

UNIVERSIDADE FEDERAL DO PARANÁ

ISABELA LUCIA DE OLIVEIRA

UMA INTRODUÇÃO ÀS TÉCNICAS DE
REGULARIZAÇÃO PARA PROBLEMAS
INVERSOS DISCRETOS LINEARES

Curitiba, PR

2024

ISABELA LUCIA DE OLIVEIRA

UMA INTRODUÇÃO ÀS TÉCNICAS DE
REGULARIZAÇÃO PARA PROBLEMAS INVERSOS
DISCRETOS LINEARES

–Trabalho acadêmico apresentado como requisito parcial para a obtenção de Título de Bacharel em Matemática Industrial–.

Universidade Federal do Paraná (UFPR)

Departamento de Matemática

Bacharelado em Matemática Industrial

Orientador: Prof^a Ana Gabriela Martínez

Curitiba, PR

2024

TERMO DE APROVAÇÃO

UMA INTRODUÇÃO ÀS TÉCNICAS DE REGULARIZAÇÃO PARA PROBLEMAS INVERSOS DISCRETOS LINEARES

ISABELA LUCIA DE OLIVEIRA

–Trabalho acadêmico apresentado como requisito parcial para a obtenção de Título de
Bacharel em Matemática Industrial–.

Avaliado pela seguinte banca examinadora:

Prof^ª: Ana Gabriela Martinez
Orientadora

Prof^ª: Ailin Ruiz de Zarate
UFPR

Prof^º: Manuel Jesus Cruz Barreda
UFPR

Curitiba, 18 de Dezembro de 2024.

AGRADECIMENTOS

A Deus que me dá forças para persistir diante de todas as dificuldades.

Aos meus pais, essa caminhada foi longa, e eu não teria chegado até aqui sem o apoio e amor, incondicional e sem limites, de vocês.

Aos meus irmãos, nossas pausas estratégicas para o café e risadas após horas de estudo na cozinha de casa são mais importantes do que posso descrever, obrigada por me trazerem equilíbrio.

Aos meus avós e demais familiares, obrigada pela torcida e orações. Agradeço imensamente a compreensão de vocês, e saibam que sei que estou devendo muitas visitas!

À minha orientadora, Ana Gabriela Martínez, este trabalho teria um título de apenas três palavras sem a sua orientação, agora tem três linhas! Brincadeiras à parte, obrigada pela sua dedicação, paciência e incentivos que tornaram este trabalho possível.

Aos demais professores do Departamento de Matemática da UFPR presentes na minha jornada acadêmica, minha admiração por cada um é imensa, só tenho a agradecer por esses cinco anos de aprendizado.

Aos meus amigos, enxergar sobre os ombros dos gigantes que nos orientam em sala de aula só foi possível porque vocês estavam do meu lado nessa escalada.

À minha psicóloga, que me ajuda a aplicar “técnicas de regularização” para amenizar o efeito das perturbações externas no meu dia a dia.

RESUMO

Problemas inversos são comumente encontrados em aplicações de restauração de imagens, processamento de sinais, estudos meteorológicos e sísmicos, imagens de tomografia, entre outras. Em todas essas áreas, o desafio que cerca o assunto é a classificação dos problemas inversos serem mal postos, isso implica que as soluções podem não ser viáveis e quando obtidas são sensíveis a pequenas perturbações nos dados iniciais, o que gera instabilidade. Essas “qualidades” motivam o estudo dos problemas inversos discretos lineares e indicam a necessidade de métodos que tenham por objetivo filtrar os ruídos e perturbações inerentes à natureza do problema. Este trabalho, portanto, objetiva fundamentar conceitos iniciais dos problemas inversos, apresentar e comparar os métodos de regularização clássicos: Regularização de Tikhonov e a Regularização TSVD, e também avaliar de maneira heurística os recursos que a Curva-L e Método de Discrepância de Morozov proporcionam para determinar o parâmetro de regularização mais adequado.

Palavras-Chave: Problemas Inversos Discretos. Problema Mal Posto. Estabilidade. Fatoração SVD. Regularização. Tikhonov. Determinação de Parâmetros.

ABSTRACT

Inverse problems are commonly encountered in applications such as image restoration, signal processing, meteorological and seismic studies, tomography imaging, among others. In all these areas, the challenge surrounding the topic is that inverse problems are often ill-posed, meaning the solutions may not be feasible and, when obtained, they are sensitive to small perturbations in the initial data, leading to instability. These “qualities” motivate the study of linear discrete inverse problems and highlight the need for methods aimed at filtering noise and disturbances inherent to the nature of the problem. This work, therefore, aims to lay the foundational concepts of inverse problems, present and compare classical regularization methods: Tikhonov Regularization and TSVD Regularization, as well as heuristically evaluate the resources that the Curve-L and Morozov’s Discrepancy Method provide to determine the most appropriate regularization parameter.

Keywords: Discrete Inverse Problems. Ill-posed Problem. Stability. SVD Factorization. Regularization. Tikhonov. Parameter Determination.

SUMÁRIO

	Sumário	11
1	INTRODUÇÃO	13
2	PROBLEMAS INVERSOS	15
2.1	Problemas Mal Postos	16
2.2	Estabilidade	18
3	FUNDAMENTAÇÃO TEÓRICA	21
3.1	Decomposição em Valores Singulares (SVD)	21
3.2	Pseudo Inversa e a Solução de Quadrados Mínimos	25
3.3	A Condição de Picard e a Instabilidade	27
4	PROBLEMAS INVERSOS DISCRETOS	31
5	TÉCNICAS DE REGULARIZAÇÃO PARA PROBLEMAS INVERSOS LINEARES	35
5.1	SVD Truncada – TSVD	36
5.2	Regularização de Tikhonov	36
6	ESCOLHA DO PARAMÊTRO DE REGULARIZAÇÃO	39
6.1	Princípio de Discrepância de Morozov	40
6.2	Curva-L	41
7	APLICAÇÕES	47
7.1	Equação Integral de Fredholm de Primeiro Tipo	47
7.2	Restauração de Sinais Unidimensionais	56
8	CONCLUSÃO	65
	Referências	67
A	IMPLEMENTAÇÕES. PROBLEMAS MAL POSTOS	69

B	IMPLEMENTAÇÕES. ESTABILIDADE	71
C	IMPLEMENTAÇÕES. PROBLEMAS INVERSOS DISCRETOS	73
D	IMPLEMENTAÇÕES. APLICAÇÕES: INTEGRAL DE PRIMEIRA ES- PÉCIE	75
E	IMPLEMENTAÇÕES. APLICAÇÕES: SINAL UNIDIMENSIONAL	87

1 INTRODUÇÃO

Problemas inversos são frequentemente classificados como problemas mal postos, uma consequência disso é que o operador associado a esse problema é mal condicionado, isso associado ao fato de que normalmente os dados conhecidos são coletados com algum nível de imprecisão, o efeito observado são grandes alterações na solução. Com isso em mente, técnicas de regularização são fundamentais para que seja possível “trocar” o problema original mal posto por um problema aproximado bem posto.

A ideia central das técnicas de regularização é incorporar ao problema inicial informações que amenizem o nível de perturbação nos dados para que seja possível encontrar uma solução estável e viável. Esse assunto é relevante visto que suas principais aplicações se encontram em áreas como medicina, geofísica, meteorologia, entre outras, nas quais a precisão da solução é fundamental.

Dessa maneira, serão apresentados métodos de regularização clássicos, como: a Regularização TSVD, um método bastante simples mas eficiente para problemas de pequeno porte, e a Regularização de Tikhonov, uma técnica bastante conceituada e que, ao longo dos anos, deu origem a muitos outros métodos. A partir desse ponto, serão avaliadas suas vantagens e desvantagens, serão discutidas também técnicas de determinação de parâmetros de regularização em problemas inversos discretos lineares.

Por se tratar de um assunto bastante abrangente, e este estudo ter um aspecto introdutório ao tema, a metodologia adotada parte de uma revisão bibliográfica abordando não apenas as técnicas de regularização e de determinação de parâmetros, como também destacando os conceitos fundamentais que cercam os problemas inversos discretos. Por fim, serão apresentados exemplos resolvidos computacionalmente que permitem a comparação dos métodos citados ao longo do texto.

Dessa forma, este trabalho é dividido em seis partes: no Capítulo 2 serão introduzidos os problemas inversos e através da caracterização dos problemas mal postos, serão apresentadas as motivações para o estudo das técnicas de regularização. Já o Capítulo 3 será dedicado à fundamentação teórica de tópicos essenciais na Álgebra Linear para o entendimento das técnicas de regularização: a Decomposição em Valores Singulares (SVD) e a Pseudo-Inversa, além disso serão apresentadas a Condição de Picard e a sua relação com a estabilidade da so-

lução. Como citado anteriormente, visto que o tema é bastante abrangente, este estudo inicia tratando dos Problemas Inversos de uma forma geral, porém no Capítulo 4 os problemas inversos discretos serão apresentados e será introduzido o exemplo de problema mal condicionado que será abordado ao longo do texto. Os Capítulos 5 e 6 discutirão a teoria dos Métodos de Regularização SVD e Tikhonov, e as técnicas de Determinação de Parâmetros de Regularização de Morozov e Curva-L, respectivamente. Por fim, no Capítulo 7 os métodos expostos serão implementados computacionalmente no software OCTAVE e aplicados na resolução de dois problemas: a resolução de uma integral de Fredholm de primeiro tipo e a restauração de um sinal unidimensional, o que permite analisar e comparar computacionalmente o desempenho dos métodos estudados. As implementações computacionais realizadas serão encontradas nos Apêndices deste trabalho.

