b - Ax_\lambda^e\|_2^2 \cong \sum_{i=k}^m \varepsilon^2 = (m - k) \varepsilon^2. \quad (3.2.21)$$

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. Então 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 3.3. A região da curva indicada com o termo “Menos filtro” refere-se a valores de λ que resultam em fatores de filtro f_i próximos de 1. Neste caso, a contribuição do erro não é filtrada o suficiente para garantir uma solução regularizada apropriada. Uma interpretação análoga vale para a região indicada com o termo “Mais filtro”.

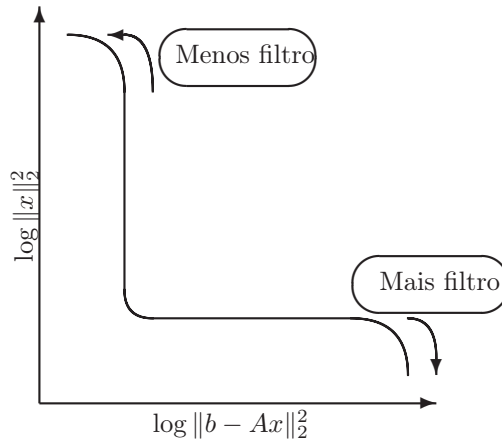


Figura 3.3: Curva-L genérica.

Baseado nessas considerações, Hansen [31] propôs escolher o parâmetro de regularização λ localizando a região em que se encontra a transição entre as duas curvas. Mais precisamente, a proposta do método é escolher o parâmetro λ que maximiza a curvatura da curva $\mathcal{L}(\lambda)$, a qual é definida por

$$\kappa = \rho\eta \frac{\rho\eta/|\eta'| - \lambda^2 - \lambda^4\eta}{(\rho^2 + \lambda^4\eta^2)^{3/2}}, \quad (3.2.22)$$

em que $\rho = \|b - Ax_\lambda\|_2$, $\eta = \|x_\lambda\|_2$, e $|\eta'|$ é dada por

$$|\eta'| = \sum_{i=1}^r \frac{2\sigma_i}{(\sigma_i^2 + \lambda^2)^3} (u_i^T b)^2. \quad (3.2.23)$$

O critério curva-L como método para a determinação do parâmetro de regularização de Tikhonov apresenta duas limitações. A primeira refere-se à reconstrução de soluções exatas muito suaves, isto é, soluções x^{exato} tal que os coeficientes da SVD $|v_i^T x^{\text{exato}}|$ decaem rápido para zero tal que a solução x^{exato} seja dominada pelas primeiras componentes da SVD. Para situações como estas o parâmetro encontrado pelo critério da curva-L normalmente é consideravelmente menor do que o parâmetro ótimo, entendemos por parâmetro ótimo como sendo aquele que $\|x_\lambda - x^{\text{exato}}\|_2 / \|x^{\text{exato}}\|_2$ seja o menor possível, e, quanto mais suave for a solução, pior o parâmetro. Detalhes a respeito das limitações da curva-L são apontados em [27, 59, 32].

Uma das questões com relação à curva-L é conhecer em que regiões ela é convexa e em que regiões ela é côncava. Hansen [32] e Regińska [54] deram os primeiros passos na tentativa de elucidar esta questão, porém, apenas algumas considerações foram desenvolvidas e uma resposta definitiva até então não existia. 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 em Bazán e Francisco [4].

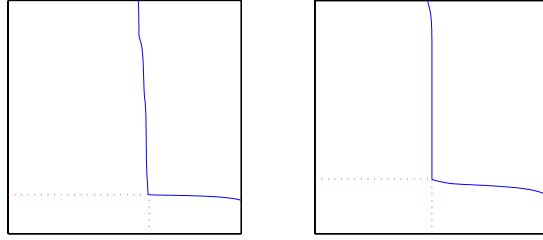


Figura 3.4: Curva-L para os problemas *foxgood* e *heat* com 2% de erros relativos nos dados.

Por fim, na Figura 3.4 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 $n = 32$.

3.2.4 Ponto-Fixo (FP)

O algoritmo de ponto-fixo foi proposto por Bazán [3] baseado num trabalho anterior de Regińska [54]. O algoritmo também utiliza informação da norma da solução regularizada e da norma do resíduo correspondente, e não requer

qualquer tipo de informação do tamanho de ruído nos dados. Um resultado preliminar envolvendo essas quantidades é que, se $\mathbf{y}(\lambda) = \|x_\lambda\|_2^2$, $\mathbf{x}(\lambda) = \|b - Ax_\lambda\|_2^2$, então

$$d\mathbf{y}/d\mathbf{x} = -1/\lambda^2 \quad (3.2.24)$$

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

$$\Psi_\mu(\lambda) = \mathbf{x}(\lambda)\mathbf{y}^\mu(\lambda), \quad \mu > 0. \quad (3.2.25)$$

Derivando (3.2.25) com respeito a λ temos

$$\Psi'_\mu(\lambda) = \mathbf{y}(\lambda)^\mu \mathbf{y}'(\lambda) \left[\mu \frac{\mathbf{x}(\lambda)}{\mathbf{y}(\lambda)} + \frac{\mathbf{x}'(\lambda)}{\mathbf{y}'(\lambda)} \right] \quad (3.2.26)$$

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

$$\lambda^{*2} = \mu \frac{\mathbf{x}(\lambda^*)}{\mathbf{y}(\lambda^*)} \Leftrightarrow \lambda^* = \sqrt{\mu} \frac{\eta(\lambda^*)}{\xi(\lambda^*)} \quad (3.2.27)$$

em que $\eta(\lambda) = \sqrt{\mathbf{x}(\lambda)}$ e $\xi(\lambda) = \sqrt{\mathbf{y}(\lambda)}$. Cosequentemente, se λ^* é um minimizador de $\Psi_\mu(\lambda)$ então λ^* deve ser um ponto fixo de $\phi_\mu : \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$ definida por

$$\phi_\mu(\lambda) = \sqrt{\mu} \frac{\|b - Ax_\lambda\|_2}{\|x_\lambda\|_2}. \quad (3.2.28)$$

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

$$\lambda_{k+1} = \phi_\mu(\lambda_k). \quad (3.2.29)$$

Na Figura (3.5) mostramos 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.

A convergência da sequência (3.2.29) é uma consequência do fato da função $\phi_\mu(\lambda)$ ser estritamente crescente, o que decorre de propriedade (3.2.24), enquanto a existência de ponto fixo da função $\phi_\mu(\lambda)$ depende do parâmetro μ (na maioria dos problemas $\mu = 1$ é suficiente para garantir tal existência). Para critérios de parada ou ajustes no μ o leitor é recomendado a ler [3].

Uma pequena dificuldade com o FP é sua forte dependência do *chute inicial*: uma escolha errada pode levar a ponto fixo de $\phi_\mu(\lambda)$ que maximiza $\tilde{\Psi}_\mu$. Para contornar possíveis dificuldades provenientes da presença de vários pontos fixos, Bazán e Francisco introduziram o conceito de ponto fixo convexo e o leitor deve-se referir a [4].

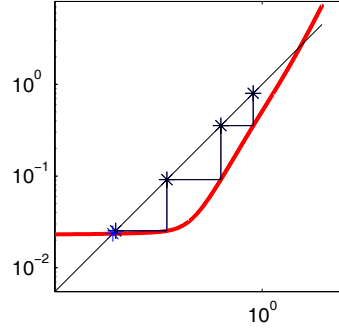


Figura 3.5: Iterações λ_k para o problema teste *shaw*.

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

O algoritmo LANC-FP foi introduzido recentemente por Bazán e Borges [5] com intuito de implementar o algoritmo de ponto-fixo para problemas de grande porte. Objetivamente, LANC-FP determina a sequência finita $\{\lambda^{(k)*}\}_k$, $k \geq p \geq 2$, onde cada elemento da sequência é um ponto-fixo da função

$$\phi_{\mu}^{(k)}(\lambda) = \sqrt{\mu} \frac{\|\beta_1 e_1 - B_k y_{\lambda}^{(k)}\|_2}{\|y_{\lambda}^{(k)}\|_2}, \quad \mu > 0. \quad (3.2.30)$$

em que $B_k = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \beta_3 & \ddots & \\ & & \ddots & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix}$, e_1 é o primeiro vetor canônico

em \mathbb{R}^{k+1} , e y_{λ} é solução do problema regularizado projetado

$$y_{\lambda}^{(k)} = \operatorname{argmin}_{y \in \mathbb{R}^k} \left\| \begin{pmatrix} \beta_1 e_1 \\ 0 \end{pmatrix} - \begin{pmatrix} B_k \\ \lambda I_k \end{pmatrix} y \right\|_2. \quad (3.2.31)$$

Algoritmicamente, a matriz B_k (juntamente com duas matrizes $U_{k+1} \in \mathbb{R}^{m \times (k+1)}$ e $V_k \in \mathbb{R}^{n \times k}$ com colunas ortonormais) é obtida no k -ésimo passo do algoritmo de bidiagonalização de Lanczos (LBD) [8] e então o ponto-fixo λ^{k*} de $\phi_{\mu}^{(k)}(\lambda)$ é determinado. O processo prossegue até o passo em que a sequência $\{\lambda^{k*}\}_k$ estaciona e, a seguir, com o último λ da sequência

$\{\lambda^{(k)*}\}_k$, denotado aqui como λ^* , a solução regularizada é construída como

$$f_{\lambda^*}^{(k)} = V_k y_{\lambda^*}^{(k)}. \quad (3.2.32)$$

Teoricamente, $f_{\lambda^*}^{(k)}$ é solução do “damped least squares problem”

$$f_{\lambda^*}^{(k)} = \operatorname{argmin}_{x \in \mathcal{K}_k(A^T A, A^T b)} \left\| \begin{pmatrix} g \\ 0 \end{pmatrix} - \begin{pmatrix} A \\ \lambda I_n \end{pmatrix} f \right\|_2, \quad (3.2.33)$$

onde $\mathcal{K}_k(A^T A, A^T b) = \operatorname{span}\{A^T b, (A^T A)A^T b, \dots, (A^T A)^{k-1}A^T b\}$ [8]. Note-mos que resolver (3.4.47) sem restrições equivale a resolver o problema (3.1.1).

O ponto forte de LANC-FP é que a sequência $\{\lambda^{(k)*}\}_k$ converge em poucas iterações para λ^* , o maior ponto-fixo convexo da função $\phi_1(\lambda)$ associada ao problema de grande porte, sugerido em [4] como parâmetro de regularização. Segue uma breve descrição do algoritmo na tabela 3.1.

LANC-FP

Dados de Entrada: $A, b, p > 1, k_{\max}, \epsilon$.

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
 $k \leftarrow k + 1$
 Volte para passo 3.
Fim se
5. Calcular solução regularizada $x_{\lambda^*}^{(k)}$

Tabela 3.1: Esquema do algoritmo LANC-FP

O critério de parada de LANC-FP é determinado pelo primeiro k tal que

$$|\lambda^{(k+1)*} - \lambda^{(k)*}| < \epsilon_1 |\lambda^{(k)*}|, \quad (3.2.34)$$

em que ϵ_1 é um parâmetro de tolerância, ou

$$|\lambda^{(k+1)*} - \lambda^{(k)*}| < \epsilon_2 |\lambda^{(0)*}|, \quad (3.2.35)$$

em que $\lambda^{(0)*}$ é o ponto-fixo calculado no passo 2 e ϵ_2 um outro parâmetro de tolerância. O critério (3.2.35) é apropriado quando a sequência $\lambda^{(k)*}$ decresce muito devagar (tornando o processo caro).

Veremos agora alguns resultados teóricos que suportam nosso algoritmo.

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

Teorema 3.1. *Para todo $\lambda > 0$ vale $\phi_\mu^{(k+1)}(\lambda) \leq \phi_\mu^{(k)}(\lambda)$. Consequentemente, $\phi_\mu(\lambda) \leq \phi_\mu^{(k)}(\lambda)$, $k = 1, \dots, n-1$.*

Uma ilustração do teorema acima segue na figura 3.6. Na esquerda temos as funções $\phi_1^{(3)}(\lambda)$, $\phi_1^{(4)}(\lambda)$, $\phi_1^{(5)}(\lambda)$, $\phi_1^{(6)}(\lambda)$ e $\phi(\lambda)$ para o problema *heat* com dimensão $n = 256$ e erro relativo nos dados de 1%. Na direita mostramos um zoom da figura da esquerda.

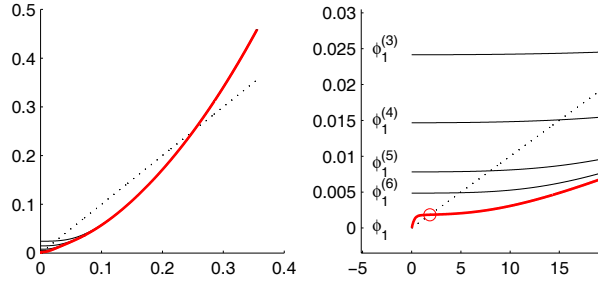


Figura 3.6: Funções $\phi_1(\lambda)$ (por baixo), $\phi_1^{(3)}(\lambda)$, $\phi_1^{(4)}(\lambda)$, $\phi_1^{(5)}(\lambda)$, $\phi_1^{(6)}(\lambda)$ e $\phi(\lambda)$.

O próximo teorema descreve o comportamento da sequência $\{\lambda^{(k)*}\}$.

Teorema 3.2. *A sequência de pontos-fixos $\lambda^{(k)*}$ é não-crescente.*

O próximo teorema descreve um dos resultados mais relevantes em torno de LANC-FP: quando as iterações estabilizam não mais existe mudança na solução calculada.

Teorema 3.3. *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, $x_{\lambda^*}^{(k)} = x_{\lambda^*}^{(k+1)} = \dots = x_{\lambda^*}^{(n)} \triangleq x_{\lambda^*}$ e, como consequência, a norma do erro $\|x^{exato} - x_{\lambda^*}^{(j)}\|_2$ permanece constante para $k \leq j \leq n$.*

Na Figura 3.7 podemos verificar o Teorema 3.3. 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 (\times) indica a iteração de parada.