2 PROBLEMAS INVERSOS

Diversos modelos matemáticos são originados a partir de observações de fenômenos do tipo causa e efeito. Logo, estabelecido um padrão de causas iniciais temos um sistema que leva a um conjunto de consequências. Matematicamente, isso poderia ser descrito da seguinte forma, f é uma função de entrada pertencente ao conjunto \mathcal{F} , e \mathcal{A} um operador do tipo $\mathcal{A}(f) = g$, onde g é a função do efeito produzido pertencente ao conjunto \mathcal{G} .

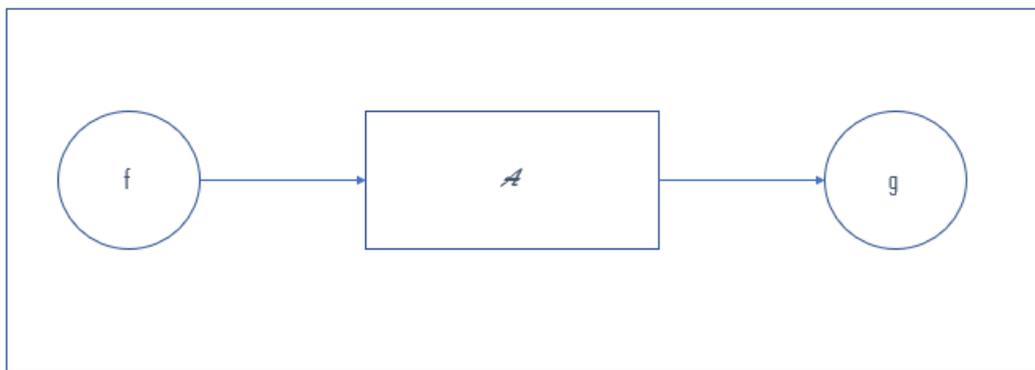


Figura 1 – Modelagem de um processo – Adaptado de BAZÁN and BORGES (2009) (Fonte: Autora)

Um processo onde o problema é encontrar g , conhecidos inicialmente o modelo \mathcal{A} e a função f , é chamado de **problema direto** (BAZÁN and BORGES (2009)). Para problemas diretos, temos o operador \mathcal{A} bem definido e contínuo, além disso para cada causa f registrada existe um único efeito g observado, esta relação pode ser descrita através da seguinte formulação:

$$\mathcal{A}(f) = g \quad (2.1)$$

Já um **problema inverso** pode ser visto como o caminho oposto ao de um problema direto (BAZÁN and BORGES (2009)). Isto é, dado um efeito final g e um modelo \mathcal{A} , defina a causa inicial f . Ou ainda, em posse de f e g qual é a transformação \mathcal{A} que melhor define este processo observado. Genericamente, podemos formular um problema inverso da seguinte forma:

$$f = \mathcal{A}^{-1}g \quad (2.2)$$

Como citado inicialmente muitos fenômenos são descritos através de problemas inversos, algumas aplicações típicas de problemas inversos podem ser encontradas em: imagens de tomografia, estudos sobre radioterapia, na área da astronomia é utilizada para estudos da

Radiação de Corpo Negro, em leitura e interpretação de sinais, tratamento e reconstrução de imagens, medições geofísicas, etc.

Na maioria dos casos os problemas são modelados através de equações integrais de primeiro tipo e lineares, esses modelos são simplificações de representações físicas reais (GROETSCH (1993)).

Mas o que torna os problemas inversos interessantes de serem estudados diz respeito à solução. Em problemas diretos é assumida a existência, unicidade e estabilidade da solução, essas propriedades porém não são garantidas em um problema inverso, classificando o na categoria de **problema mal posto** (GROETSCH (1993)).

2.1 PROBLEMAS MAL POSTOS

Enquanto estudava problemas de valor de contorno, o matemático francês Hadamard estabeleceu três condições necessárias para que um problema seja bem-posto (BAZÁN and BORGES (2009)), são elas:

1. Existência: Para cada $g \in \mathcal{G}$ existe $f \in \mathcal{F}$ tal que $\mathcal{A}(f) = g$.
2. Unicidade: Para cada $g \in \mathcal{G}$, $\mathcal{A}(f) = g$ possui apenas uma solução.
3. Estabilidade: f depende continuamente dos dados, isto é, \mathcal{A}^{-1} é contínua.

Se alguma dessas condições falha, o problema é dito mal posto. O sistema linear apresentado abaixo é mal-posto, pois não há nenhum valor de x que satisfaça simultaneamente ambas as equações (HANSEN (2010)):

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix} x = \begin{pmatrix} 1 \\ 2.2 \end{pmatrix}.$$

Outro exemplo de problema mal-posto são os sistemas indeterminados, veja que $x_1 + x_2 = 1$ não possui uma única solução. Geometricamente, temos uma reta de soluções possíveis.

Quando um dos critérios não é atendido, o problema precisa ser reformulado para que uma solução possa ser encontrada. Observe que, para o primeiro sistema, não há uma solução definida inicialmente. No entanto, ao reformular o problema como um problema de mínimos quadrados, é possível determinar uma solução única. No caso do segundo sistema, ao

impor a restrição de que $\|x\|_2 = (x_1^2 + x_2^2)^{\frac{1}{2}}$ seja mínima, também é possível encontrar uma solução única.

No que diz respeito a estabilidade, a situação é um pouco mais delicada. Em problemas mal condicionados, pequenas perturbações nos dados iniciais geram grandes diferenças na solução. Portanto, é necessário cuidados extras para que a solução aproximada seja uma aproximação da solução exata e não um ajuste dos ruídos. O exemplo a seguir, ilustra uma questão importante a respeito da má condição dos problemas inversos e a aproximação da solução.

Suponha:

$$A = \begin{pmatrix} 1 & 2.1 \\ 2.1 & 3.99 \end{pmatrix}, \quad b = A \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 3.1 \\ 6.09 \end{pmatrix}$$

Considerando $[-0.01 \quad -0.09]^T$ o vetor de perturbação, o qual influencia o vetor b gerando um novo vetor com ruído de perturbação:

$$b_{\text{pert}} = \begin{pmatrix} 3.09 \\ 6 \end{pmatrix}.$$

Pela solução de mínimos quadrados, ou seja, $x = (A^T A)^{-1} A^T b_{\text{pert}}$ é possível obter:

$$x_{QM} = \begin{pmatrix} 0.6450 \\ 1.1164 \end{pmatrix}$$

$$\|Ax_{QM} - b\|_2 = 4.4295 \times 10^{-13}, \quad \frac{\|Ax_{QM} - b\|_2}{\|b\|_2} = 6.4767 \times 10^{-14}.$$

A solução possui resíduos pequenos, veja que a norma do resíduo e a norma do resíduo relativo estão na casa da 10^{-13} e 10^{-14} , porém x_{QM} está longe do valor exato $x = [1 \quad 1]^T$ do problema inverso. Calculando a norma do erro da solução isso é visível:

$$\|x_{QM} - x_{\text{exato}}\| = 0.3912$$

$$\frac{\|x_{QM} - x_{\text{exato}}\|}{\|x_{\text{exato}}\|} = 0.2766$$

Na figura 2, as duas soluções foram graficadas; em azul, a solução exata do problema $Ax = b$, $[1 \quad 1]^T$ e em vermelho a solução do problema com os dados iniciais perturbados $Ax = b_{\text{pert}}$, x_{QM} . Na figura 3, estão representados os dois vetores b e b_{pert} , veja que existe pouca

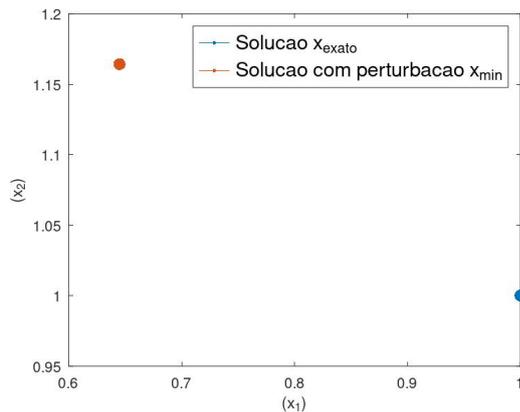


Figura 2 – Soluções do problema mal posto $Ax = b$ (Fonte: Autora)

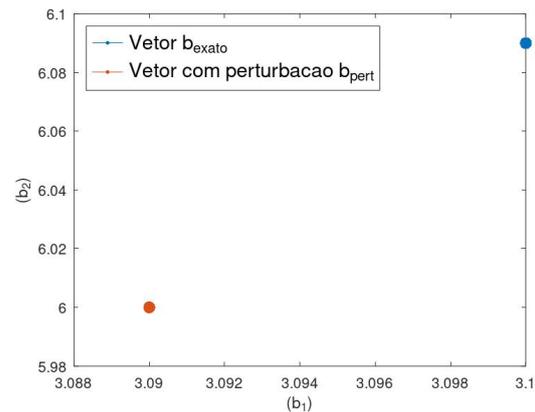


Figura 3 – Vetores b e b_{pert} (Fonte: Autora)

diferença entre ambos, ou seja, a mínima variação nos dados iniciais do problema causou uma instabilidade na solução final.

Observando os gráficos é evidente que os resíduos pequenos das soluções não são garantia de uma boa aproximação da solução, o efeito ilustrado na figura é causado pelo mal condicionamento de A , onde $\text{cond}(A) = 61.270$, nos capítulos seguintes abordaremos mais esse tópico.