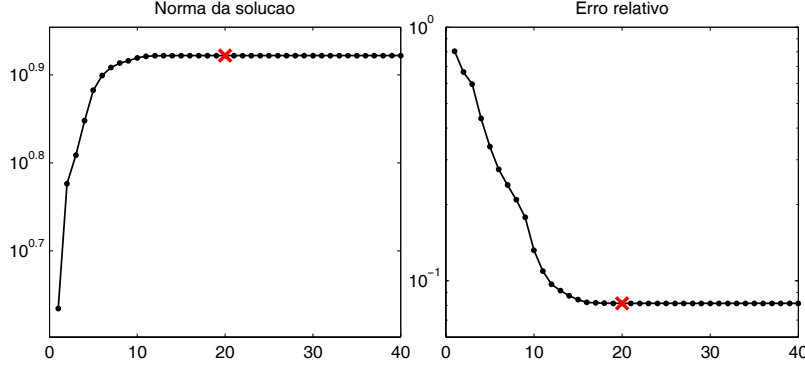


Figura 3.7: Ilustração do teorema 3.3. A cruz (\times) indica a iteração de parada.

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 $k = 20, \dots, 40$ não sofreram alterações, o mesmo ocorrendo com o erro relativo nas soluções.

3.3 Regularização iterativa

Para problemas de grande porte em que o cálculo dos valores singulares da matriz A não pode ser realizado, método iterativo são recomendáveis. A ideia básica neste caso é usar métodos iterativos clássicos tais como Gradientes conjugados, LSQR, SOR, etc, e parar as iterações quando a qualidade das iteradas começa a incorporar componentes do ruído nos dados. Assim, uma dificuldade comum a muitos métodos iterativos é determinar a iteração de parada uma vez que após atingir o número ótimo de passos, as iteradas 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 [17], o método dos Gradientes Conjugados [33] e, os métodos LSQR, GMRES,

e a técnica de Reconstrução Algébrica [17]. Outras técnicas podem ser encontradas em [9, 33].

3.3.1 Iteração de Landweber

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

$$x^{(k+1)} = x^{(k)} + \gamma^2 A^T (b - Ax^{(k)}) \quad (3.3.36)$$

em que $\gamma > 0$ é um parâmetro real e usualmente $x^{(0)} = 0$. Usando a SVD da matriz A podemos escrever

$$x^{(k)} = \sum_{i=1}^p c_i^{(k)} v_i \quad (3.3.37)$$

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

$$c_i^{(k+1)} = (1 - \gamma^2 \sigma_i^2) c_i^{(k)} + \gamma^2 \sigma_i \langle b, u_i \rangle. \quad (3.3.38)$$

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

$$c_i^{(k)} = F_k(\sigma_i) \sigma_i^{-1} \langle b, u_i \rangle \quad (3.3.39)$$

$$F_k(\sigma) = 1 - (1 - \gamma^2 \sigma^2)^{(k)} \quad (3.3.40)$$

então,

$$x^{(k)} = \sum_{i=1}^p F_k(\sigma_i) \sigma_i^{-1} \langle b, u_i \rangle v_i. \quad (3.3.41)$$

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

3.3.2 Método dos Gradientes Conjugados

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 visa minimizar $\|b - Ax\|_2$ e trabalha 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 recursivamente por

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} d^{(k)}, \quad \alpha^{(k)} = -\frac{\langle g^{(k)}, d^{(k)} \rangle}{\langle d^{(k)}, Qd^{(k)} \rangle}, \quad g^{(k)} = -A^T b + Qx^{(k)} \quad (3.3.42)$$

$$d^{(k+1)} = -g^{(k+1)} + \beta^{(k)} d^{(k)}, \quad \beta^{(k)} = \frac{\langle g^{(k+1)}, Qd^{(k)} \rangle}{\langle d^{(k)}, Qd^{(k)} \rangle}. \quad (3.3.43)$$

Notemos que o método CG forma explicitamente a matriz $A^T A$, logo ele deve ser evitado se existirem valores singulares do tamanho $10^{-t/2}$ em que t é o número de casas decimais que o computador consegue trabalhar. Uma versão do GC que evita a formação explícita do produto $A^T A$ é conhecida como CGLS. Uma descrição do método bem como de suas propriedades regularizantes pode ser encontradas em [9, 33].

3.3.3 LSQR e GMRES

Resumidamente, a ideia do método LSQR (least squares QR) [51] é construir em cada iteração uma solução aproximada x_k através da resolução do subproblema

$$x_k = \operatorname{argmin}_{x \in \mathcal{K}_k(A^T A, A^T b)} \|b - Ax\|_2,$$

onde $\mathcal{K}_k(A^T A, A^T b) = \operatorname{span}\{A^T b, (A^T A)A^T b, \dots, (A^T A)^{k-1}A^T b\}$. Na prática a minimização acima é substituída por um processo recursivo baseado na bidiagonalização de Lanczos e a fatoração QR.

O método GMRES (método dos resíduos mínimos generalizados) [57] constrói soluções aproximadas x_k para a solução de sistemas lineares $Ax = b$, com $A \in \mathbb{R}^{n \times n}$ não singular, resolvendo o subproblema

$$x_k = \operatorname{argmin}_{x \in \mathcal{K}_k(A, b)} \|b - Ax\|_2.$$

Análogo a LSQR, GMRES constrói x_k recursivamente usando uma base ortonormal calculada pelo método de Arnoldi e a fatoração QR de certa matriz de Hessenberg. Detalhes das propriedades regularizantes de GMRES são encontradas em [11].

3.3.4 Técnica de Reconstrução Algébrica

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

$$\langle a_j, x \rangle = b_j, \quad j = 1, \dots, m, \quad \|a_j\| = 1 \quad (3.3.44)$$

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

$$P_j x = x + (b_j - \langle a_j, x \rangle) a_j \quad (3.3.45)$$

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

$$P_j^w = (1 - w)I + wP_j, \quad P^w = P_m^w \cdots P_1^w. \quad (3.3.46)$$

Então, as iterações ART são $x^{(k+1)} = P^w x^{(k)}$. 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 $A^T Ax = A^T b$.

3.4 Exercícios

1. Seja ϕ_μ a função de iteração do método de ponto fixo definido em (3.2.29). Demonstre que $\phi'_\mu(\lambda) > 0 \quad \forall \lambda > 0$ e conclua que ϕ_μ estritamente crescente. Use esse fato para provar que dependendo da aproximação inicial, a iteração $\lambda_{k+1} = \phi_\mu(\lambda_k)$, $k \geq 0$, gera uma sequência crescente ou decrescente.
2. Considere a solução do “damped least squares problem”

$$f_\lambda^{(k)} = \underset{x \in \mathcal{K}_k(A^T A, A^T b)}{\operatorname{argmin}} \left\| \begin{pmatrix} g \\ 0 \end{pmatrix} - \begin{pmatrix} A \\ \lambda I_n \end{pmatrix} f \right\|_2, \quad (3.4.47)$$

onde $\mathcal{K}_k(A^T A, A^T b)$ é o subespaço de Krylov definido na seção 3.2, e seja $r_\lambda^{(k)} = g - A f_\lambda^{(k)}$ o vetor residual correspondente. Prove que

$$\|f_\lambda^{(k+1)}\|_2 \geq \|f_\lambda^{(k)}\|_2, \quad \|r_\lambda^{(k+1)}\|_2 \leq \|r_\lambda^{(k)}\|_2.$$

3. Prove que o projetor P_j definido no método ART satisfaz (3.3.45).
4. Reescreva a iteração de Landweber na forma $x^{(k+1)} = R x^{(k)} + c$, $k \geq 0$. Analise o raio espectral da matriz R e conclua que as iterações convergem para $A^\dagger b$ quando $\gamma \sigma_1 < 1$.

Capítulo 4

Uso de Toolboxes na solução de Problemas Inversos

Com intuito de divulgar ferramentas computacionais de uso público, descrevemos neste capítulo duas *toolboxes* que têm sido frequentemente utilizadas seja na resolução de problemas inversos ou como fonte de problemas teste para avaliar novos algoritmos. A estrutura, funcionamento e avaliação preliminar do algoritmo LANC-FP, quando utilizado na resolução de problemas teste da literatura, também são incluídos.

4.1 RegularizationTools: um tutorial

A toolbox RegularizationTools, devida a Hansen [34], é um conjunto de rotinas para o Matlab destinadas ao tratamento de problemas discretos mal-postos (criação, análise e resolução), que inclui vários problemas teste com uma rotina para cada um deles. A maioria dos problemas teste, descritos na Tabela 4.1, provém da discretização de equações integrais de Fredholm de primeira espécie e, de modo geral, “funcionam” da mesma maneira: o usuário fornece o tamanho do sistema desejado, recebendo como variáveis de saída a matriz A , o vetor b “exato” e a solução “exata” do problema no vetor x . Nem todos os problemas têm solução e algumas rotinas possuem mais de um parâmetro de entrada. Para mais detalhes a respeito das rotinas basta usar `help "nome da rotina"`.

Como um exemplo da criação, análise e solução de um problema teste, vamos usar a rotina `shaw` com $n = 64$ e vetor $b = b^{\text{exato}} + e$, com e contendo números aleatórios Gaussianos com média zero e $\|e\|_2 / \|b^{\text{exato}}\|_2 = 0.01$. Os

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

Tabela 4.1: Problemas teste

dados podem ser gerados da seguinte maneira

```
>> [A,b_true,x_true] = shaw(64);
>> e = randn(size(b_true));
>> b = b_true + e/norm(e)*norm(b_true)*0.01;
```

Dentre as rotinas para análise de problema mal-postos discretos disponíveis na toolbox, as mais relevantes são encontradas na Tabela 4.2

Rotina	Descrição
corner	Localiza o canto da curva-L discreta.
fil_fac	Calcula os fatores de filtro para alguns métodos de regularização.
gcv	Plota a função GCV e calcula o mínimo.
l_corner	Localiza o “canto” da curva-L.
l_curve	Calcula a curva-L, plota a curva e encontra o “canto”.
picard	Plota os valores singulares, os coeficientes de Fourier e a razão entre eles.
plot_lc	Plota a curva-L.

Tabela 4.2: Análise

Vamos ilustrar o uso da rotina **picard** na verificação da condição discreta de Picard (CDP) para o problema **shaw** usando os dados gerados acima. A

rotina `picard` requer o cálculo dos valores singulares da matriz A . Ambas as tarefas podem ser feitas da seguinte forma:

```
>> [U,s,V]=csvd(A);
>> figure(1)
>> subplot(1,2,1),picard(U,s,b);xlabel('CDP com b perturbado');
>> subplot(1,2,2),picard(U,s,b_true);xlabel('CDP com b exato');
```

Observamos que a rotina `csvd` produz exatamente o mesmo que a sequência `[U,s,V]=svd(A,0);s=diag(s)`; no Matlab, mas aqui s é um vetor contendo os valores singulares, não uma matriz. Os resultados da sequência acima são mostrados na Figura 4.1. Como os resultados sugerem que a CDP é satisfeita pelo problema com os dados limpos, conclui-se que poderemos encontrar uma solução aceitável por métodos de regularização, apesar da CDP não ser satisfeita pelo problema com dados perturbados.

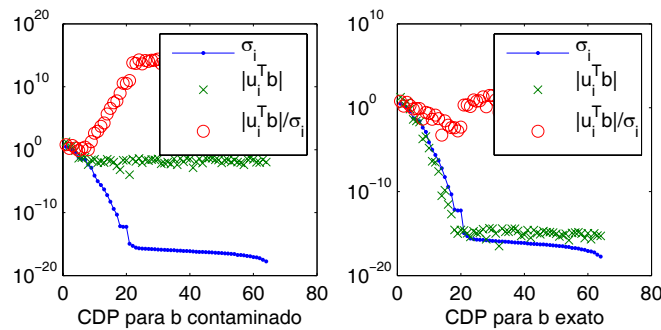


Figura 4.1: Inspeção visual da condição discreta de Picard.

Veremos agora o uso das rotinas `gcv` e `l_curve` na determinação do parâmetro de regularização. Após a execução da sequência de comandos:

```
>> figure(2);
>> subplot(1,2,1);
>> [reg_min,G,reg_param] = gcv(U,s,b,'Tikh');
>> subplot(1,2,2);
>> [reg_corner,rho,eta,reg_param] = l_curve(U,s,b,'Tikh');
```

podemos conferir na Figura 4.2 que os métodos proporcionaram parâmetros de regularização da mesma ordem de grandeza.

Na sequência acima podemos substituir `'Tikh'` por `'tsvd'` ou outras opções encontradas no help de cada rotina. A Figura 4.3 mostra o resultado dos mesmos comandos porém usando a opção `'tsvd'`.

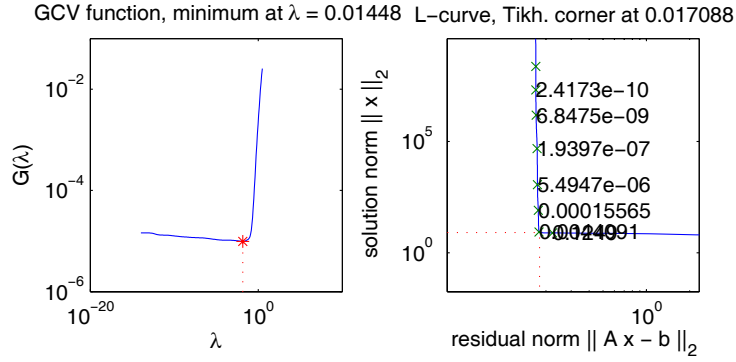


Figura 4.2: Função GCV e Curva-L para Tikhonov.

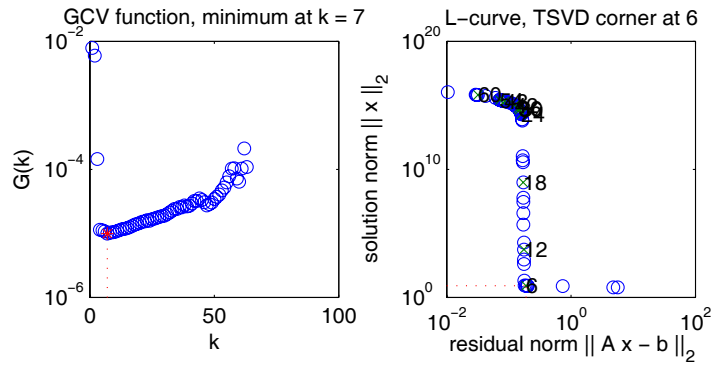


Figura 4.3: Função GCV e Curva-L para a SVD Truncada.

Por fim, o cálculo solução regularizada pode ser feito por meio de diversas rotinas: um resumo das principais opções é mostrado na tabela 4.3.