Portanto, no que diz respeito à estabilidade, a reformulação do problema se torna mais complexa e exige métodos mais sofisticados. É nesse contexto que os métodos de regularização, que são o foco principal deste trabalho, se tornam essenciais.

2.2 ESTABILIDADE

Como já citado a estabilidade é uma propriedade fundamental que garante que erros e perturbações não cresçam descontroladamente durante o processo de solução de problemas. De maneira geral, um sistema é considerado estável se pequenas alterações nos dados de entrada não resultarem em grandes variações nos resultados calculados. Esta característica é crucial para a confiabilidade, pois assegura que as soluções obtidas sejam sensíveis apenas às verdadeiras características do problema e não aos erros acumulados durante os cálculos. O exemplo a seguir, ilustra essa situação e serve como motivação para os tópicos que serão explorados nos próximos capítulos. Para isso, considere a Integral de Fredholm de primeira espécie:

$$\int_0^1 k(s, t)f(t)dt = g(s), \quad 0 \leq s \leq 1, \quad (2.3)$$

em um problema direto, $g(s)$ é influenciada pelas propriedades e restrições das funções f e k , desta forma a existência da solução $g(s)$ depende dos valores das demais funções associadas ao problema. O mesmo pode ser dito para problemas inversos, se a busca for por $f(t)$ que satisfaça a expressão de igualdade da integral, então suas características serão vinculadas as funções $k(s, t)$ e $g(s)$, ([GROETSCH \(1993\)](#)).

Exemplo 1. Suponha $k(s, t) = 1$ para todo $(s, t) \in [0, 1] \times [0, 1]$.

Então a solução para a integral de Fredholm existe apenas se $g(s)$ é uma constante, isto porque f não depende da variável s . Desta forma,

$$\int_0^1 f(t)dt = C.$$

Note ainda, que C pode assumir qualquer valor. Suponha $C = 1$, então o grupo de funções polinômias do tipo:

$$f(t) = nt^{n-1}, n = 1, 2, \dots$$

formam um conjunto de soluções para a equação:

$$\int_0^1 f(t)dt = 1.$$

Portanto, a equação integral é um problema mal posto, pois não há unicidade.

Afim de ajudar na análise do próximo exemplo será importante lembrar o Lema de Riemann-Lebesgue.

Lema 1 (Lema de Riemman - Lebesgue). Seja $f : \mathbb{R} \rightarrow \mathbb{C}$ uma função pertencente ao espaço \mathcal{L}^1 e de período 1, então:

$$\lim_{n \rightarrow \infty} \int_0^1 f(t) \cos(nt)dt = \lim_{n \rightarrow \infty} \int_0^1 f(t) \sin(nt)dt = 0, \quad \text{onde } n > 0 \text{ e } n \in \mathbb{Z}.$$

Ou seja, a transformada de Fourier de f converge para zero em altas frequências (quando n tende ao infinito).

Com isso em mente, continuando com a integral de Fredholm.

Exemplo 2. Suponha $k(s, t) = \sqrt{s} \sin \pi t$ e $g(s) = 0$, então a integral de Fredholm tem a forma:

$$\int_0^1 \sqrt{s} \sin(\pi t)x(t)dt = 0.$$

E um conjunto de soluções pode ser dado pela função, sendo $n \neq 1$:

$$x(t) = \sin(n\pi t).$$

Uma das características mais notáveis das equações integrais de Fredholm de primeira espécie é a instabilidade. Como o núcleo da integral de Fredholm resulta em uma função de quadrado-integrável, podemos aplicar o Lema de Riemann-Lebesgue para analisar seu comportamento (SILTANEN (2009) e HANSEN (2010)). Note,

$$g_n(s) = \int_0^1 k(s, t) \sin(n\pi t) dt \rightarrow 0 \text{ quando } n \rightarrow \infty.$$

Isto significa que a frequência de f aumenta conforme n cresce, mas o efeito em $g_n(s)$ é um decrescimento da amplitude, o que pode ser observado nas figuras 4 e 5, isto é, quanto maior a frequência da f , maior a atenuação na amplitude de $g_n(s)$. Logo, o problema inverso, recuperar ou obter f a partir de $g_n(s)$, será bem mais complexo devido a dificuldade de obter as componentes de f associada às altas frequências devido à amenização das amplitudes dadas.

Exemplo 3. Se $f_n(t) = \sin(2\pi nt)$ e $k_n(s, t) = e^{s \cos(2\pi nt)}$, onde $n \in \mathbb{N}$, então $g_n(s) = \frac{\sinh(s)}{2\pi ns}$ (PAREDES et al. (2014)).

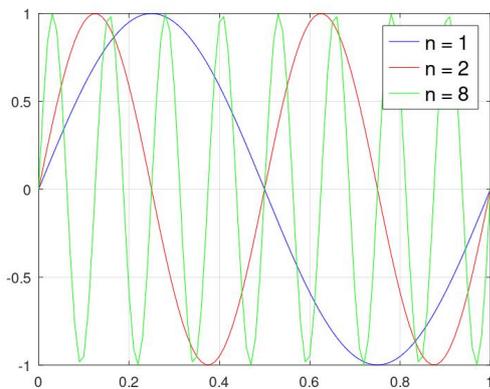


Figura 4 – $f_n(t) = \sin(2\pi nt)$ – Adaptado de HANSEN (2010) (Fonte: Autora)

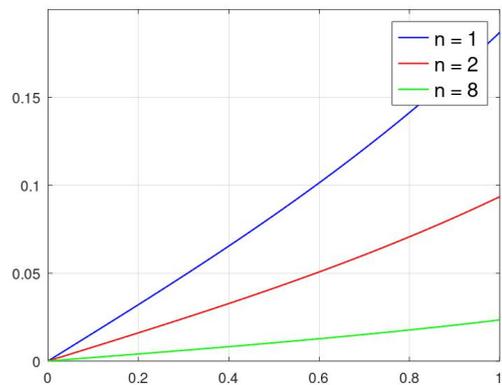


Figura 5 – $g_n(s) = \frac{\sinh(s)}{2\pi ns}$ – Adaptado de HANSEN (2010) e PAREDES et al. (2014) (Fonte: Autora)

3 FUNDAMENTAÇÃO TEÓRICA

Este capítulo será dedicado à fundamentação dos conceitos necessários para introduzir as técnicas de resolução dos problemas inversos. Além disso, muito foi falado sobre a solução em um problema inverso, os tópicos a seguir esclarecerão quais fatores influenciam essas características e os motivos por trás disso.

A estratégia comumente utilizada para aproximar uma solução de um problema inverso consiste em aproximar o operador inverso através de uma família de operadores bem postos, com o propósito de encontrar a melhor aproximação possível para a solução do problema (DE CEZARO and DE CEZARO (2012)). Desta forma, serão expostos a seguir dois tópicos essenciais na Álgebra Linear: a Decomposição em Valores Singulares (SVD) e a Pseudo-Inversa. Com isso será possível encontrar uma solução para o problema inverso decompondo o operador.

3.1 DECOMPOSIÇÃO EM VALORES SINGULARES (SVD)

Para iniciar a discussão a respeito da fatoração de A , é necessário introduzir um dos teoremas mais importantes da Álgebra Linear, o Teorema Espectral, do qual a SVD é uma consequência. As referências utilizadas para esta seção foram: LIMA (2014), BURDEN (2016) e STRANG (2014).

Em poucas palavras, o Teorema Espectral garante que todo espaço vetorial de dimensão finita, com produto interno bem definido, possui uma base que fature este operador.

Teorema 1 (Teorema Espectral). *Para todo operador autoadjunto ¹ $A : E \rightarrow E$, num espaço vetorial C de dimensão finita $n \neq 0$, munido de produto interno, existe uma base ortonormal $\{u_1, \dots, u_n\} \subset E$ formada por autovetores de A .*

No caso particular, em que A é uma matriz simétrica de ordem n , o teorema acima garante a existência de um conjunto de n autovetores linearmente independente (LI), mais ainda, um conjunto de vetores LI ortogonais que formam uma base. Em consequência disso A é

¹ Se $T : X \rightarrow X$ é um operador linear autoadjunto se $T = T^*$, ou seja, quando

$$\langle Tx, y \rangle = \langle x, Ty \rangle, \quad \forall x, y \in X.$$

ortogonalmente diagonalizável, ou seja, $A = QDQ^{-1}$, por definição Q é a matriz de autovetores e D é a matriz composta pelos autovalores associados na sua diagonal principal.

Ainda, para casos gerais os seguintes resultados são válidos:

Teorema 2. *Seja A uma transformação linear de tamanho $m \times n$.*

- (i) *As matrizes $A^T A$ e AA^T são simétricas,*
- (ii) *$\dim \mathcal{N}(A) = \dim \mathcal{N}(A^T A)$.*
- (iii) *$\dim \mathcal{Im}(A) = \dim \mathcal{Im}(A^T A)$.*
- (iv) *Os autovalores de $A^T A$ são reais e não negativos.*
- (v) *Os autovalores não nulos de AA^T são os mesmos que os autovalores não nulos de $A^T A$.*

Assim, dada uma matriz A , de tamanho $m \times n$, é possível fatorar este operador através da decomposição em valores singulares, que possui a forma:

$$A = U\Sigma V^T, \tag{3.1}$$

onde:

- U é uma matriz ortogonal de tamanho $m \times m$, derivada de AA^T ,
- V é uma matriz ortogonal de tamanho $n \times n$, derivada de $A^T A$,
- Σ é uma matriz diagonal de tamanho $m \times n$, também derivada de $A^T A$.