Rotina	Descrição
discrep	Princípio da discrepância.
tgsvd	Solução via GSVD Truncada.
tikhonov	Solução via regularização de Tikhonov.
tsvd	Solução via SVD Truncada.

Tabela 4.3: Rotinas baseadas na SVD.

A sequência de comandos a seguir calcula as soluções regularizadas construídas pelo método de Tikhonov usando os parâmetros de regularização do método da discrepância (usando a norma $\|e\|_2$ como dado de entrada) e curva-L, e a solução construída pelo método da SVD truncada. Os resulta-

dos são mostrados na figura 4.4.

```
>> figure(4);
>> [x_delta,la] = discrep(U,s,V,b,norm(b-b_true));
>> [x_la,rho,eta] = tikhonov(U,s,V,b,[reg_min reg_corner] );
>> [x_k,rho,eta] = tsvd(U,s,V,b,[reg_min_tsvd reg_corner_tsvd]);
>> subplot(1,3,1);plot(1:n,x_delta,1:n,x_true,'k:');axis tight;
>> subplot(1,3,2);plot(1:n,x_la(:,1),1:n,x_la(:,2),1:n,x_true,'k:');
>> axis tight
>> subplot(1,3,3);plot(1:n,x_k(:,1),1:n,x_k(:,2),1:n,x_true,'k:');
>> axis tight
```

Tanto a rotina `tikhonov` quanto a `tsvd` necessitam de um parâmetro de regularização que pode ser escalar ou vetor; no segundo caso, como no exemplo anterior, a primeira variável de saída é uma matriz em que cada coluna é a solução associada a cada elemento do vetor de parâmetros.

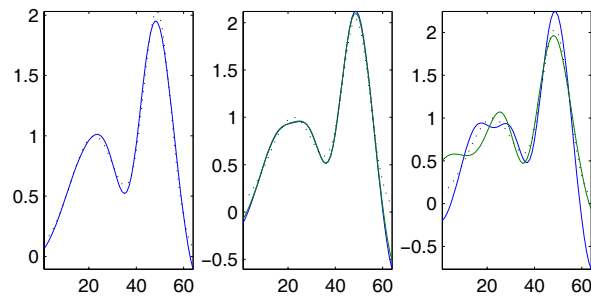


Figura 4.4: Esquerda: Princípio da discrepância. Centro: curva-L e GCV. Direita: TSVD.

A toolbox também inclui técnicas de regularização iterativa. Algumas rotinas clássicas são listadas Tabela 4.4. Convém ressaltar que algumas delas possuem versões para a forma geral de regularização de Tikhonov, como `pcgls` para os Gradientes Conjugados e `plsqr_b` para o LSQR.

Rotina	Descrição
<code>art</code>	Técnica de reconstrução algébrica (método de Kaczmarz).
<code>cgl</code>	Método dos Gradientes Conjugados.
<code>lsqr_b</code>	Método LSQR.
<code>maxent</code>	Calcula a solução regularizada de máxima entropia.

Tabela 4.4: Rotinas para regularização iterativa.

Para os métodos iterativos funcionarem como métodos de regularização, o número de iterações deve ser inserido como um parâmetro de entrada (exceto para `maxent`). Um outro parâmetro que pode ser escolhido pelo usuário serve para decidir pelo uso ou não de reortogonalização: é bem conhecido que na prática a ortogonalidade deteriora à medida que as iterações avançam. Mais detalhes podem ser encontrados no comando de ajuda às rotinas.

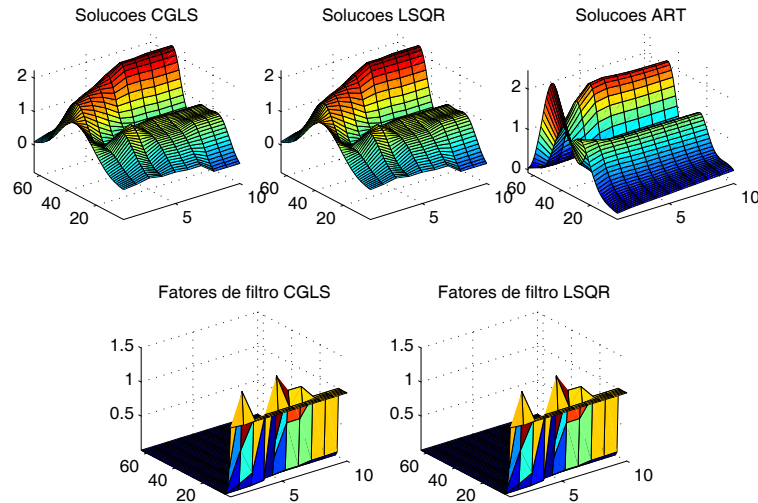


Figura 4.5: Parte superior: Soluções iteradas para CGLS, LSQR e ART. Parte inferior: Fatores de filtro para CGLS e LSQR.

Na Figura 4.5 podemos observar o comportamento das 10 primeiras soluções iteradas dos métodos CGLS, LSQR e ART bem como os fatores de filtro associados a cada uma das iteradas de CGLS e LSQR. Os algoritmos CGLS e LSQR apresentam praticamente as mesmas soluções, e o mesmo vale para o ART após um certo número de iterações. O leitor pode verificar que embora a sequência de iteradas parece estabilizar, as subseqüentes iteradas serão dominadas pelo erro contido nos dados fazendo com que essas soluções sejam de pouca utilidade. A figura é obtida pela sequência de comandos:

```
>> figure(5);k=10;
>> [X_c,rho_c,eta_c,F_c] = cglsl(A,b,k,0,s);
>> [X_l,rho_l,eta_l,F_l] = lsqr_b(A,b,k,0,s);
>> [X_a,rho_a,eta_a] = art(A,b,k);
```

```
>> subplot('position',[0.08 0.6 0.25 0.3]);
>> surf(X_c);title('Solucoes CGLS');axis tight
>> subplot('position',[0.38 0.6 0.25 0.3])
>> surf(X_l);title('Solucoes LSQR');axis tight
>> subplot('position',[0.68 0.6 0.25 0.3])
>> surf(X_a);title('Solucoes ART');axis tight
>> subplot('position',[0.21 0.15 0.25 0.3])
>> surf(F_c);title('Fatores de filtro CGLS');axis tight
>> subplot('position',[0.55 0.15 0.25 0.3])
>> surf(F_l);title('Fatores de filtro LSQR');axis tight
```

Detalhes adicionais do uso e a versatilidade da toolbox podem ser encontrados no manual. Uma limitação da toolbox é o fato da maioria das rotinas funcionarem bem quando aplicadas a problemas de pequena dimensão para os quais a SVD da matriz A pode ser calculada.

4.2 RestoreTools

RestoreTools é uma *toolbox* desenvolvida por J. G. Nagy e colaboradores [49] criada especialmente para tratar problemas de restauração de imagem usando Matlab. Basicamente, a *toolbox* contém problemas teste e métodos iterativos dentre os quais destaca-se o algoritmo chamado HyBR. Neste algoritmo o usuário pode escolher o método WGCV (Weighted-GCV) como critério de parada [15]. A *toolbox* de processamento de imagens do MatLab contém alguns métodos para restauração de imagens, mas com limitações. Já a *toolbox* RestoreTools, segue a linha de programação orientada a objetos, o que facilita o cálculo de produtos matriz-vetor que é a base de qualquer método iterativo para problemas de grande porte.

Assim como a toolbox RegularizationTools, todas as rotinas aqui encontradas possuem um *help* que pode ser usado quando necessário. Uma sucinta descrição dos problemas teste pode ser conferida na Tabela 4.5

Rotina	Descrição
satellite	Imagem de um satelite.
Grain	Imagem de grãos.
Text/Text2	Identificação do conteúdo de texto.
star_cluster	Detecção de um grupo de estrelas.

Tabela 4.5: Problemas testes - RestoreTools

Estas rotinas na verdade são arquivos para o Matlab que contém as informações do problema. Vamos ilustrar o uso da toolbox com um exemplo

usando o problema satellite carregando o arquivo correspondente:

```
>> load satellite
>> whos
```

Name	Size	Bytes	Class	Attributes
PSF	256x256	524288	double	
b	256x256	524288	double	
x_true	256x256	524288	double	

A variável PSF é uma imagem chamada de *point spread function* que modela o efeito de espalhamento de um ponto de luz quando fotografado (a visualização desta imagem pode ser feita através do comando `imshow(PSF)`), enquanto as variáveis b e x_true representam a imagem ruim (imagem embaçada e com ruído, ver Figura 4.6) e a imagem exata, respectivamente.

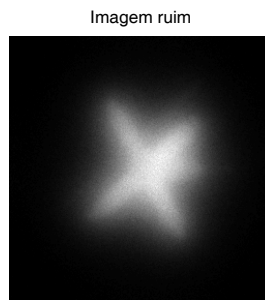


Figura 4.6: Imagem ruim de um satélite.

A PSF serve para gerar a matriz do sistema linear através da rotina `psfMatrix`:

```
>> A = psfMatrix(PSF);
```

enquanto o lado direito do sistema é um vetor \hat{b} obtido concatenando as colunas da matriz b . Observamos que a matriz A não é do tipo comum (tipo double). Ela é um objeto da classe `psfMatrix`, implicando com isso que a matriz é estruturada e que as adições e multiplicações matriz-vetor são feitas eficientemente levando este fato em consideração. Maiores detalhes sobre o significado da PSF e da associação imagem-matriz-vetor podem ser encontrados em [49] e no capítulo seguinte.

O próximo passo é resolver o sistema linear $Ax = \hat{b}$ que neste caso é um problema de tamanho 65.536×65.536 . A toolbox se concentra na utilização de métodos iterativos uma vez que problemas de restauração de imagens

geram sistemas lineares de grande porte que inviabiliza a SVD. Um resumo das opções que a toolbox disponibiliza segue na Tabela 4.6

Rotina	Descrição
cgl	Gradientes conjugados.
mr2	Modificação do “Método dos Resíduos Mínimos” para sistemas simétricos, possivelmente indefinidos.
mrnsd	Método de descida modificado.
HyBR	Método híbrido baseado na bidiagonalização de Lanczos.
Versões preconditionadas	
pcgl	cgl com preconditionador.
pmr2	mr2 com preconditionador.
pmrnsd	mrnsd com preconditionador.

Tabela 4.6: Solução de problemas - RestoreTools

A seguir geramos um preconditionador usando a rotina `psfPrec` (que exige a matriz A e o vetor b) e utilizamos a rotina `pcgl` para calcular uma solução aproximada usando a aproximação inicial $x_0 = b/\|b\|_2$ e escolhendo o número máximo de iterações como $k = 100$.

```
>> P = psfPrec(A,b);
>> [x,k,Rnrm,Xnrm,Enrm] = pcgl(A,P,b,b/norm(b),100);
```

O método dos Gradientes Conjugados neste problema não é muito eficaz: quatro soluções iteradas mostradas na Figura 4.7 mostram que nenhuma delas pode ser aproveitada.

O grande atrativo desta toolbox está na rotina `HyBR` que implementa um algoritmo híbrido baseado na bidiagonalização de Lanczos e uma variação do método GCV para a determinação do parâmetro de regularização de Tikhonov, detalhes podem ser encontrados em [15].

A rotina `HyBR` está repleta de opções que podem ser conferidas no help da mesma. A maneira mais fácil de utilizar a rotina é informando a matriz A , o vetor b e o preconditionador P caso exista:

```
>> [x_out, output] = HyBR(A, b, P);
```

Executando o código acima teremos uma solução de qualidade visivelmente melhor do que as obtidas anteriormente por `pcgl`. Na Figura 4.8 temos a solução encontrada em 10 iterações.

Vejamos um outro exemplo usando esta toolbox. Agora usamos uma imagem que descreve o conteúdo de um texto embaçado. Tanto o problema teste `Text` quanto `Text2` podem ser utilizados e escolhemos aqui a primeira

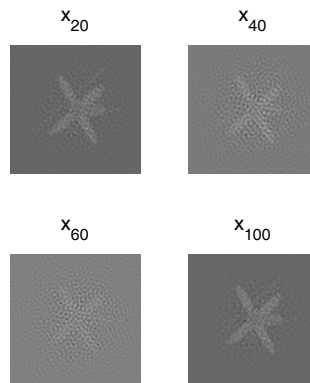


Figura 4.7: Quatro soluções iteradas do método dos Gradientes Conjugados utilizando um preconditionador.

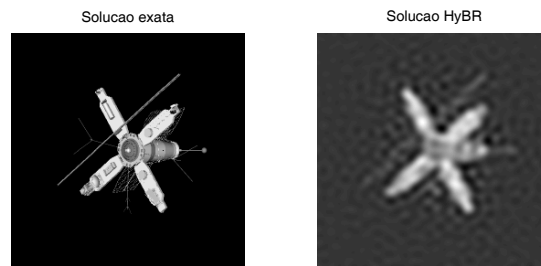


Figura 4.8: Solução exata (esquerda) e solução obtida por HyBR (direita).

opção. O cálculo de duas soluções, uma com preconditionador e outra sem preconditionador, pode ser feita através da seguinte sequência de comandos:

```
>> clear all,close all
>> load Text
>> A = psfMatrix(PSF,'zero');
>> P = psfPrec(A,b);
>> [x_out, output] = HyBR(A, b);
>> [x_out_p, output] = HyBR(A, b, P);
>> subplot(2,2,1)
>> imshow(b),title('Imagem ruim')
>> subplot(2,2,2)
>> imshow(x_true),title('Solucao exata')
```

```
>> subplot(2,2,3)
>> imshow(x_out),title('Solucao sem prec.')
>> subplot(2,2,4)
>> imshow(x_out_p),title('Solucao com prec.')
```

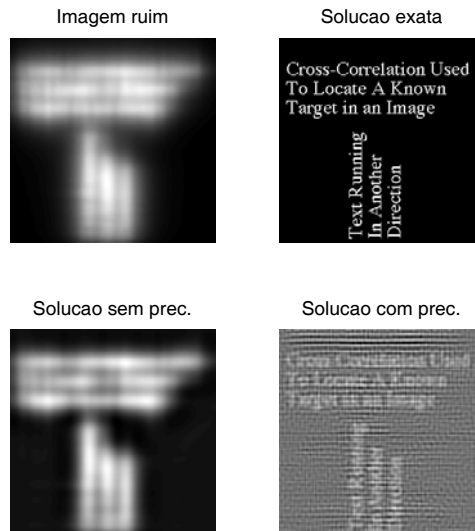


Figura 4.9: Imagem ruim, solução exata, solução HyBR sem preconditionador e solução HyBR com preconditionador.

Os resultados na Figura 4.9 mostram que a utilização de um preconditionador é capaz de deixar o texto levemente legível e que o problema de restauração é complicado de ser resolvido uma vez que além do grande número de variáveis existe o fato do problema ser mal-condicionado.

4.3 FP-tools: alguns avanços

FP-tools é uma *toolbox* em desenvolvimento que visa explorar as idéias do método de ponto-fixo proposto recentemente por Bazán [3] para a determinação do parâmetro de Tikhonov e a construção da solução regularizada correspondente. Objetivamente, até o momento são disponíveis um conjunto de rotinas que servem como ferramentas de análise das regiões de convexidade/concavidade da curva-L em escala logarítmica e que implementam o método de ponto-fixo. Segue na Tabela 4.7 as rotinas disponíveis.

Rotina	Descrição
phi_curve	Calcula e plota a curva $\phi_1(\lambda)$.
dphi_curve	Calcula $\phi'_1(\lambda)$.
fp_auto	Determina o ponto-fixo de ϕ_μ com escolha automática do parâmetro μ
lanc_fp	Constrói o problema projetado e implementa o critério de parada para sequência de pontos-fixos $\lambda^{(k)*}$
lambda_k	Avalia $\phi_\mu^{(k)}$ e determina ponto-fixo do problema projetado
lsqr_fix	Determina a solução regularizada correspondente ao ponto fixo λ^* usando LSQR

Tabela 4.7: Rotinas da Toolbox FP-tools

As três primeiras rotinas são baseadas na SVD da matriz (e portanto apropriadas para problemas de pequeno porte) e as duas primeiras servem como ferramentas de análise. Por exemplo, enquanto a primeira delas pode ser usada para saber se a curva ϕ_1 possui ou não ponto-fixo, a segunda pode ser usada para a determinação das regiões de concavidade/convexidade da curva-L, detalhes podem ser encontrados em [4]. Na Figura 4.10 podemos conferir o comportamento das distintas regiões de convexidade da curva-L.

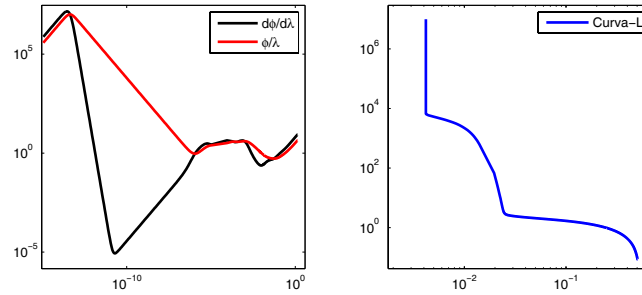


Figura 4.10: Regiões de convexidade da curva-L (os intervalos nos quais vale $\phi_1(\lambda)/\lambda \geq \phi'_1(\lambda)$) e Curva-L.

As últimas três rotinas implementam o algoritmo LANC-FP para o caso em que trabalhamos com regularização de Tikhonov na forma padrão, isto é, quando $L = I$. A implementação dos resultados para uma matriz $L \neq I$ está em fase de estudos e em breve estará disponível.

Por fim, segue uma síntese de como o algoritmo LANC-FP trabalha.

- Projetar o problema original num subespaço de Krylov de dimensão p .
- Obter um ponto-fixo da função $\phi_1(\lambda)^{(p)}$ associada ao problema projetado.
- Caso o ponto-fixo obtido não satisfaça o critério de parada, aumentar a dimensão do subespaço e obter novo ponto-fixo. Seguir o procedimento até que a sequência de pontos-fixos obtidos estabilize.

4.4 LANC-FP em ação

Apresentamos nesta secção um conjunto de resultados numéricos obtidos por LANC-FP quando aplicado aos problemas teste da toolbox RegularizationTools *foxgood*, *heat*, *wing*, *shaw*, *baart*, *deriv2*, *gravity* e *phillips*. Começamos com uma sucinta descrição de cada um deles incluindo uma análise da condição discreta de Picard. Para a análise da CDP usamos os seguintes símbolos:

- (\circ) : coeficientes $|u_i^T b|/\sigma_i$,
- (\times) : coeficientes,
- (\cdot) : valores singulares.

Foxgood

Este é um problema considerado severamente mal-posto [1], com a propriedade de não satisfazer a condição discreta de Picard na faixa dos valores singulares pequenos, mesmo no caso do vetor b ser “exato”, ver Figura 4.11.

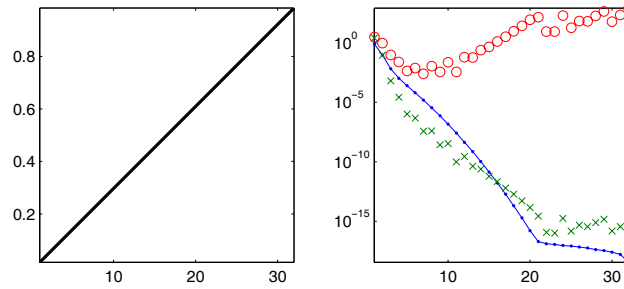


Figura 4.11: *Foxgood*: Solução exata e avaliação da CDP.

Heat

Este problema envolve uma equação integral de Volterra de primeira espécie, com $[0, 1]$ como intervalo de integração e núcleo $K(x, y) = k(x - y)$ em que

$$k(z) = \frac{z^{-\frac{3}{2}}}{2\kappa\sqrt{\pi}} e^{-\frac{1}{4\kappa^2 z}} \quad (4.4.1)$$

e κ controla o mal-condicionamento da matriz A do problema discretizado. Na Figura 4.12 temos a solução exata na esquerda e na direita os coeficientes para a inspeção visual da condição discreta de Picard.

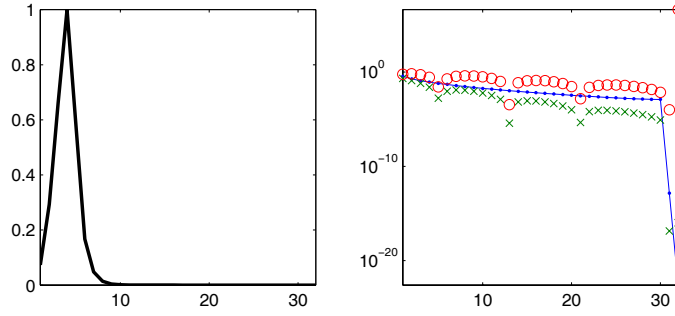


Figura 4.12: *Heat*: Solução exata e avaliação da CDP.

Wing

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

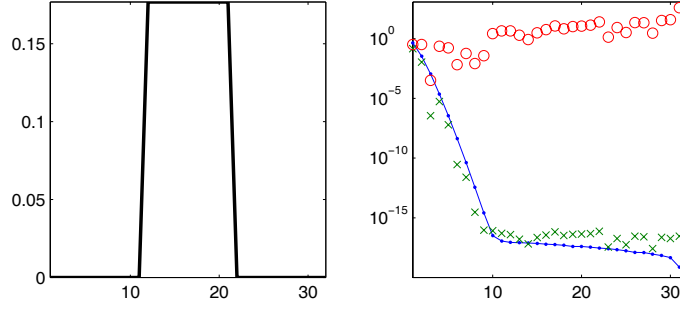
$$K(x, y) = ye^{-xy^2}, \quad 0 \leq x, y \leq 1 \quad (4.4.2)$$

$$g(x) = \frac{e^{-xt_1^2} - e^{-xt_2^2}}{2x} \quad (4.4.3)$$

e as constantes t_1 e t_2 satisfazem $t_1 < t_2$. A solução f é dada por

$$f(y) = \begin{cases} 1 & \text{para } t_1 < y < t_2 \\ 0 & \text{para } y \notin (t_1, t_2) \end{cases} \quad (4.4.4)$$

Na Figura 4.13 temos a solução exata na esquerda (que neste caso é descontínua), e na direita os coeficientes para a inspeção visual da condição discreta de Picard.

Figura 4.13: *Wing*: Solução exata e avaliação da CDP.

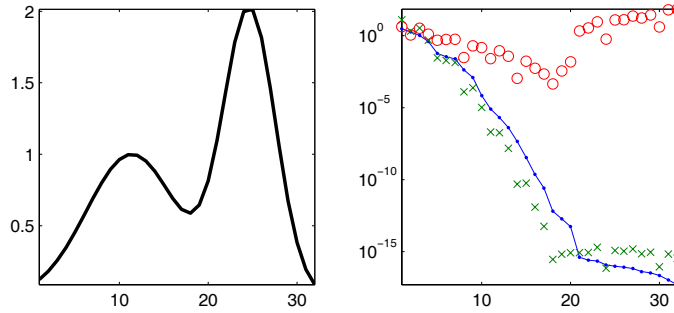
Shaw (Modelo de restauração de imagem unidimensional)

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

$$K(x, y) = (\cos(x) + \cos(y)) \left(\frac{\sin(u)}{u} \right)^2 \quad (4.4.5)$$

$$f(y) = a_1 e^{-c_1(y-t_1)^2} + a_2 e^{-c_2(y-t_2)^2} \quad (4.4.6)$$

em que $u = \pi (\sin(x) + \sin(y))$ 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 4.14 mostramos 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.

Figura 4.14: *Shaw*: Solução exata e avaliação da CDP.

Baart

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

$$K(x, y) = e^{x \cos(y)} \quad (4.4.7)$$

$$g(x) = 2 \frac{\sinh(x)}{x} \quad (4.4.8)$$

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

$$f(y) = \sin(y) \quad (4.4.9)$$

Na Figura 4.15 temos a solução exata na esquerda e na direita os coeficientes para a inspeção visual da condição discreta de Picard.

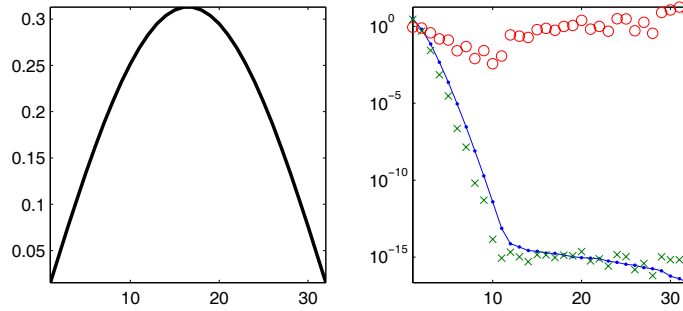


Figura 4.15: *Baart*: Solução exata e avaliação da CDP.

Deriv2 (Cálculo da segunda derivada)

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

$$K(x, y) = \begin{cases} x(y-1), & x < y \\ y(x-1), & x \geq y \end{cases} \quad (4.4.10)$$

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

exemplo 1 : $g(x) = (x^3 - x)/6, \quad f(y) = y$

exemplo 2 : $g(x) = e^x + (1 - e)x - 1, \quad f(y) = e^y$

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

Na Figura 4.16 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.

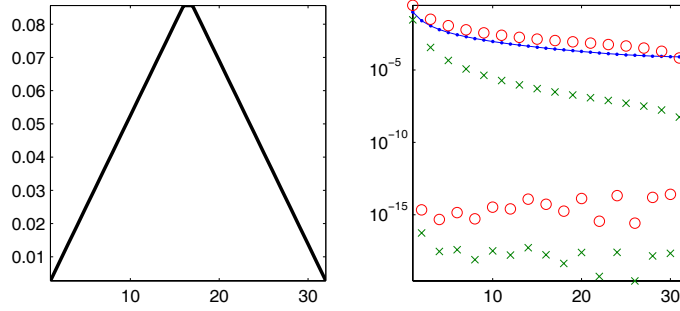


Figura 4.16: *Deriv2*: Solução exata e avaliação da CDP.

Gravity

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

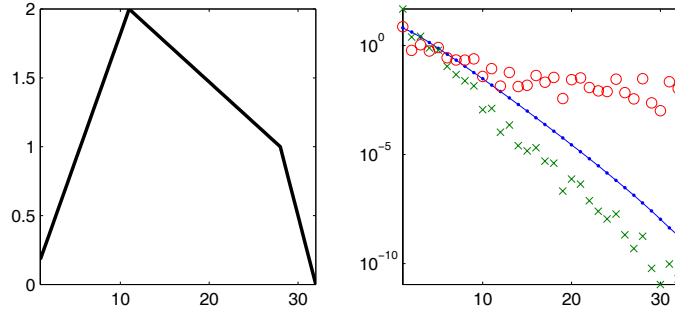
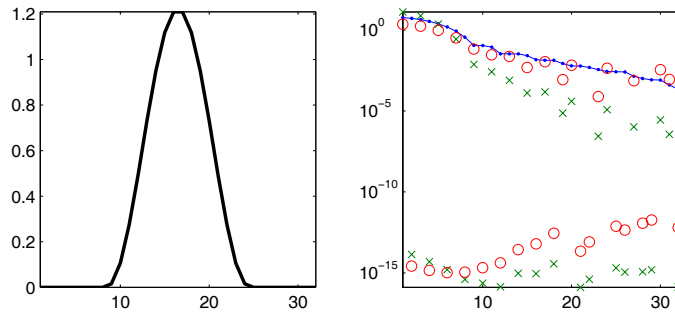
$$K(x, y) = d(d^2 + (x - y)^2)^{-3/2} \quad (4.4.11)$$

e função f é dada por

$$f(y) = \sin(\pi y) + \frac{1}{2} \sin(2\pi y), \quad (4.4.12)$$

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 4.17 temos a solução exata para o exemplo em que a solução é uma função contínua por partes, porém a equação (4.4.12) 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.

Figura 4.17: *Gravity*: Solução exata e avaliação da CDP.Figura 4.18: *Phillips*: Solução exata e avaliação da CDP.

Phillips

Esta é a equação que Phillips [50] estudou. Vamos definir a função

$$\zeta(p) = \begin{cases} 1 + \cos\left(p\frac{\pi}{3}\right), & |p| < 3 \\ 0, & |p| \geq 3 \end{cases} \quad (4.4.13)$$

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

$$\begin{aligned} K(x, y) &= \zeta(x - y) \\ f(y) &= \zeta(y) \\ g(x) &= (6 - |x|) \left(1 + \frac{1}{2} \cos\left(x\frac{\pi}{3}\right)\right) + \frac{9}{2\pi} \sin\left(|x|\frac{\pi}{3}\right) \end{aligned} \quad (4.4.14)$$

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

Na Figura 4.18 temos a solução exata na esquerda e na direita os coeficientes para a inspeção visual da condição discreta de Picard.