Iniciando por Σ , onde os elementos da sua diagonal principal são os autovalores, reais e não negativos, associados à matriz $A^T A$. Ordenados de forma que o maior autovalor esteja na primeira coluna de Σ , conforme abaixo:

$$\sigma_1^2 \geq \dots \geq \sigma_k^2 \geq \dots \geq \sigma_n^2.$$

As raízes quadradas positivas desses autovalores de $A^T A$ resultam nos elementos da diagonal de Σ , os quais são chamados valores singulares de A . Supondo $n > m$ então Σ é conforme a seguir:

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \sigma_2 & \ddots & \vdots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \sigma_n & 0 & 0 \end{bmatrix}$$

Visto que $A^T A$, de tamanho $n \times n$ é simétrica, sabe-se que: $A^T A = V D V^T$, onde D possui em sua diagonal principal os autovalores associados a matriz $A^T A$ e V é a matriz ortogonal com os autovetores associados a $A^T A$.

Por fim, U , $m \times m$, é definida através dos r valores singulares não nulos e aos autovetores de V , conforme abaixo:

$$u_i = \frac{1}{\sigma_i} A v_i, \text{ com } i = 1, \dots, r.$$

Veja que se $r \leq m$, então $n < m$ portanto é necessário encontrar as demais $m - r$ colunas de U linearmente independentes e ortonormais. Isso porque Σ é representada da seguinte maneira:

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \dots & \sigma_n \\ \vdots & \dots & \dots & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Uma forma alternativa de encontrar a matriz U , se dá através da definição de um conjunto de autovetores ortonormais associados a matriz AA^T .

O resultado final é $Av_j = \sigma_j u_j$. Veja o porquê a seguir, mas antes considere $A^T Av_j = \sigma_j^2 v_j$.²

$$\begin{aligned} A^T Av_j &= \sigma_j^2 v_j \\ AA^T Av_j &= \sigma_j^2 Av_j \\ (AA^T)(Av_j) &= \sigma_j^2 Av_j \end{aligned}$$

² Por definição:

$$\begin{aligned} AA^T &= (U \Sigma V^T)(V \Sigma^T U^T) = U \Sigma \Sigma^T U^T, \\ A^T A &= (V \Sigma^T U^T)(U \Sigma V^T) = V \Sigma^T \Sigma V^T. \end{aligned}$$

isto é, Av_j é autovetor de AA^T . E ainda, σ_j é o módulo desse autovetor, veja que:

$$\begin{aligned} A^T Av_j &= \sigma_j^2 v_j \\ v_j^T A^T Av_j &= \sigma_j^2 v_j^T v_j \\ \|Av_j\|^2 &= \sigma_j^2 \end{aligned}$$

dai, $\frac{Av_j}{\sigma_j} = u_j$. Resultando em:

$$\begin{aligned} AV &= U\Sigma \\ \iff A &= U\Sigma V^T \\ \iff A &= \sum_{j=1}^r \sigma_j u_j v_j^T \end{aligned}$$

Todos os passos anteriores, se resumem no seguinte teorema:

Teorema 3 (Teorema dos Valores Singulares). *Seja $A : E \rightarrow F$ uma transformação linear de posto³ $r > 0$ entre espaços vetoriais de dimensão finita com produto interno. Existem bases ortonormais $\{u_1, \dots, u_n\} \subset E$ e $\{v_1, \dots, v_m\} \subset F$ tais que $Au_i = \sigma_i v_i$, $Av_i = \sigma_i u_i$, com $\sigma_i > 0$ para $i = 1, \dots, r$ e $Au_i = 0$, $A^*v_i = 0$ se $i > r$.*

Referente a estabilidade, através da ordem de grandeza entre os valores singulares, define-se a razão chamada de **número de condição**. Suponha, que Σ uma matriz quadrada de tamanho $n \times n$ não singular, portanto todos os seus valores singulares são não nulos e positivos o que garante a existência e a unicidade da solução do problema $Ax = b$. Invertendo a matriz A , o resultado é:

$$A^{-1} = V\Sigma^{-1}U^T,$$

note que Σ^{-1} é resultado da inversão da diagonal principal de Σ , portanto:

$$\Sigma^{-1} = \begin{bmatrix} \frac{1}{\sigma_1} & 0 & \dots & 0 \\ 0 & \frac{1}{\sigma_2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \frac{1}{\sigma_n} \end{bmatrix}$$

O número de condição da matriz é definido pela razão entre o maior e menor valor singular, assim:

$$\text{cond}(A) = \frac{\sigma_1}{\sigma_n},$$

³ Posto: dimensão da $\text{Im}(A)$.

se σ_n é pequeno frente a σ_1 então o número de condição de A é suficientemente grande, portanto A é mal condicionada. Ainda, através do número de condição da matriz que a propagação do erro relativo na solução com respeito ao erro relativo nos dados é controlada. Quando resolve-se um problema perturbado, ou seja, $Ax_{\text{exato}} = b_{\text{exato}} + e$ obtemos:

$$\frac{\|x_{\text{exato}} - x\|}{\|x_{\text{exato}}\|} \leq \text{cond}(A) \frac{\|e\|}{\|b_{\text{exato}}\|},$$

onde e é o ruído nos dados e $x_{\text{exato}} - x$ é a variação na solução. Se $\text{cond}(A)$ é pequeno, o problema é bem-condicionado e a solução é estável. Porém, se $\text{cond}(A)$ é grande, o problema é mal-condicionado e instável.

Tendo isto em vista, inverter A não é o melhor caminho levando em consideração a amplificação dos erros de truncamento que podem ocorrer devido a ordem de grandeza dos valores singulares. Além disso, é necessário garantir existência e unicidade para a solução, e isso será possível através da **Pseudo-Inversa**.

3.2 PSEUDO INVERSA E A SOLUÇÃO DE QUADRADOS MÍNIMOS

Se $A : \mathcal{H}_1 \rightarrow \mathcal{H}_2$, um operador linear limitado, com \mathcal{H}_1 e \mathcal{H}_2 espaços de Hilbert ⁴, é possível definir um problema linear da seguinte forma:

$$Ax = y. \tag{3.2}$$

Veja que se A possui inversa, ou seja, existe A^{-1} então existe uma solução única para o problema exposto. Porém, podem ocorrer casos em que x encontrado pode não ser único quando $\mathcal{N}(A) \neq 0$, ou ainda, x pode não existir se $y \notin \mathcal{I}m(A)$ (DE CEZARO and DE CEZARO (2012)). Para contornar esta situação, é possível definir uma solução generalizada para o problema. suponha que A não só é um operador linear limitado como também $\mathcal{I}m(A)$ é fechada em \mathcal{H}_2 .

É possível definir $P : \mathcal{H}_2 \rightarrow \mathcal{I}m(A)$, operador de projeção ortogonal convexo e fechado (compacto), tal que $Py \in \mathcal{I}m(A)$ é o vetor mais próximo de y (DE CEZARO and DE CEZARO (2012)).

⁴ Espaço de Hilbert é um espaço vetorial completo com um produto interno. Por exemplo, o espaço Euclidiano \mathbb{R}^n , o espaço de funções quadrado-integráveis L^2 – como visto no primeiro capítulo, e o espaço de seqüências quadrado-integráveis l^2 .

Desta forma, uma solução generalizada para $Ax = y$, é qualquer solução $u \in \mathcal{H}_1$ da equação:

$$Ax = Py. \quad (3.3)$$

Geometricamente, busca-se encontrar $\tilde{x} \in \mathcal{H}_1$ que é solução do problema de minimização

$$\tilde{x} = \arg \min_{x \in \mathcal{H}_1} \frac{1}{2} \|Ax - y\|^2. \quad (3.4)$$

Teorema 4. *Se $A : \mathcal{H}_1 \rightarrow \mathcal{H}_2$ é um operador linear limitado, $y \in \mathcal{H}_2$, $P : \mathcal{H}_2 \rightarrow \mathcal{I}m(A)$ um operador de projeção ortogonal e $\mathcal{I}m(A)$ é fechada em \mathcal{H}_2 . Então, as seguintes condições sobre $\tilde{x} \in \mathcal{H}_1$ são equivalentes:*

(i) $A\tilde{x} = Py$,

(ii) $\tilde{x} = \arg \min_{x \in \mathcal{H}_1} \frac{1}{2} \|Ax - y\|^2$

(iii) $A^*A\tilde{x} = A^*y$.

Assim se $\tilde{x} \in \mathcal{H}_1$ satisfaz qualquer uma das condições do teorema anterior, então \tilde{x} é uma solução de quadrados mínimos da equação $Ax = y$. Além disso, garante-se a existência de uma solução para todo $y \in \mathcal{H}_2$. A demonstração do teorema anterior pode ser encontrada em [DE CEZARO and DE CEZARO \(2012\)](#).

Porém, além da existência, garantir a unicidade como visto anteriormente possui sua relevância. Portanto, é necessário escolher dentre todas as soluções obtidas $y \in \mathcal{H}_2$ uma que possua uma característica particular, a solução de norma mínima ([STRANG \(2014\)](#)).

Desta forma, com A satisfazendo as condições já estabelecidas até aqui, a aplicação $A^\dagger : \mathcal{H}_1 \rightarrow \mathcal{H}_2$, definida por:

$$A^\dagger y = \tilde{x}, \quad (3.5)$$

indica como obter \tilde{x} , sendo \tilde{x} o único vetor solução de quadrados mínimos com norma mínima e A^\dagger a **pseudo inversa** de A . Vale ressaltar que:

1. Se A possui uma inversa, então $A^\dagger = A^{-1}$
2. Sendo $\alpha \neq 0$ então $(\alpha A)^\dagger = \alpha^{-1} A^\dagger, \forall \alpha \in \mathbb{R}^*$,
3. $(A^\dagger)^\dagger = A$,
4. $(A^T A)^\dagger = A^\dagger (A^\dagger)^T$ e $(A A^T)^\dagger = (A^\dagger)^T A^\dagger$.