4.4.1 Resultados numéricos

Nos resultados a seguir, utilizamos um vetor de dados com erros relativos $\|b^{\text{exato}} - b\|_2 / \|b^{\text{exato}}\|_2 = t$, com $t = 0,01$, $t = 0,025$, $t = 0,05$, o que representa, respectivamente, 1%, 2,5% e 5%. Em todos os problemas o vetor de perturbações $e = b - b^{\text{exato}}$, contém números aleatórios Gaussianos com média zero gerados pela rotina *randn* do Matlab e inicializados com *seed*=0. Os testes foram realizados com a dimensão $n = 4096$ e executados 500 vezes para que possamos obter algumas medidas estatísticas. 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 *deriv2* 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} . Os seguintes símbolos são utilizados na descrição de resultados:

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. mínima) do subespaço projetado.
$\text{It}_{M,m}$	Número máximo (resp. mínimo) de avaliações da função $\phi(\lambda)$.

Tabela 4.8: Legenda para os dados colhidos nos testes numéricos realizados.

Vejamos os resultados obtidos quando temos 1% de erro relativo nos dados. Analisando a Tabela 4.9 podemos perceber LANC-FP preserva a característica do algoritmo de ponto-fixo devido a Bazán [3] de ser consistente na determinação do parâmetro de regularização independente do tamanho do erro nos dados. A dimensão do espaço projetado (k_M) utilizado foi um resultado interessante: nenhum dos problemas precisou mais do que 0,6% da dimensão original do problema ($n = 4096$) para atingir bons resultados.

Vejamos o que acontece quando existe um pouco mais de erro no dados, ou seja, quando temos $\|e\| / \|b^{\text{exato}}\|_2 = 0,025$. Os resultados obtidos nas 500 execuções podem ser conferidos na tabela 4.10.

Assim como no caso de 1% de erro relativo nos dados, para o caso de 2,5%

	<i>foxgood</i>	<i>heat</i>	<i>wing</i>	<i>shaw</i>	<i>baart</i>	<i>deriv2</i>	<i>gravity</i>	<i>phillips</i>
λ	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
It_M	5	41	5	7	6	10	10	14
It_m	4	36	5	6	5	4	6	4

Tabela 4.9: Resultados obtidos após 500 execuções e 1% de erro relativo.

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 diminuíram 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).

	<i>foxgood</i>	<i>heat</i>	<i>wing</i>	<i>shaw</i>	<i>baart</i>	<i>deriv2</i>	<i>gravity</i>	<i>phillips</i>
λ	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
It_M	6	33	7	5	7	5	8	12
It_m	5	29	6	5	6	4	4	4

Tabela 4.10: Resultados obtidos após 500 execuções e 2,5% de erro relativo.

Finalmente a última tabela com resultados para o caso de dados com 5% de erro relativo pode ser conferida na tabela 4.11 a seguir.

	<i>foxgood</i>	<i>heat</i>	<i>wing</i>	<i>shaw</i>	<i>baart</i>	<i>deriv2</i>	<i>gravity</i>	<i>phillips</i>
λ	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
It_M	6	30	11	6	7	5	7	11
It_m	6	25	10	5	6	5	4	4

Tabela 4.11: 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 a performance de LANC-FP com outras técnicas, faremos uma avaliação com alguns dos problemas teste descritos utilizando erros de 1% e dimensão $n = 1024$ de modo que possamos calcular a SVD das matrizes para cada problema e aplicar a curva-L, a GCV e a discrepância.

Nas tabelas 4.12, 4.13 e 4.14 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.

<i>foxgood</i>	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 4.12: Comparação entre Curva-L, GCV, FP, DP e Lanc-FP para o problema *foxgood*.

Desta comparação podemos concluir que além do algoritmo LANC-FP obter resultados praticamente idênticos aos do algoritmo de ponto-fixe original, os problemas projetados usaram subespaços de dimensão consideravelmente menores.

<i>heat</i>	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 4.13: Comparação entre Curva-L, GCV, FP, DP e LANC-FP para o problema *heat*.

<i>shaw</i>	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 4.14: Comparação entre Curva-L, GCV, FP, DP e LANC-FP para o problema *shaw*.

Capítulo 5

Reconstrução e Super Resolução de Imagens

Este capítulo descreve a performance de vários métodos de regularização quando aplicados a problemas de reconstrução de imagens.

5.1 Reconstrução de Imagem

Nesta seção abordamos um problema discreto mal-posto proveniente da discretização de uma equação integral de primeira espécie em duas dimensões

$$\int_0^1 \int_0^1 K(s, t; x, y) f(x, y) dx dy = g(s, t), \quad 0 \leq s, t \leq 1. \quad (5.1.1)$$

Na área de reconstrução de imagens, a função K é chamada de point spread function (PSF), f é uma imagem contínua (a “imagem exata”), e g é a imagem embaçada (blurred image) contínua. A determinação de estimativas de f a partir de uma imagem embaçada e contaminada por ruídos $\tilde{g} = g + e$, é conhecida como o problema de “deblurring”, que aqui chamamos de problema de reconstrução [37, 60].

Na prática, trabalha-se com imagens digitais, que informalmente falando, provém de amostragens de imagens contínuas, e são portanto armazenadas em matrizes. Assim, se o problema (5.1.1) é discretizado usando alguma regra de quadratura, obtem-se um sistema de equações lineares

$$A\mathbf{f} = \mathbf{g}, \quad (5.1.2)$$

com

$$\mathbf{f} = \text{vec}(\mathbf{F}), \quad \mathbf{g} = \text{vec}(\mathbf{G}). \quad (5.1.3)$$

Aqui o símbolo $\text{vec}(V)$ denota o vetor em $\mathbb{R}^{(m \times n) \times 1}$, obtido concatenando as colunas da matriz $V \in \mathbb{R}^{m \times n}$, e \mathbf{F} e \mathbf{G} são as matrizes que contém as amostras de f e g respectivamente.

Quando a PSF é do tipo $K(s, t; x, y) = K_1(s, x)K_2(t, y)$ (conhecido como o caso em que a PSF é separável), o problema discretizado torna-se

$$A_1 \mathbf{F} A_2^T = \mathbf{G}, \quad (5.1.4)$$

com $A_1 \in \mathbb{R}^{p \times m}$, $\mathbf{F} \in \mathbb{R}^{m \times n}$, $A_2 \in \mathbb{R}^{r \times n}$, e $\mathbf{G} \in \mathbb{R}^{p \times r}$. Usando em (5.1.4) a propriedade do produto de Kronecker de matrizes [36]

$$\text{vec}(A_1 \mathbf{F} A_2^T) = (A_2 \otimes A_1) \text{vec}(\mathbf{F}),$$

conclui-se que a matriz do sistema de equações lineares (5.1.2) é da forma

$$A = (A_2 \otimes A_1).$$

Na prática, imagens digitais são obtidas através de algum dispositivo eletrônico tais como câmeras ou sensores, e são compostas por elementos chamados de *pixels*, cada pixel representando uma variável do vetor de incógnitas em (5.1.2). Imagens pequenas normalmente têm aproximadamente $256^2 = 65.536$ pixels, mas outras podem facilmente chegar a 5 ou 10 milhões de pixels. Existem diversos fatores que podem fazer com que a imagem fique embaçada. O efeito *blur* pode ser, entre outros, devido à limitação em se representar as cores num sistema computacional ou devido à uma má qualidade das lentes que podem desfocar a imagem. Um exemplo do efeito *blur* sobre uma imagem pode ser visto na figura 4.6 no capítulo precedente.

Para entender melhor o efeito *blur*, 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 do ponto na sua vizinhança, como ilustrado na figura 5.1. O tipo de espalhamento depende da PSF. Assim, um modelo matemático que produza o efeito *blur* deve ser introduzido ou estimado.



Figura 5.1: Esquerda: um único ponto de luz, chamado de *point source*. Direita: o ponto de luz espalhado.

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 usualmente definido por

$$p_{ij} = \begin{cases} \frac{1}{\pi r^2} & \text{se } (i-k)^2 + (j-l)^2 \leq r^2 \\ 0 & \text{caso contrário,} \end{cases} \quad (5.1.5)$$

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 [37, 56] e os elementos da PSF são dados por

$$p_{ij} = \exp \left(-\frac{1}{2} \begin{bmatrix} i-k \\ j-l \end{bmatrix}^T \begin{bmatrix} s_1^2 & \rho^2 \\ \rho^2 & s_2^2 \end{bmatrix}^{-1} \begin{bmatrix} i-k \\ j-l \end{bmatrix} \right), \quad (5.1.6)$$

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) . Para outras PSF's, ver [36] e as referências contidas nele.

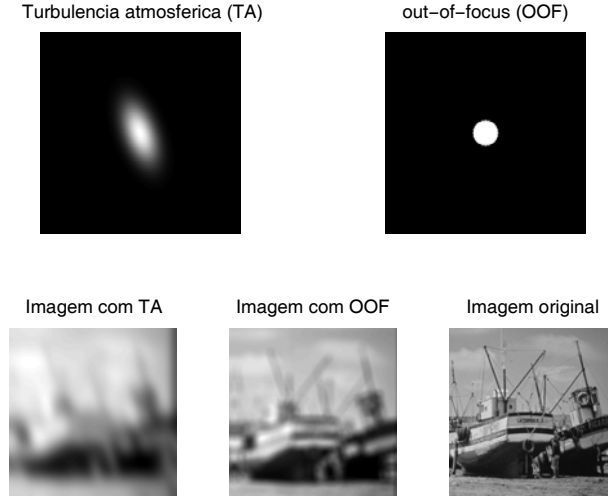


Figura 5.2: Da esquerda para a direita. Parte superior: PSF para turbulência atmosférica e *out-of-focus*. Parte inferior: Imagem embaçada com turbulência atmosférica, imagem fora-de-foco e imagem original.

Diferentes PSF's produzem diferentes matrizes e efeitos na imagem original como podemos conferir na figura 5.2 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 detalhes da modelagem e outras informações do efeito *blur*, ver [36, 49].

A performance dos algoritmos descritos nos capítulos precedentes será agora ilustrada na resolução do problema de reconstrução de imagens.

5.1.1 Astronomia

Neste caso usamos uma PSF e uma imagem conhecida como *satellite*, ambas da toolbox RestoreTools [15]. A imagem exata vem em tons de cinza, e tem 256×256 pixels, o que significa que temos um problema com 65.536 variáveis. Para este problema usamos Lanc-FP e HyBR, com W-GCV (Weighted-GCV) [15] como critério de parada conforme descrito em [15]. Apresentamos resultados correspondentes a 50 execuções onde o vetor de “dados” em (5.1.2) é contaminado por ruídos com erros relativos de 1%, 5% e 0,1% respectivamente. A imagem original e uma imagem embaçada e contaminada por ruído branco são apresentadas na figura 5.3.

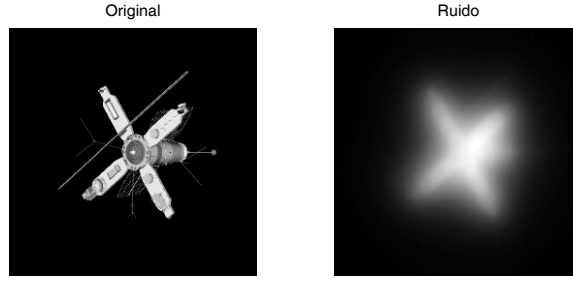


Figura 5.3: Solução exata e vetor de dados b perturbado.

Nas Tabelas 5.1, 5.2 e 5.3, o subíndice p e o símbolo \perp significam que foi utilizado preconditionador e reortogonalização no processo de bidiagonalização. Os símbolos restantes são os mesmos descritos na tabela 4.8 exceto que $It_{M,n}$ significa o número máximo (resp. mínimo) de avaliações da função $\phi^{(2)}(\lambda)$ já que neste teste escolhemos $p = 2$.

Os resultados obtidos quando o erro relativo nos dados é de 1% são descritos na tabela 5.1. Observações relevantes dos resultados são que o uso de um preconditionador tem um impacto na convergência dos algoritmos reduzindo a dimensão do espaço projetado (de 166 para menos de 13 no caso de LANC-FP, o que significa uma redução de mais de 92%), e que LANC-FP mostra uma vez mais uma certa robusteza no cálculo do parâmetro de regularização, no sentido do parâmetro não diferir muito uma vez estabelecido

	Lanc-FP _p [⊥]	Lanc-FP _p	Lanc-FP [⊥]	Lanc-FP	HyBR _p	HyBR
λ	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 _{λ}	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
It _{M}	5	3	6	6	-	-
It _{m}	5	3	6	6	-	-

Tabela 5.1: Resultados obtidos com lado direito com 1% de erro relativo.

o nível de ruído nos dados (ver o campo do desvio padrão), independente da utilização de preconditionador ou de reortogonalização. Comparando os resultados obtidos nos campos referentes aos erros relativos nas soluções, pode-se concluir que LANC-FP é competitivo uma vez que os erros relativos de ambos os algoritmos ficaram próximos dos 30%. Os resultados de uma execução são ilustrados na figura 5.4.

Vejamos o que acontece quando o nível de ruído é elevado para 5%. A grande surpresa neste caso foi que o algoritmo HyBR, sem a utilização de preconditionador, apresentou um erro relativo médio \bar{E} de 48% enquanto que os demais não ultrapassaram a casa dos 40%.

	Lanc-FP _p [⊥]	Lanc-FP _p	Lanc-FP [⊥]	Lanc-FP	HyBR _p	HyBR
λ	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 _{λ}	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
It _{M}	5	3	6	6	-	-
It _{m}	5	3	6	6	-	-

Tabela 5.2: Resultados obtidos com lado direito com 5% de erro relativo.

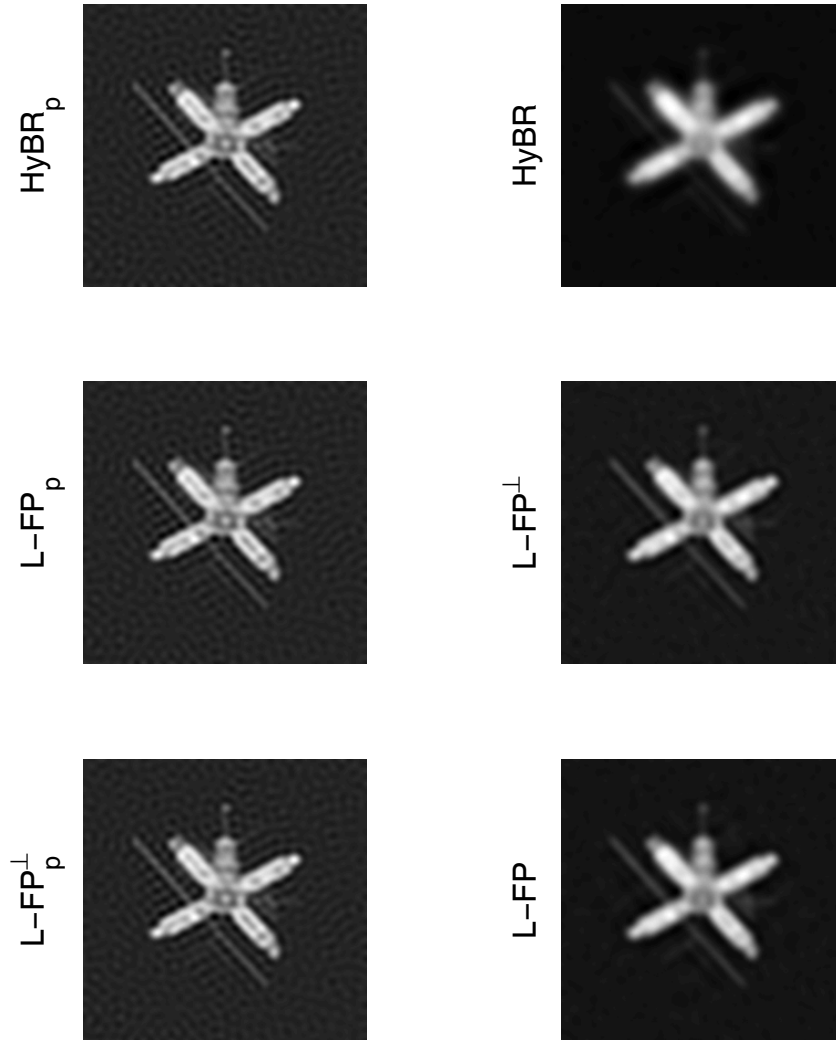


Figura 5.4: Soluções obtidas pelos métodos HyBR e Lanc-FP com e sem preconditionador e/ou reortogonalização.

Finalmente, os resultados para o caso em que o nível de ruído é baixo (erros da ordem de 0,1%) vêm na tabela 5.3. Nenhuma surpresa: LANC-FP apresenta resultados similares aos descritos anteriormente.

	Lanc-FP_p^\perp	Lanc-FP_p	Lanc-FP^\perp	Lanc-FP	HyBR_p	HyBR
λ	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_λ	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
It_M	3	3	3	3	-	-
It_m	3	3	3	3	-	-

Tabela 5.3: Resultados obtidos com lado direito com 0,1% de erro relativo.

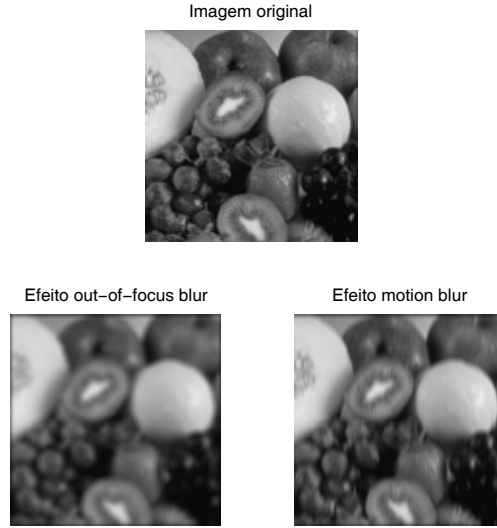


Figura 5.5: Imagem original (acima). Imagens com *out-of-focus blur* (esquerda) e *motion blur* (direita).

5.1.2 Paisagem

Independentemente do tipo da imagem, podemos ter vários tipos de embaçamento. Como um exemplo, na figura 5.5 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).

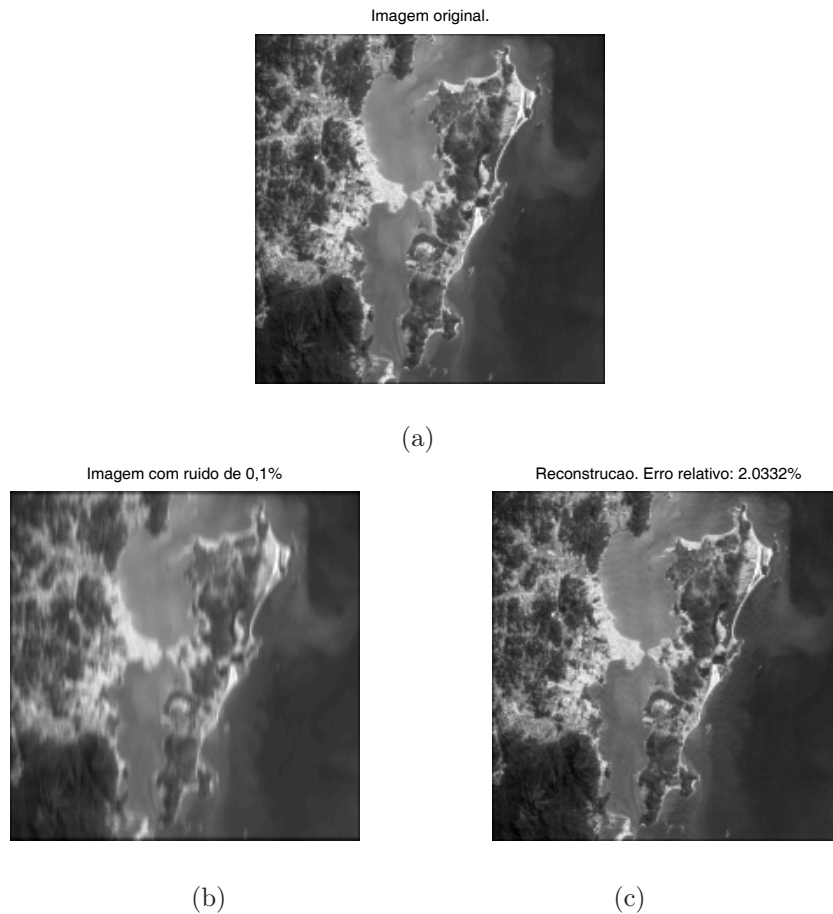


Figura 5.6: (a): Ilha de Santa Catarina, Imagem original. (b): Ilha de Santa Catarina com *motion blur* e ruído de 0,1% . (c): Imagem reconstruída.

Nesta seção apresentamos a tentativa de remover o efeito *blur* em dois tipos de paisagem, um da ilha de Santa Catarina - Florianópolis - e outro

de uma foto colorida da praia da Armação (Florianópolis).

Ilha de Santa Catarina

Para simular um efeito *blur* usamos a rotina *mblur* [35] com dimensão $n = 200$, o que gera um sistema de 40.000 variáveis, e o ruído é gerado pela rotina *WhiteNoise* da toolbox *RestoreTools* com 0,1% de intensidade. Os resultados de usar Lanc-FP são mostrados na figura 5.6, na qual aparecem a imagem original, a imagem embaçada e contaminada por ruído, e a imagem recuperada. 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.

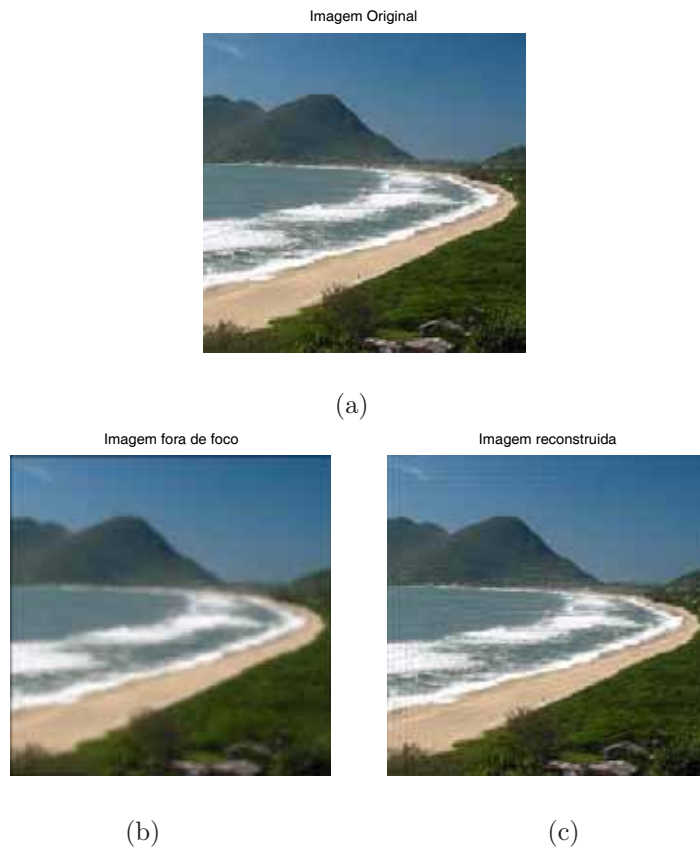


Figura 5.7: (a): Praia da Armação, Imagem original. (b): Praia da Armação com efeito *out-of-focus blur* e ruído de 0,1%. (c): Imagem reconstruída

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 *obblur* com dimensão $n = 256$ o que gera um sistema com 65.536 variáveis. Para problemas com fotos coloridas existem basicamente duas abordagens [36]. Uma é separar o problema em três, sendo um para cada camada, e a outra é formar um sistema com uma dimensão três vezes maior. Mais detalhes podem ser encontrados em [36].

Neste exemplo escolhemos tratar a imagem colorida, usando o sistema de cores RGB, resolvendo três sistemas de 65.536 variáveis. O ruído foi gerado como no exemplo anterior mas com intensidade de 0,1%. Os resultados de algoritmo Lanc-FP são ilustrados na Figura 5.7. O erro relativo na solução foi de 5,47% e o tempo de execução foi de 2,95 segundos num PC Athlon 64X2, 1.8GHz e 1GB de memória RAM. Os parâmetros de regularização encontrados (um para cada canal de cor) foram

$$\lambda_R = 0,0022, \quad \lambda_G = 0,0021 \quad \text{e} \quad \lambda_B = 0,0020$$

e os três subproblemas precisaram de 17 iterações para atingir o critério de parada.

5.1.3 Ressonância magnética

Em medicina existe uma grande variedade de problemas envolvendo imagens como tomografias e ressonâncias magnéticas, ultrassom, raio-X e muitas outras. No Matlab existe uma sequência de 27 imagens que representam um cérebro humano que no conjunto podem ser usadas para formar o desenho tridimensional do crânio. Outros detalhes podem ser encontrados no site [43]. Cada uma das imagens representa uma fatia do cérebro, ver figura 5.8.

Neste exemplo testamos o algoritmos Lanc-FP e HyBR usando erro relativo nos dados de 1%. Na Tabela 5.4 mostramos os erros obtidos em cada uma das fatias. Um aspecto interessante do experimento é que Lanc-FP obteve resultados melhores em todos os problemas associados a cada uma das 27 fatias.

Resultados correspondentes à fatia 15 são mostrados na Figura 5.9. Comparando as imagens e os erros (ver tabela 5.4) constatamos que Lanc-FP obteve um resultado ligeiramente melhor do que o encontrado por HyBR.

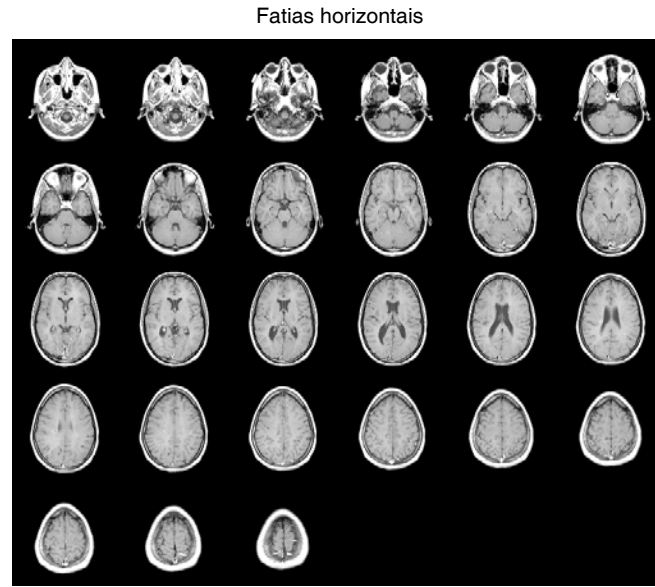


Figura 5.8: 27 fatias horizontais para o problema de ressonância magnética.

Fatia	Lanc-FP	HyBR	Fatia	Lanc-FP	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 5.4: Erros relativos em cada fatia obtidos com Lanc-FP e HyBR.

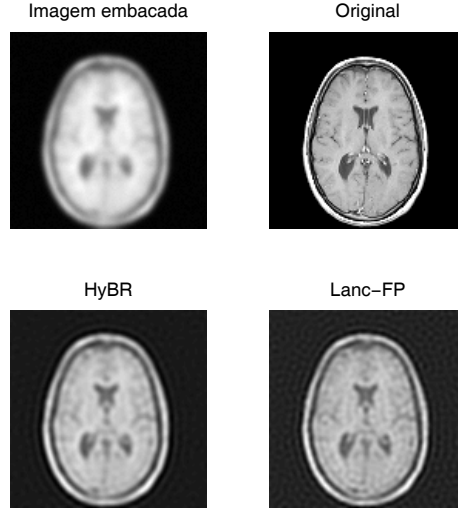


Figura 5.9: Fatia número 15 (dentre 27 disponíveis). Imagem original e duas soluções obtidas.

5.2 Super resolução de imagens

A resolução de uma imagem digital é representada por dois parâmetros: a *resolução espacial*, que especifica o número de pixels, e a *resolução da intensidade ou quantização*, que especifica o número de níveis possíveis para representar a intensidade em determinado pixel. Imagens de alta resolução (HR - high resolution) são de fundamental importância em várias aplicações. Por exemplo, o diagnóstico por imagem pode detectar doenças ainda em estágios iniciais, ou em astronomia, imagens HR podem revelar estrelas e galáxias nunca antes vistas. Basicamente, o objetivo das chamadas técnicas de super resolução é melhorar a resolução de imagens capturadas através de algum dispositivo de aquisição. Para tanto, assume-se que as imagens capturadas, chamadas imagens de baixa resolução (LR - low resolution), são altamente correlacionadas: pequenas diferenças de informação presente em cada imagem capturada proporciona uma nova informação que permite a recuperação de detalhes da imagem HR. Um exemplo de uma imagem HR e duas LR pode ser visto na figure 5.10.