Agora, através da Decomposição em Valores Singulares e das propriedades da pseudo inversa, podemos obter a seguinte expressão:

$$\begin{aligned}
 A &= U\Sigma V^T \\
 A^\dagger &= (U\Sigma V^T)^\dagger \\
 &= (V^T)^\dagger \Sigma^\dagger U^\dagger \\
 &= V\Sigma^\dagger U^\dagger \\
 \implies A^\dagger &= V\Sigma^\dagger U^T
 \end{aligned} \tag{3.6}$$

isso, porque U e V são ortogonais, e, assim $V^T = V^{-1}$ e $U^T = U^{-1}$. E ainda, Σ^\dagger é obtida através da transposição de Σ e a inversão dos valores singulares não nulos.

Desta forma, é garantida a existência e a unicidade da solução de mínimos quadrados de norma mínima.

3.3 A CONDIÇÃO DE PICARD E A INSTABILIDADE

Anteriormente, foi apresentado através de exemplos o impacto da instabilidade em problemas inversos. Agora, a discussão será voltada para a relação entre a instabilidade e os valores singulares.

Se A não é um operador auto-adjunto, podemos considerar a decomposição em valores singulares como mencionado no capítulo anterior.

Teorema 5. *Seja A , um operador linear compacto, e \mathcal{F}, \mathcal{G} espaços de Hilbert, onde $A \in \mathcal{K}(\mathcal{F}, \mathcal{G})$. Então existe,*

1. *Uma sequência, não necessariamente infinita, $\{\sigma_j\}_{j \in \mathbb{N}}$ com $\sigma_1 \geq \sigma_2 \geq \dots \geq 0$,*
2. *Uma base ortonormal $\{u_j\}_{j \in \mathbb{N}} \in \mathcal{F}$ do $N(A)^\perp$,*
3. *Uma base ortonormal $\{v_j\}_{j \in \mathbb{N}} \in \mathcal{G}$ da $\overline{\text{Im}(A)}$, com*

$$Au_j = \sigma_j v_j, \quad A^* v_j = \sigma_j u_j, \quad \forall j \in \mathbb{N}.$$

Além disso, para todo $f \in \mathcal{F}$ a seguinte representação é válida:

$$Af = \sum_{j=1}^{\infty} \sigma_j \langle f, u_j \rangle v_j.$$

Define-se a $\{(\sigma_j, u_j, v_j)\}$ como o sistema singular de A .

Para a demonstração ver [KEKKONEN and KOROLEV \(2019\)](#).

Suponha então o sistema $Af = g$, a partir do Teorema Espectral, se o operador linear A é compacto e possui inversa, o teorema a seguir, chamado de Condição de Picard, garante uma condição necessária e suficiente para a existência de uma solução para o problema $Af = g$.

Teorema 6 (Condição de Picard). *Seja A , definido conforme o teorema anterior e $g \in \overline{\mathcal{I}m}(A)$. Então $g \in \mathcal{I}m(A)$ se, e somente se, o critério de Picard é satisfeito*

$$\sum_{j=1}^{\infty} \left(\frac{1}{\sigma_j} \langle v_j, g \rangle \right)^2 < \infty. \quad (3.7)$$

Demonstração 1. (\implies) *Se $g \in \mathcal{I}m(A)$, então existe $f \in \mathcal{F}$ tal que $Af = g$. Veja que*

$$\langle g, v_j \rangle = \langle Af, v_j \rangle = \langle f, A^*v_j \rangle = \sigma_j \langle f, u_j \rangle.$$

Daí,

$$\sum_{i=1}^{\infty} \frac{|\langle g, v_j \rangle|^2}{\sigma_j^2} = \sum_{i=1}^{\infty} \frac{|\sigma_j \langle f, u_j \rangle|^2}{\sigma_j^2} = \sum_{i=1}^{\infty} |\langle f, u_j \rangle|^2$$

ainda, pela Desigualdade de Bessel: ⁵

$$\sum_{i=1}^{\infty} \frac{|\langle g, v_j \rangle|^2}{\sigma_j^2} = \sum_{i=1}^{\infty} |\langle f, u_j \rangle|^2 \leq \|f\|^2 < \infty.$$

(\impliedby) *Agora, definindo $f := \sum_{i=1}^{\infty} \frac{\langle g, v_j \rangle u_j}{\sigma_j} \in \mathcal{F}$. Pelo critério de Picard é bem definido, assim*

$$Af = \sum_{i=1}^{\infty} \frac{\langle g, v_j \rangle}{\sigma_j} Au_j = \sum_{i=1}^{\infty} \frac{\langle g, v_j \rangle}{\sigma_j} \sigma_j v_j = \sum_{i=1}^{\infty} \langle g, v_j \rangle v_j.$$

Note que chegamos à definição de projeção. Assim, sendo $v_j \in \{v_j\}$ uma base ortonormal de $\overline{\mathcal{I}m}(A)$, e ainda, por hipótese $g \in \overline{\mathcal{I}m}(A)$, logo obtemos,

$$P_{\overline{\mathcal{I}m}(A)}g = \sum_{i=1}^{\infty} \langle g, v_j \rangle v_j = g.$$

Assim, $g \in \mathcal{I}m(A)$. Com isso, está concluída a demonstração do teorema.

⁵ Uma sequência de vetores $\{f_\alpha\}_{\alpha \in A} \subseteq \mathcal{H}$ é dita ortogonal quando $\langle f_\alpha, f_\beta \rangle = 0$ para $\alpha \neq \beta$. Ainda, se $\|f_\alpha\| = 1$ para cada $\alpha \in A$, então a sequência é ortonormal. Veja que se $\{f_\alpha\}$ é uma sequência ortonormal, então para qualquer $f \in \mathcal{H}$, o produto interno $\langle f, f_\alpha \rangle$ é não nulo e pela Desigualdade de Bessel

$$\sum_{\alpha \in A} |\langle f, f_\alpha \rangle|^2 \leq \|f\|^2.$$

□

Veja que pelo critério de Picard, se $\sigma_j \rightarrow 0$ quando $j \rightarrow \infty$, observa-se o efeito da instabilidade causada pelos valores singulares do operador A , pois $\lim_{j \rightarrow \infty} \frac{1}{\sigma_j^2} = \infty$. Além disso, indica que as sequências geradas pelo produto interno $\langle v_j, g \rangle$ devem decrescer mais rápido, para conter esse crescimento dos fatores definidos pelos valores singulares.

Por fim, com os resultados anteriores, se $g \in \mathcal{I}m(A)$ e A possui inversa, é possível expressar a solução do problema inverso da seguinte maneira:

$$A^\dagger g = \sum_{j=1}^{\infty} \frac{1}{\sigma_j} \langle v_j, g \rangle u_j. \quad (3.8)$$

Caso não exista garantia de que A possui inversa, utilizando a pseudo-inversa de A e considerando uma soma finita de termos, ou seja,

$$A^\dagger g = \sum_{j=1}^k \frac{1}{\sigma_j} \langle v_j, g \rangle u_j, \quad (3.9)$$

a Condição de Picard sempre será satisfeita, isto porque uma soma finita de elementos sempre terá a convergência garantida. Porém, acompanhar o decrescimento dos coeficientes $\langle v_j, g \rangle$ em relação aos valores singulares σ_j não é menos importante. É possível observar que a componente de g na direção de v_j é amplificado através do fator σ_j . Portanto, se existir em g componentes com alguma perturbação na direção dos vetores singulares associados aos menores valores singulares, esses ruídos serão também ampliados. Agora, se as componentes $\langle v_j, g \rangle$ tenderem para zero com mais velocidade do que os coeficientes σ_j , então é satisfeita a chamada **Condição Discreta de Picard**, ou seja, existindo um controle do crescimento dos valores singulares σ_i , então assim o problema de amplificar componentes ruidosas é amenizado. A seguir, a definição formal da Condição Discreta de Picard.

Definição 1. *g satisfaz a **Condição Discreta de Picard** para o problema $Af = g$, se os coeficientes $|v_j^T g|$ decaem para zero mais rápido do que os valores singulares σ_j de A , onde v_j são os vetores singulares de A .*

Ao observar a questão da estabilidade, indicada pela Condição Discreta de Picard temos a motivação inicial para entender o fundamento das técnicas de regularização que tem por objetivo amenizar o efeito das altas frequências causadas pelos valores singulares muito pequenos.

4 PROBLEMAS INVERSOS DISCRETOS

No capítulo 1, foram fundamentados e caracterizados os problemas inversos de maneira geral. Porém, computacionalmente resolver uma equação integral de primeira espécie do tipo

$$\int_a^b k(s, t) f(t) dt = g(s), \quad c \leq s \leq d, \quad (4.1)$$

exige que esse problema seja discretizado, ou seja, o problema deve ser levado de um conjunto de dados contínuo para um conjunto de dados discreto. Em resumo, a discretização irá transformar dados contínuos em valores discretos, a partir da discretização do domínio. Para a integral de primeira espécie utilizam-se regras de quadratura, por exemplo

$$\sum_{j=1}^n w_j k(s, t_j) f_j = g(s) \quad (4.2)$$

onde w_j são os pesos de quadratura e t_j são os pontos de quadratura. Além disso, x_j são aproximações para $x(t_j)$. Porém, s permanece contínua, ou seja, a equação anterior representa um sistema que restringe resultados para infinitos valores de s (GROETSCH (1993)).