Para descrever o problema abordado aqui, seja \mathbf{f} uma imagem HR de tamanho $M = M_1 \times M_2$ cuja representação usando a operação “vec” é $\mathbf{f} = [f_1, \dots, f_M]^T$, sendo M o número de pixels, e seja $\mathbf{g}_k = [g_{k,1}, g_{k,2}, \dots, g_{k,N}]^T$, $k = 1, 2, \dots, p$, a k -ésima imagem LR de tamanho $N = N_1 \times N_2$, com

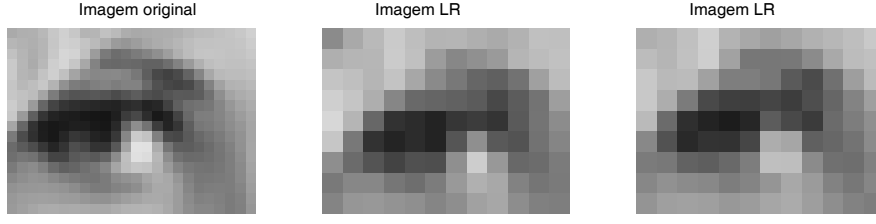


Figura 5.10: Imagem HR e duas imagens LR.

$M_1 = N_1 \times L_1$, $M_2 = N_2 \times L_2$, sendo L_1 e L_2 os fatores de subamostragem nas direções horizontal and vertical, respectivamente, do modelo que relaciona imagens HR e LR. Se assumimos que o processo de aquisição das imagens \mathbf{g}_k envolve embaçamento (blurring), movimento, subamostragem e ruído aditivo, um modelo linear que relaciona a imagem HR \mathbf{f} com as imagens LR \mathbf{g}_k pode ser escrito como [53]

$$\mathbf{g}_k = \mathbf{C}_k \mathbf{f} + \epsilon_k \quad (5.2.7)$$

com $\mathbf{C}_k \in \mathbb{R}^{N \times M}$ e ϵ_k denotando um vetor de ruídos. O problema de super resolução consiste na determinação de estimativas da imagem \mathbf{f} a partir da sequência de imagens \mathbf{g}_k .

Devido ao fato dos sistemas em (5.2.7) serem subdeterminados, podemos encontrar infinitas soluções \mathbf{f} e assim o problema de super resolução é mal posto. Para contornar esta dificuldade, dentre vários métodos disponíveis na literatura, um deles é o método de regularização de Tikhonov, que neste contexto pode ser formulado como

$$\hat{\mathbf{f}}_\lambda = \arg \min_{\mathbf{f}} \|\mathbf{g} - \mathbf{C}\mathbf{f}\|_2^2 + \lambda^2 \|\mathbf{R}\mathbf{f}\|_2^2 \quad (5.2.8)$$

sendo

$$\mathbf{g} = [\mathbf{g}_1^T \dots \mathbf{g}_p^T]^T, \quad \mathbf{C} = [\mathbf{C}_1^T \dots \mathbf{C}_p^T]^T,$$

e \mathbf{R} uma versão discreta de algum operador diferencial nas direções horizontal e vertical, respectivamente. E uma vez mais o desafio é determinar um parâmetro de regularização apropriado.

Como ilustração, abordamos o problema de estimar a imagem HR de tamanho 18×24 pixels mostrada na fig. 5.10 a partir da sequência of 5 imagens LR com fatores de subamostragem $L_1 = L_2 = 2$. Assim, $\mathbf{C} \in \mathbb{R}^{540 \times 432}$ e $\mathbf{R} \in \mathbb{R}^{822 \times 432}$. Neste caso, a matriz \mathbf{R} foi escolhida como um operador diferencial de primeira ordem. As imagens LR foram formadas contaminando as imagens degradadas por ruído branco aditivo ϵ de modo

que a razão ruído sinal $\text{SNR} = 15\text{dB}$; duas imagens LR são mostradas na Fig.5.10.

Neste problema verificamos que $\text{posto}(\mathbf{C}) = 390$ (considerando uma precisão de aproximadamente 16 casas decimais) e $\text{posto}(\mathbf{R}) = 341$. Também, como \mathbf{R} é sobredeterminada, primeiro computamos uma fatoração QR de \mathbf{R} , $\mathbf{R} = \mathbf{Q}\mathbf{R}$, e então substituímos em (5.2.8) $\|\mathbf{R}\mathbf{f}\|_2$ por $\|\mathbf{L}\mathbf{f}\|_2$ onde \mathbf{L} é o bloco superior de \mathbf{R} de tamanho 341×342 . Como nos exemplos anteriores, o problema foi resolvido 500 vezes a fim de comparar a performance do método de ponto-fixo (FP) com outros da literatura, tal como aquele proposto por Belge *et. al.* [7], denotado por BFP. Os resultados são apresentados na tabela 5.5.

	FP	BFP	curva-L	GCV
\bar{E}_f	0.0617	0.1471	0.0615	0.0622
E_{\max}	0.0748	0.2975	0.0751	0.0911
Std_E	0.0034	0.0670	0.0035	0.0045
$\bar{\lambda}$	0.3318	1.7753	0.1968	0.1944
I_{\max}	26	386	—	—
I_{\min}	19	21	—	—

Tabela 5.5: Resultados numéricos para o problema de super resolução.

A curva-L da última execução é mostrada na esquerda da Fig. 5.11; já na direita, aparecem os erros relativos das soluções calculadas nas primeiras 200 execuções. As imagens estimadas por FP, BFP e curva-L da última execução aparecem na segunda linha na Fig. 5.12. A imagem produzida por GCV é muito parecida com aquelas produzidas por FP e curva-L e por isso ela não é mostrada. Por outro lado, percebe-se que as imagens obtidas por FP e curva-L são muito mais precisas que aquelas obtidas por BFP, e isso pode ser também visto a partir dos erros relativos médios na tabela 5.5. Tabela 5.5 mostra também que o número de iterações usadas por BFP neste problema é significativamente maior daquele usado por FP e que BFP estima soluções que podem variar muito de uma execução para outra (veja o grande valor do desvio padrão e os erros relativos na Fig. 5.11). Neste problema a escolha $\mu = 1$ não funcionou em aproximadamente 5% dos casos.

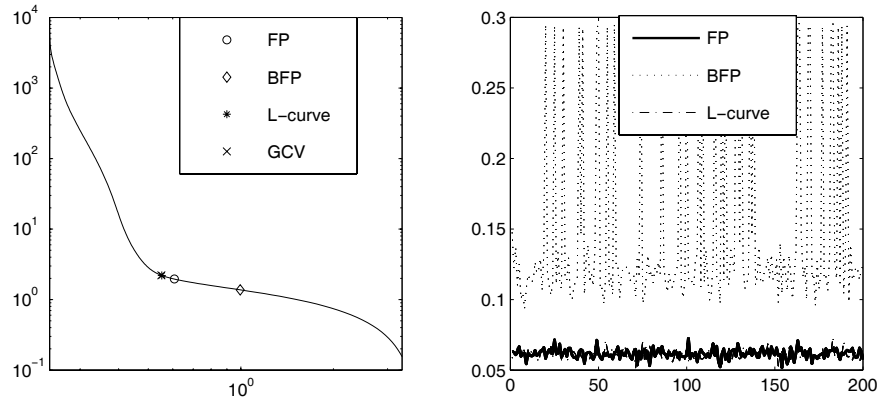


Figura 5.11: Esquerda: curva-L. Direita: erros relativos das primeiras 200 execuções.

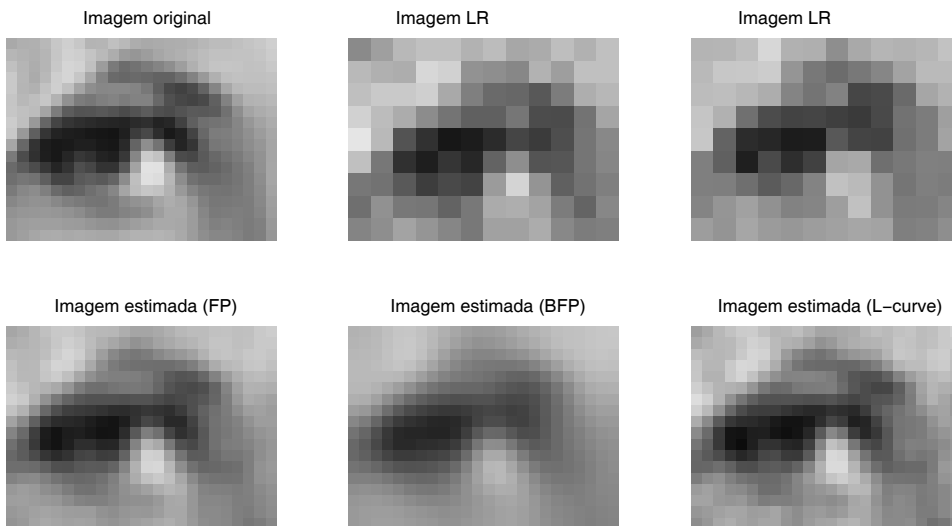


Figura 5.12: Imagem HR, duas imagens LR, e imagens estimadas por FP, BFP e curva-L.

5.3 Exercícios

1. Considere a equação integral

$$\int_0^1 \int_0^1 K(s, t; x, y) f(x, y) dx dy = g(s, t), \quad 0 \leq s, t \leq 1.$$

Análogo ao caso unidimensional, combine regras de quadratura e o método de colocação para discretizar a equação acima, descrevendo os elementos $a_{i,j}$ da matriz A e as componentes dos vetores \mathbf{f} e \mathbf{g} no problema discretizado (5.1.2).

2. Assuma que o “kernel” da equação integral do exercício anterior é da forma $K(s, t; x, y) = K_1(s, x)K_2(t, y)$, e que no problema discretizado correspondente,

$$A_1 \mathbf{F} A_2^T = \mathbf{G},$$

ambas as matrizes A_1 e A_2 são $m \times m$ e inversíveis. Assuma também que devido a incertezas nos aparelhos de medição, a imagem “dados” é da forma $\mathbf{G} = \mathbf{G}_{\text{exata}} + \mathbf{E}$, onde $\mathbf{G}_{\text{exata}} = A_1 \mathbf{F}_{\text{exata}} A_2^T$. Se $\mathbf{F} = A_1^{-1} \mathbf{G} A_2^{-T}$, demonstre que

$$\frac{\|\mathbf{F} - \mathbf{F}_{\text{exata}}\|_F}{\|\mathbf{F}_{\text{exata}}\|_F} \leq \|A_1\|_2 \|A_1^{-1}\|_2 \|A_2\|_2 \|A_2^{-1}\|_2 \frac{\|\mathbf{E}\|_F}{\|\mathbf{G}\|_F},$$

onde $\|(\cdot)\|_F$ denota a norma Frobenius da matriz (\cdot) .

3. Prove que

- $\text{vec}(A_1 \mathbf{X} A_2^T) = (A_2 \otimes A_1) \text{vec}(\mathbf{X})$,
- $(A_1 \otimes A_2)^T = A_2^T \otimes A_1^T$, $(A_1 \otimes A_2)^{-1} = A_2^{-1} \otimes A_1^{-1}$,
- $(U_1 \Sigma_1 V_1^T) \otimes (U_2 \Sigma_2 V_2^T) = (U_1 \otimes U_2)(\Sigma_1 \otimes \Sigma_2)(V_1 \otimes V_2)^T$.

Bibliografia

- [1] C. T. H. Baker, *The Numerical Treatment of Integral Equations*, Clarendon Press, Oxford, pp. 665, 1977.
- [2] J. Baumeister, *Stable Solution of Inverse Problems*, Fried. Vieweg& Sohn, Braunschweig, 1987.
- [3] F. S. V. Bazán, *Fixed-point iterations in determining the Tikhonov regularization parameter*, Inverse Problems, 24, 2008.
- [4] F. S. V. Bazán and J. B. Francisco, *Improved Fixed-point algorithm for determining the Tikhonov regularization parameter*, Inverse Problems, 25, 2009.
- [5] F. S. V. Bazán and L. S. Borges, *LANC-FP: an algorithm for large-scale discrete ill-posed problems*, submitted, março 2009.
- [6] F. S. V. Bazán, *CGLS-GCV: a hybrid algorithm for low-rank-deficient problems*, App. Num. Math. 47, pp. 91-108, 2003.
- [7] M. Belge, M. E. Kilmer and E. L. Miller, *Efficient determination of multiple regularization parameters in a generalized L-curve framework*, Inverse Problems 18, pp. 1161-1183, 2002.
- [8] Å. Björck, *A Bidiagonalization Algorithm for Solving Large and Sparse Ill-Posed Systems of Linear Equations*, BIT 38, pp. 659-670, 1988.
- [9] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, 1996.
- [10] D. Calvetti, G. H. Golub and L. Reichel, *Estimation of the L-curve via Lanczos bidiagonalization*, BIT, 39, pp 603-619, 1999.
- [11] D. Calvetti, B. Lewis and L. Reichel, *On the regularizing properties of the GMRES method*, Numer Math, 91, pp. 605-625, 2002.