O próximo passo é restringir os valores de s para alguns pontos específicos através de um método de colocação (BAZÁN and BORGES (2009)), isto é, a soma do lado direito será igualada a uma série de pontos s_j , conforme abaixo

$$\sum_{j=1}^n w_j k(s_i, t_j) f_j = g(s_i), \quad i = 1, \dots, m. \quad (4.3)$$

Esta discretização resulta no problema matricial de tamanho $m \times n$, $Ax = b$, onde: A é a matriz de tamanho $m \times n$ com cada elemento $a_{ij} = w_j k(s_i, t_j)$, x é o vetor em \mathbb{R}^n dado por $[f(t_1), \dots, f(t_n)]^T$ e b o vetor em \mathbb{R}^m definido por $[g(s_1), \dots, g(s_m)]^T$.

Note que os mesmos problemas de existência, unicidade e estabilidade também são encontrados em um problema inverso discreto. Isso porque pode ocorrer que $b \notin \mathcal{I}m(A)$, ou $\mathcal{N}(A) \neq 0$, e assim, como um problema inverso contínuo envolve questões de instabilidade, problemas inversos discretos também adotam esta característica. Isso porque ao discretizar uma integral que já é mal posta, o sistema linear produzido também é mal posto. Mais do que isso, quanto mais próximo o sistema discretizado fica do problema contínuo mais mal colocado o sistema algébrico se torna (GROETSCH (1993)).

Exemplo 4. *O exemplo a seguir, ilustra o mal condicionamento de um problema inverso discreto (GROETSCH (1993)):*

$$\int_0^1 (s^2 + t^2)^{\frac{1}{2}} f(t) dt = \frac{1}{3}((1 + s^2)^{\frac{3}{2}} - s^3).$$

A solução exata do problema é $f(t) = t$, e aplicando a regra do ponto médio com $h = \frac{1}{n}$, obtemos o seguinte problema discreto:

$$\frac{1}{n} \sum_{j=1}^n (t_i^2 + t_j^2)^{\frac{1}{2}} x_j = b_i, \quad i = 1, \dots, n,$$

sendo $t_j = \frac{(2j-1)}{(2n)}$; $b_i = \frac{1}{3}((1 + t_i^2)^{\frac{3}{2}} - t_i^3)$, e $x_j \approx f(t_j)$, $i = 1, \dots, n$. Obtemos a versão discretizada do problema, $Ax = b$, onde $a_{ij} = \frac{1}{n}(t_i^2 + t_j^2)^{\frac{1}{2}}$.

A expectativa é de que ao discretizar a solução, a cada refinamento na malha, a solução se aproxime ainda mais da solução exata. Porém, para problemas mal postos isso não é uma verdade, a medida que aumentamos o valor de n , isto é o número de nós na malha, há uma deterioração na aproximação. Observe que neste caso, o número de condição da matriz A , é dado na tabela a seguir para diferentes valores de n :

n	Cond(A)
5	5.4×10^4
10	1.9×10^{10}
20	1.1×10^{18}
35	2.3×10^{18}
45	1.0×10^{19}

Tabela 1 – Número de condição da matriz A .

Ao graficar a solução exata e a solução aproximada após a discretização, para $n = 10$ e $n = 20$, também é visível o quão sensível é o resultado obtido devido as características de A , veja as figuras 6 e 7.

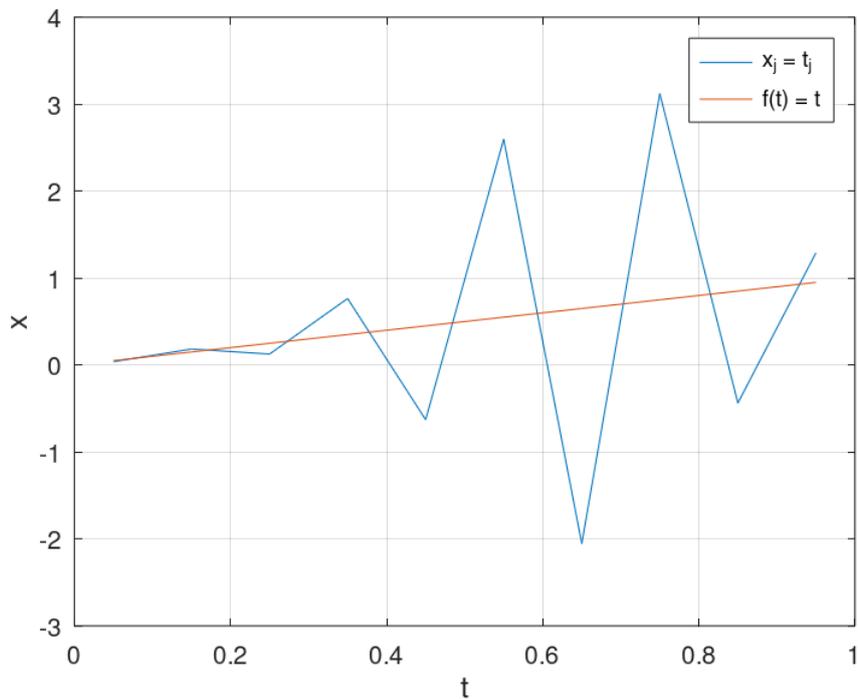


Figura 6 – Discretização, $n = 10$ (Fonte: Autora)

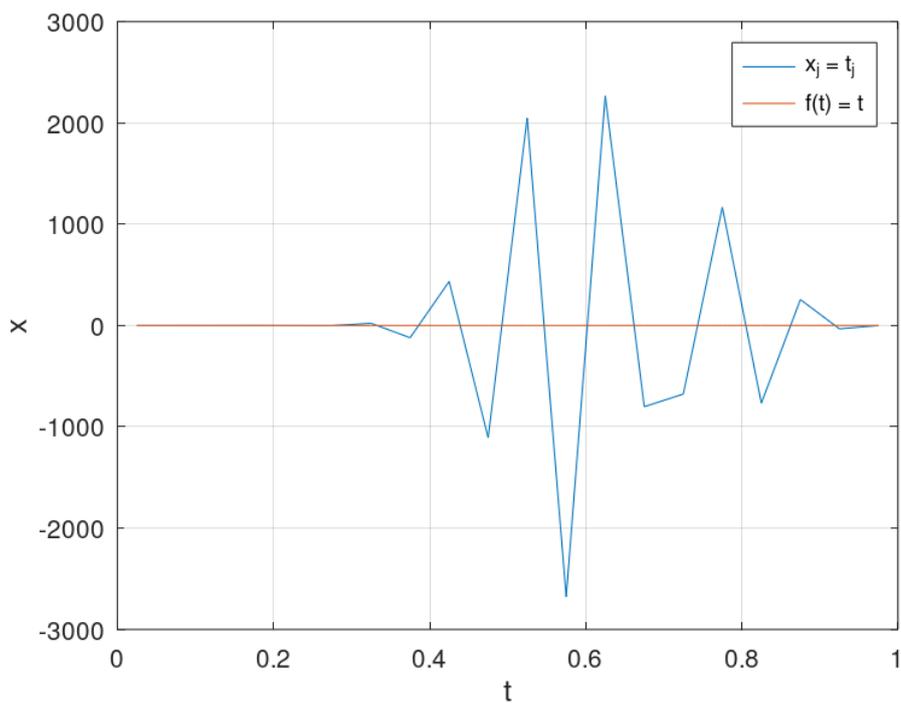


Figura 7 – Discretização, $n = 20$ (Fonte: Autora)

5 TÉCNICAS DE REGULARIZAÇÃO PARA PROBLEMAS INVERSOS LINEARES

Após apresentar os problemas inversos e classificá-los como problemas mal-postos, com base nas condições de Hadamard, agora os conceitos da SVD e a pseudo-inversa serão incorporados ao arsenal de ferramentas para a resolução desses problemas.

No capítulo anterior, o resultado obtido após a discretização de um problema inverso foi um problema matricial definido por $Ax = b$. No capítulo 2, vimos que, se A é uma matriz de tamanho $m \times n$ e tem posto n , é possível estimar o valor de x por meio de um problema de minimização.

Outro ponto já visto, é que a SVD é um importante recurso para a avaliação da solução de um problema de minimização, isto é, através da tendência dos valores singulares se aproximarem de zero, observa-se o aumento gradativo na instabilidade da solução.

Ao considerar o problema $Ax = b$, adicionando ainda o erro na medição do vetor b , ou seja, considerando $b = b_{\text{exato}} + e$, onde e é um vetor de perturbação, a conclusão é que $x = A^\dagger b$, (A^\dagger sendo a pseudo-inversa de A) não é uma boa aproximação para a solução do problema. Isso porque como visto na seção 2.1, a seguinte inequação é válida

$$\frac{\|x_{\text{exato}} - x\|}{\|x_{\text{exato}}\|} \leq \text{cond}(A) \frac{\|e\|}{\|b_{\text{exato}}\|}.$$

Se $\text{cond}(A)$ for muito grande isso implica que x estará muito distante de x_{exato} . Esse é o tipo de característica comumente encontrada em problemas inversos discretos. Sendo assim, pensando em encontrar soluções que sejam menos sensíveis a perturbações do que as soluções citadas anteriormente, é que os métodos de regularização são introduzidos. A essência dos métodos de regularização é suavizar as componentes dos valores singulares menores e dessa forma amenizar os efeitos dos ruídos das medições, aumentando a estabilidade das soluções.

Serão abordados neste capítulo dois métodos de regularização: SVD Truncada (TSVD) e a Regularização de Tikhonov.