- [12] D. Calvetti and L. Reichel, *Tikhonov Regularization of Large Linear Problems*, BIT 43, pp. 263-283, 2003.
- [13] D. Calvetti, L. Reichel and A. Shuibi, *L-curve and curvature bounds for Tikhonov regularization*, Numerical Algorithms, 35, pp 301-314, 2004.
- [14] A. S. Carasso, *Determining surface temperatures from interior observations*, SIAM J. Appl. Math., 42, pp. 558-574, 1982.
- [15] J. Chung, J. G. Nagy and D. P. O'Leary, *A Weighted-GCV Method for Lanczos-Hybrid Regularization*, Electronic Transaction on Numerical Analysis, Vol. 28, pp 149-167, 2008.
- [16] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, 1997.
- [17] A. Dold and B. Eckmann, *Lecture Notes in Mathematics*, Inverse Problems, 1986.
- [18] E. Eldén, *A weighted pseudoinverse, generalized singular values, and constrained least squares problems*, BIT, 22, pp. 487-501. 1982.
- [19] E. Eldén, *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.
- [20] H. W. Engl, M. Hanke and A. Neubauer, *Regularization of Inverse Problems*, Kluwer Academic Publishers, Dordrecht, 1996.
- [21] G. H. Golub, *Numerical Methods for Solving Linear Least Squares Problems*, Numerische Mathematik 7, pp 206-216, 1965.
- [22] G. H. Golub, M. T. Heath and G. Wahba, *Generalized cross-validation as a method for choosing a good ridge parameter*, Technometrics, 21, pp. 215-223, 1979.
- [23] G. H. Golub, C. F. Van Loan, *Matrix Computations*, Third Edition, The Johns Hopkins University, London, 1996.
- [24] C. W. Groetsch, *Elements of Applicable Functional Analysis*, Dekker, New-York, 1980.
- [25] C. W. Groetsch, *The theory of Tikhonov regularization for Fredholm equations of the first kind*, Research Notes in Mathematics, 105, Pitman, Boston, 1984.

- [26] M. Gu and S. C. Eisenstat, *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.
- [27] M. Hanke, *Limitations of the L-curve method in ill-posed problems*, BIT 36, pp 287-301, 1996.
- [28] M. Hanke and P. C. Hansen, *Regularization methods for large-scale problems*, Surveys Math. Indust., v3, pp. 253-315, 1993.
- [29] P. C. Hansen, *Regularization, GSVD and Truncated GSVD*, BIT 29, pp. 491-504, 1989.
- [30] P. C. Hansen, *The Discrete Picard Condition For Discrete Ill-Posed Problems*, BIT 30, pp. 658-672, 1990.
- [31] P. C. Hansen, *Analysis of discrete ill-posed problems by means of the L-curve*, SIAM Review, 34, pp. 561-580, 1992.
- [32] P. C. Hansen, *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.
- [33] P. C. Hansen, *Rank-deficient and discrete ill-posed problems*, SIAM Philadelphia, PA, 1998.
- [34] P. C. Hansen, *Regularization Tools: A MATLAB package for analysis and solution of discrete ill-posed problems*, Numer. Algorithms, 6 pp. 1-35, 1994.
- [35] P. C. Hansen, *Regularization Tools*, <http://www2.imm.dtu.dk/~pch/Regutools/>.
- [36] P. C. Hansen, J. G. Nagy and D. P. O’Leary, *Deblurring Images. Matrices, Spectra and Filtering.*, SIAM, 2006.
- [37] A. K., Jain, *Fundamentals of Digital Image Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1989.
- [38] M. Jiang, L. Xia, L., and G. Shou, *Combining regularization frameworks for solving the electrocardiography inverse problem*, Communications in Computer and Information Science, Vol. 2, pp. 1210-1219, 2007.

- [39] M. Jiang, L. Xia, G. Shou and M. Tang, *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.
- [40] M. Jiang, L. Xia, G. Shou, F. Liu and S. Crozier, *Two hybrid regularization frameworks for solving the electrocardiography inverse problem*, Physics in Medicine and Biology, 53, pp. 5151-5164, 2008.
- [41] P. R. Johnston and R. M. Gulrajan, *An analysis of the zero-crossing method for choosing regularization parameters*, SIAM J. Sci. Comput. Vol 24, No. 2, pp 428-442, 2002.
- [42] M. E. Kilmer and D. P. O'Leary, *Choosing regularization parameters in iterative methods for ill-posed problems*, SIAM J. Matrix Anal. Appl., Vol. 22, No. 4, pp. 1204-1221, 2001.
- [43] Mathworks, <http://www.mathworks.com/products/image/demos.html?file=/products/demos/shipping/images/ipexmri.html>
- [44] M. Lavrent'ev and M. Romanov, *Ill-posed problems of mathematical physics and analysis*, Translation of Mathematical Monographs, AMS, vol. 64, Providence 1986.
- [45] P. Linz, *Theoretical Numerical Analysis; an introduction to advances techniques*, New-York, Wiley, 1979.
- [46] S. Morigi, L. Reichel, F. Sgallari e F. Zama, *Iterative methods for ill-posed problems and semiconvergent sequences*. J. Comput. Appl. Math., vol. 193, pp. 157-167, 2006.
- [47] V. A. Morozov, *The error principle in the solution of operational equations by the regularization method*, USSR Comput. Math. and Phys. 8 (no.2), pp. 63-87, 1968.
- [48] V. A. Morozov, *On the solution of functional equations by the method of regularization*, Soviet Math. Dokl., 7, pp. 414-417, 1966.
- [49] J. G. Nagy, K. Palmer and L. Perrone, *Iterative methods for image deblurring: A MATLAB object oriented approach*, Numer. Algorithms, 36, pp. 73-93, 2004.
- [50] D. L. Phillips, *A technique for the numerical solution of certain integral equations of the first kind*, J. of the ACM, Vol. 9, pp. 84-97, 1962.

- [51] C. C. Paige and M. A. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Softw, Vol. 8, No. 1, pp 43-71, 1982.
- [52] C. C. Paige and M. A. Saunders, *LSQR: Sparse linear equations and least squares problems*, ACM Trans. Math. Softw, Vol. 8, No. 2, pp 195-209, 1982.
- [53] S. C. Park, M. K. Park, and M. G. Kang, *Super-resolution image reconstruction: a technical overview*, IEEE Signal Processing Magazine, vol. 20, no. 3, pp.21-36, May 2003.
- [54] T. Regińska, *A regularization parameter in discrete ill-posed problems*, SIAM J. Sci. Comput. Vol. 17, No. 3, pp. 740-749, 1996.
- [55] J. Riley, *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.
- [56] M. C. Roggemann and B. Welsh, *Imaging Through Turbulence*. CRC Press, Boca Raton, FL, 1996.
- [57] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput. 7, pp. 856-869, 1986.
- [58] A. N. Tikhonov, *Solution of incorrectly formulated problems and the regularization method*, Soviet Math. Dokl., pp. 1035-1038, Vol. 4, 1963.
- [59] C. R. Vogel, *Non-convergence of the L-curve regularization parameter selection method*, Inverse Problems, Vol. 12, pp. 535-547, 1996.
- [60] C. R. Vogel, *Computational Methods for Inverse Problems*, Frontiers in Applied Mathematics, SIAM, Philadelphia, 2002.
- [61] D. S. Watkins, *Fundamentals of Matrix Computations*, John Wiley & Sons, 1991.
- [62] M. V. W. Zibbeti, F. S. V. Bazán, and J. Mayer, *Determining the regularization parameters for super-resolution problems*, Signal Process. **88**(2008), 2890-2901.

Índice

- Algoritmo
 - de ponto-fixo, 35, 44
 - LANC-FP, 47
- Condição
 - de Picard, 16
 - Discreta de Picard, 30, 31
- Curva-L, 35, 41
- deblurring, 75
- Decomposição
 - em Valores Singulares, 22
 - em Valores Singulares Generalizados, 26
- delta de Dirac, 12
- Equação
 - do calor, 15
 - integral, 14
- Equações
 - normais, 25
 - normais regularizadas, 25
- Espaço de Hilbert, 16
- Fatores de filtro, 29
- Forma
 - geral, 25
 - padrão, 25
- FP-tools, 63
- Função de transição, 12
- Funcional de Tikhonov, 19
- GCV, 35, 37
- GMRES, 36
- Gradientes Conjugados, 36
- GSVD, 26
- GSVD Truncada, 28
- HyBR, 61
- Inversa generalizada, 26
- Iteração de Landweber, 50
- LANC-FP, 46
- LSQR, 36
- Método
 - de quase soluções, 18
 - dos Gradientes Conjugados, 50
 - GMRES, 51
 - LSQR, 51
- Métodos de regularização, 17
- Operador projeção ortogonal, 17
- Operadores regularizantes, 17
- Parâmetro de regularização, 17
- pixel, 76
- Princípio da Discrepância, 35, 39
- Problema
 - bem-posto, 13
 - de reconstrução, 75
 - direto, 12
 - inverso, 12
 - mal-posto, 13
 - regularizado projetado, 46

PSF, 60, 75
 Reconstrução
 Astronomia, 78
 de Imagem, 75
 Paisagem, 82
 Ilha de Santa Catarina, 83
 Praia da Armação, 84
 Ressonância magnética, 84
 Regularização
 de Tikhonov, 18, 24
 iterativa, 49
 RegularizationTools, 53
 Resolução
 espacial, 86
 resolução
 da intensidade, 86
 da quantização, 86
 RestoreTools, 59

 Sistema singular, 16
 Sistemas ortogonais, 16
 Solução
 regularizada, 47, 56
 Super resolução de imagens, 86
 SVD, 22
 SVD Truncada, 23

 Técnica de Reconstrução Algébrica,
 52
 Teoria de regularização, 17
 TGSVD, 28, 36
 Tikhonov iterado, 25
 toolbox
 FP-tools, 63
 RegularizationTools, 53
 RestoreTools, 59
 TSVD, 23, 36

 Valores singulares, 16

 W-GCV, 36, 38
 Weighted-GCV, 36, 38

NOTAS EM MATEMÁTICA APLICADA

Arquivos em pdf disponíveis em <http://www.sbmact.org.br/notas.php>

1. Restauração de Imagens com Aplicações em Biologia e Engenharia
Geraldo Cidade, Antônio Silva Neto e Nilson Costa Roberty
2. Fundamentos, Potencialidades e Aplicações de Algoritmos Evolutivos
Leandro dos Santos Coelho
3. Modelos Matemáticos e Métodos Numéricos em Águas Subterrâneas
Edson Wendlander
4. Métodos Numéricos para Equações Diferenciais Parciais
Maria Cristina de Castro Cunha e Maria Amélia Novais Schleicher
5. Modelagem em Biomatemática
Joyce da Silva Bevilacqua, Marat Rafikov e Cláudia de Lello
Courtouke Guedes
6. Métodos de Otimização Randômica: algoritmos genéticos e “simulated annealing”
Sezimária F. Pereira Saramago
7. “Matemática Aplicada à Fisiologia e Epidemiologia”
H.M. Yang, R. Sampaio e A. Sri Ranga
8. Uma Introdução à Computação Quântica
Renato Portugal, Carlile Campos Lavor, Luiz Mariano Carvalho
e Nelson Maculan
9. Aplicações de Análise Fatorial de Correspondências para Análise de Dados
Homero Chaib Filho

10. Modelos Matemáticos baseados em autômatos celulares para Geoprocessamento
Marilton Sanchotene de Aguiar, Fábila Amorim da Costa,
Graçaliz Pereira Dimuro e Antônio Carlos da Rocha Costa
11. Computabilidade: os limites da Computação
Regivan H. N. Santiago e Benjamín R. C. Bedregal
12. Modelagem Multiescala em Materiais e Estruturas
Fernando Rochinha e Alexandre Madureira
13. Modelagem em Biomatemática (Coraci Malta ed.)
 - 1 - “Modelagem matemática do comportamento elétrico de neurônios e algumas aplicações”
Reynaldo D. Pinto
 - 2 - “Redes complexas e aplicações nas Ciências”
José Carlos M. Mombach
 - 3 - “Possíveis níveis de complexidade na modelagem de sistemas biológicos”
Henrique L. Lenzi, Waldemiro de Souza Romanha e Marcelo Pelajo- Machado
14. A lógica na construção dos argumentos
Angela Cruz e José Eduardo de Almeida Moura
15. Modelagem Matemática e Simulação Numérica em Dinâmica dos Fluidos
Valdemir G. Ferreira, Hélio A. Navarro, Magda K. Kaibara
16. Introdução ao Tratamento da Informação nos Ensinos Fundamental e Médio
Marcilia Andrade Campos, Paulo Figueiredo Lima
17. Teoria dos Conjuntos Fuzzy com Aplicações
Rosana Sueli da Motta Jafelice, Laércio Carvalho de Barros,
Rodney Carlos Bassanezi
18. Introdução à Construção de Modelos de Otimização Linear e Inteira
Socorro Rangel

19. Observar e Pensar, antes de Modelar
Flavio Shigeo Yamamoto, Sérgio Alves, Edson P. Marques Filho,
Amauri P. de Oliveira
20. Frações Contínuas: Propriedades e Aplicações
Eliana Xavier Linhares de Andrade, Cleonice Fátima Bracciali
21. Uma Introdução à Teoria de Códigos
Carlile Campos Lavor, Marcelo Muniz Silva Alves, Rogério
Monteiro de Siqueira, Sueli Irene Rodrigues Costa
22. Análise e Processamento de Sinais
Rubens Sampaio, Edson Cataldo, Alexandre de Souza Brandão
23. Introdução aos Métodos Discretos de Análise Numérica de EDO e
EDP
David Soares Pinto Júnior
24. Representações Computacionais de Grafos
Lílian Markenzon, Oswaldo Vernet
25. Ondas Oceânicas de Superfície
Leandro Farina
26. Técnicas de Modelagem de Processos Epidêmicos e Evolucionários
Domingos Alves, Henrique Fabrício Gagliardi
27. Introdução à teoria espectral de grafos com aplicações
Nair Maria Maia de Abreu, Renata Raposo Del-Vecchio, Cybele
Tavares Maia Vinagre e Dragan Stevanović
28. Modelagem e convexidade
Eduardo Cursi e Rubens Sampaio
29. Modelagem matemática em finanças quantitativas em tempo discreto
Max Oliveira de Souza e Jorge Zubelli
30. Programação não linear em dois níveis: aplicação em Engenharia
Mecânica
Ana Friedlander e Eduardo Fancello

31. Funções simétricas e aplicações em Combinatória
José Plínio de Oliveira Santos e Robson da Silva
32. Semigrupos aplicados a sistemas dissipativos em EDP
Carlos Raposo da Cunha
33. Introdução à Simulação Estocástica para Atuária e Finanças Usando R
Hélio Côrtes Vieira, Alejandro C. Frery e Luciano Vereda
34. Modelos de Sustentabilidade nas Paisagens Amazônicas Alagáveis
Maurício Vieira Kritz, Jaqueline Maria da Silva e Cláudia Mazza
35. Uma Introdução à Dinâmica Estocástica de Populações
Leonardo Paulo Maia
36. Geometria de Algoritmos Numéricos
Gregorio Malajovich
37. Equações Diferenciais, Teorema do Resíduo e as Transformadas Integrais
Edmundo Capelas de Oliveira e Jayme Vaz Júnior