5.1 SVD TRUNCADA – TSVD

Uma boa forma de amenizar esse erro para aproximar x é truncar a SVD, isto é determinar o melhor índice k , sem que haja perda de informações importantes no problema e evitando incluir erros na solução. Essa escolha para o índice k é conhecido como o **método da SVD Truncada**:

$$x_k = \sum_{i=1}^k \frac{u_i^T b}{\sigma_i} v_i.$$

Truncar a SVD é uma ideia razoável, uma vez que para os índices i pequenos temos:

$$\frac{u_i^T b}{\sigma_i} \approx \frac{u_i^T b^{\text{exato}}}{\sigma_i} = v_i^T x^{\text{exato}}.$$

O parâmetro de truncamento, k , deve ser escolhido de maneira a eliminar os ruídos da SVD. Porém, para garantir que a solução aproximada seja única é necessário inserir uma restrição para x , do tipo:

$$\begin{aligned} & \min \|x\|^2 \\ & \text{sujeito a } \|A_k x - b\|_2. \end{aligned}$$

Assim, a escolha de k é determinada avaliando o comportamento dos $u_i^T b$ que propagam maiores erros para a solução.

5.2 REGULARIZAÇÃO DE TIKHONOV

O Método de Regularização de Tikhonov é a técnica mais famosa para resolver problemas mal condicionados – classificação na qual os problemas inversos discretos estão inseridos, isso porque são problemas em que normalmente existe uma instabilidade na solução ou uma grande sensibilidade causada pelas perturbações.

Este método se destaca, pois enquanto uma das dificuldades da solução por mínimos quadrados é a contaminação por erros, o método de Tikhonov ao adicionar um parâmetro de regularização, balanceia o erro da aproximação da solução e penaliza as componentes que causam o crescimento acelerado e instabilidade na solução, é por esse motivo que o método de Regularização de Tikhonov pertence à classe dos métodos de penalização. Ou seja, através de λ controla-se o nível de perturbação e regularização presente na solução.

Neste método (3.4) é substituído por

$$x_\lambda = \arg \min_{x \in \mathbb{R}^n} \{ \|b - Ax\|_2^2 + \lambda^2 \|L(x - x_0)\|_2^2 \} \quad (5.1)$$

o ponto chave deste método é determinar $\lambda > 0$, no próximo capítulo serão vistos dois métodos: a Curva-L e o método de discrepância de Morozov, que definem boas aproximações para λ .

Voltando a expressão anterior, $\lambda > 0$ é o parâmetro a ser determinado; $\|L(x - x_0)\|$ é o termo de regularização, onde L pode ser tanto um operador diferencial ou até mesmo a matriz identidade, nas aplicações abordadas deste trabalho $L = I$ e x_0 é uma aproximação inicial para a solução, na maioria das vezes considera-se o vetor nulo $x_0 = 0$. E por fim, $\|b - Ax\|$ é a norma do resíduo, portanto mede o erro no ajuste dos dados.

Assim, no caso do λ determinado ser muito pequeno, a penalização da norma do termo regularizador é pequena, o que faz com que a norma da solução aproximada x_λ seja muito grande, indicando um distanciamento da solução exata. E, avaliando o caso de λ determinado ser muito grande, nota-se a norma muito pequena para x_λ e maior estabilidade é incorporada na aproximação, porém com isso $\|b - Ax\|$ se sobressai, o que por consequência afasta a solução x_λ da solução exata.

As equações normais de (5.1) são:

$$(A^T A + \lambda^2 L^T L)x_\lambda = A^T b + \lambda^2 L^T L x_0 \quad (5.2)$$

Além disso, se $x_0 = 0$ e substituindo L por I o problema se resume a

$$(A^T A + \lambda^2 I)x_\lambda = A^T b \quad (5.3)$$

Utilizando a SVD da matriz A e a truncando em seu r -ésimo valor singular não nulo na diagonal de Σ , segue que:

$$((V\Sigma^T U^T)(U\Sigma V^T) + \lambda^2 I)x_\lambda = A^T b$$

$$((V\Sigma^T \Sigma V^T) + \lambda^2 I)x_\lambda = (V\Sigma^T U^T)b$$

como $VV^T = I$, pois é ortogonal

$$(V\Sigma^T \Sigma V^T + V\lambda^2 I V^T)x_\lambda = (V\Sigma^T U^T)b$$

$$V(\Sigma^T \Sigma + \lambda^2 I)V^T x_\lambda = (V\Sigma^T U^T)b$$

$$(\Sigma^T \Sigma + \lambda^2 I)V^T x_\lambda = (\Sigma^T U^T)b$$

$$x_\lambda = [(\Sigma^T \Sigma + \lambda^2 I)V^T]^{-1} \Sigma^T U^T b$$

$$x_\lambda = (\Sigma^T \Sigma + \lambda^2 I)^{-1} V \Sigma^T U^T b$$

$$\begin{aligned}
 x_\lambda &= \sum_{i=1}^r \left(\frac{1}{\sigma_i^2 + \lambda^2} \right) \sigma_i u_i^T b v_i \\
 x_\lambda &= \sum_{i=1}^r \left(\frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} \right) \frac{u_i^T b}{\sigma_i} v_i \\
 \implies x_\lambda &= \sum_{i=1}^r f_i \frac{u_i^T b}{\sigma_i} v_i
 \end{aligned}$$

onde, os fatores de filtragem f_i com $i = 1, \dots, n$, são:

$$f_i = \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} \approx \begin{cases} 1 & \text{se } \sigma_i \gg \lambda \\ \frac{\sigma_i^2}{\lambda^2} & \text{se } \sigma_i \ll \lambda \end{cases}$$

Note que, quando os valores singulares são maiores que o parâmetro λ , eles influenciam diretamente na solução. Porém, quando os valores singulares são muito menores que λ , eles são filtrados, e sua influência é atenuada na solução.

Desta maneira, se F é a matriz de filtragem, ou seja, onde $F_{ij} = 0$ para $i \neq j$ e $F_{ij} = f_i$ se $i = j$, podemos expressar a solução da seguinte forma matricial:

$$x_\lambda = V F_\lambda \Sigma^{-1} U^T b. \tag{5.4}$$

No caso de utilizar a TSVD, os fatores seguem a condição:

$$f_i = \begin{cases} 1 & \text{se } 1 \leq i \leq k \\ 0 & \text{se } k + 1 \leq i \leq r \end{cases}$$

E a solução pode ser expressa por:

$$x_k = V F_k \Sigma^{-1} U^T b. \tag{5.5}$$

6 ESCOLHA DO PARAMÊTRO DE REGULARI- ZAÇÃO

Baseados na ideia de filtragem da SVD, foram apresentados dois métodos de regularização para problemas inversos, e a chave para a estabilidade alcançada por esses métodos está na escolha adequada do parâmetro de regularização. No método da TSVD, é necessário determinar o valor de k e no método de Tikhonov, o parâmetro a ser definido é λ .

No entanto, antes de abordar os métodos para a escolha do parâmetro de regularização, Hansen motiva e introduz os métodos da Curva-L e do Princípio de Discrepância ao analisar os erros associados às soluções regularizadas. A seguir, adotaremos a mesma estratégia.

Através do método de Tikhonov, a solução regularizada x_λ é dada por:

$$x_\lambda = VF_\lambda \Sigma^{-1} U^T b.$$

Visto que o foco é no sistema $b = Ax_{\text{exato}} + e$, ou seja, o sinal exato adicionado de uma perturbação e , segue que o erro na Regularização de Tikhonov é dado por:

$$\begin{aligned} x_{\text{exato}} - x_\lambda &= x_{\text{exato}} - VF_\lambda \Sigma^{-1} U^T b \\ &= x_{\text{exato}} - VF_\lambda \Sigma^{-1} U^T (U \Sigma V^T x_{\text{exato}}) - VF_\lambda \Sigma^{-1} U^T e \\ &= (I - VF_\lambda \Sigma^{-1} U^T U \Sigma V^T) x_{\text{exato}} - VF_\lambda \Sigma^{-1} U^T e \\ &= V(I - F_\lambda) V^T x_{\text{exato}} - VF_\lambda \Sigma^{-1} U^T e. \end{aligned}$$

O primeiro termo da expressão é chamado de erro de regularização, o qual resulta ao acrescentar os fatores de filtragem. A fórmula do erro de regularização é dada a seguir, e nela é possível avaliar o desvio que x_λ sofre se comparado com x_{exato} :

$$\Delta x_{\text{reg}} = V(I - F_\lambda) V^T x_{\text{exato}} = \sum_{i=1}^n (1 - f_i) (v_i^T x_{\text{exato}}) v_i. \quad (6.1)$$

O segundo termo é chamado erro de perturbação, resultante da inversão e filtragem do erro presente nos dados iniciais do problema, conforme ilustrado na seguinte fórmula:

$$\Delta x_{\text{pert}} = VF_\lambda \Sigma^{-1} U^T e. \quad (6.2)$$

Para a TSVD, como a diagonal principal é do tipo $\text{diag}(F_s) = (1_1, \dots, 1_k, 0_{k+1}, \dots, 0_n)$, as expressões de erro definidas anteriormente podem ser simplificadas:

$$\Delta x_{\text{reg}} = \sum_{i=k+1}^n (1 - f_i)(v_i^T x_{\text{exato}})v_i, \quad \Delta x_{\text{pert}} = \sum_{i=1}^k \frac{u_i^T e}{\sigma_i} v_i.$$

Ao tentar reduzir o erro de perturbação, é introduzido o erro de regularização, e vice-versa. Assim, os erros estarão sempre presentes nas aproximações; o objetivo é encontrar um equilíbrio ideal entre os dois. Assumindo, por simplicidade, a regularização TSVD, supondo que a condição de Picard é satisfeita; os coeficientes $\frac{|u_i^T b|}{\sigma_i}$, irão decair até um nível determinado pelo desvio padrão do ruído η . Denotamos por k_η o índice que marca a transição entre a região de decaimento e a de transação.

A norma da solução obtida pela regularização TSVD, é dada por:

$$\|x_k\|_2^2 = \sum_{i=1}^k \left(\frac{u_i^T b}{\sigma_i} \right)^2 \leq \|x_{k+1}\|_2^2.$$

$$\text{Se } k < k_\eta : \quad \|x_k\|_2^2 \approx \sum_{i=1}^k \left(\frac{u_i^T b_{\text{exato}}}{\sigma_i} \right)^2,$$

$$\text{Se } k > k_\eta : \quad \|x_k\|_2^2 \approx \sum_{i=1}^{k_\eta} \left(\frac{u_i^T b_{\text{exato}}}{\sigma_i} \right)^2 + \sum_{i=k_\eta+1}^k \left(\frac{\eta}{\sigma_i} \right)^2.$$

Note, se $k < k_\eta$, a norma da solução descrece suavemente, se $k > k_\eta$ a norma da solução cresce com muito mais velocidade. Desta forma, $k = k_\eta$ é o ponto de inflexão, o qual diferencia os dois tipos de comportamentos.

Em resumo, o que pode-se concluir é que a escolha do parâmetro de truncamento é $k = k_\eta$. A partir de agora, serão apresentados métodos para determinar esse ponto de transição.

6.1 PRINCÍPIO DE DISCREPÂNCIA DE MOROZOV

Se, em vez de avaliar o lado direito da norma da solução mencionado anteriormente, considerarmos a escolha de k onde a norma do vetor de perturbação é $\|e\|_2 = n^{\frac{1}{2}}\eta$, e além disso, introduzirmos um fator de segurança ν_{PD} , usando como limite da norma do resíduo $\nu_{\text{PD}}\|e\|_2$, obtemos um método para determinar o parâmetro k , conhecido como Princípio da Discrepância, atribuído a Morozov.

Dessa forma, a estratégia é escolher $k = k_{\text{DP}}$ tal que $\|Ax_k - b\|_2 \geq \nu_{\text{PD}}\|e\|_2 > \|Ax_{k+1} - b\|_2$.

Aplicando essa estratégia partindo do método de Tikhonov a escolha seria por: $\lambda = \lambda_{PD}$ tal que $\|Ax_\lambda - b\|_2 = \nu_{PD}\|e\|_2$.

A vantagem dessa estratégia é a solução única, uma vez que a norma do resíduo é monótona. No entanto, é importante destacar a necessidade de possuir uma estimativa do ruído nos dados, isto é $\|e\|_2$. Se essa estimativa for muito grande ou muito pequena, a solução poderá incluir os ruídos de erro de regularização ou perturbação, como já discutido, isto claramente é uma desvantagem do método pois em aplicações reais é algo que não é possível controlar.

6.2 CURVA-L

O nome do método se deve ao fato da curva gerada se assemelhar com a letra L. O critério do método da Curva-L consiste em selecionar o λ que equilibre o ajuste dos dados e a estabilidade da solução. Para isso são realizadas simulações numéricas com diferentes valores de λ positivos e maiores do que zero, pois ao utilizar $\lambda = 0$, o termo de regularização é perdido.

Dessa forma, através das várias soluções geradas em função de λ e utilizando a SVD podemos escrever a norma de cada solução, x_λ , e a norma de cada resíduo da solução, $Ax_\lambda - b$, obtidos através da regularização de Tikhonov:

$$\|x_\lambda\|^2 = \sum_{i=1}^r \left(f_i \frac{u_i^T b}{\sigma_i} v_i \right)^2, \quad (6.3)$$

$$\|Ax_\lambda - b\|^2 = \sum_{i=1}^r ((1 - f_i) u_i^T b)^2 + b_\perp^2, \quad (6.4)$$

em que $b_\perp = b - \sum_{i=1}^r u_i^T b u_i = \sum_{i=1}^m u_i^T b u_i$, que é a componente de b que não pertence ao espaço das colunas de A . Ainda, veja que u_i é uma base ortonormal para o espaço \mathbb{R}^m .

Com base nisso, ao graficar o logaritmo da norma do resíduo $Ax_\lambda - b = r_\lambda$ contra o logaritmo da norma da solução x_λ , para que seja mais evidente o formato da curva gerada, o método de determinação de parâmetro através da Curva-L indica o parâmetro de regularização através do valor que maximiza a curvatura da curva parametrizada gerada em função de $\lambda > 0$. Isto é, avaliamos a curvatura \hat{c} da curva gerada por $(\log \|Ax_\lambda - b\|_2, \log \|x_\lambda\|_2)$, veja que isso irá ocorrer no ‘joelho’ ou canto da curva, como pode ser visto na figura 8.

Assim a Curva-L é caracterizada por dois trechos bastante distintos. O comportamento desses intervalos da curva estão ligados ao erro de perturbação e ao erro de regularização na solução obtida (veja na figura 8).

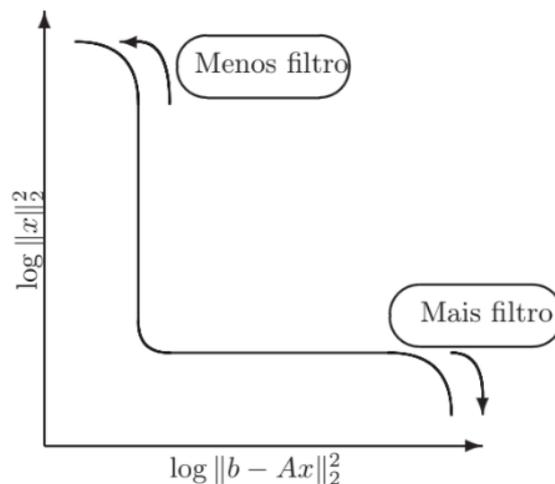


Figura 8 – Curva-L (Fonte: BAZÁN and BORGES (2009))

Portanto, pode-se aprender muito sobre a dependência da solução em relação ao erro de regularização ao estudar a Curva-L. Recorde-se que no início do capítulo foi definido k_η , agora para avaliar a Curva-L utilizaremos a notação i_η a qual indica a indexação de transição entre a parte vertical e a parte horizontal da Curva-L gerada. Analisando intuitivamente, suponha um vetor com i valores de λ_i , quando i é menor do que i_η , ou seja, é selecionado um λ muito menor do que o adequado então o erro de regularização é mais dominante (Δ_{reg}), porque serão adicionadas menos componentes na aproximação de x_λ , fazendo com que o resíduo seja grande porém como λ é menor a norma da solução é pequena.

Quando i é muito maior do que i_η , o λ indicado irá amenizar o efeito dos valores singulares. Porém, irá também aumentar o erro de perturbação da solução (Δ_{pert}). Dessa forma, a norma da solução $\|x_\lambda\|_2$ aumenta quando i aumenta, isso porque as maiores componentes com ruídos estão presentes na aproximação da solução; mas, por outro lado a norma do resíduo é praticamente constante em $\|e\|_2$.

Portanto, a Curva-L apresenta um trecho em que a influência do erro de regularização é mais evidente, caracterizado por uma curva quase horizontal que decai suavemente. Em contraste, há um trecho onde os erros de perturbação têm maior impacto, resultando em uma curva praticamente vertical.

Ao analisar a região de transição dessa curva, encontramos o ponto ótimo de equilíbrio entre os dois erros, conhecido como o canto da Curva-L.

Para determinar o ponto de curvatura máxima, suponha $\xi = \|x_\lambda\|^2$ e $\rho = \|Ax_\lambda - b\|^2$, dessa maneira é possível derivar ξ e ρ em relação a λ . Repare que ao fazer isso ambas as derivadas possuem a componente de filtro $f_i(\lambda)$ e pela regra da cadeia o produto $f_i f'_i$, aparece

na equação. Portanto, lembrando que $f_i = \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2}$ e derivando em relação a λ o resultado obtido é:

$$\begin{aligned} f'_i(\lambda) &= (-1) \sigma_i^2 (\sigma_i^2 + \lambda^2)^{-2} 2\lambda \\ f'_i &= -2\lambda \frac{\sigma_i^2}{(\sigma_i^2 + \lambda^2)^2}. \end{aligned} \quad (6.5)$$

Ainda, sendo $1 - f_i = \frac{\lambda^2}{\sigma_i^2 + \lambda^2}$, obtemos:

$$(1 - f_i)f_i^2 = \lambda^2 \frac{(\sigma_i^2)^2}{(\sigma_i^2 + \lambda^2)^3}. \quad (6.6)$$

Por fim multiplicando (6.5) por f_i e utilizando o resultado de (6.6), obtemos a seguinte expressão para $f_i f'_i$:

$$\begin{aligned} f_i f'_i(\lambda) &= f_i f'_i = -2\lambda \frac{(\sigma_i^2)^2}{(\sigma_i^2 + \lambda^2)^3} \\ &= \frac{-2}{\lambda} \left(\lambda^2 \frac{(\sigma_i^2)^2}{(\sigma_i^2 + \lambda^2)^3} \right) \\ &= \frac{-2}{\lambda} (1 - f_i) f_i^2. \end{aligned} \quad (6.7)$$

A partir desses resultados preliminares é possível obter ξ' e ρ' , veja a seguir:

$$\begin{aligned} \xi' &= \frac{d\xi}{d\lambda} = \sum_{i=1}^n 2 \left(f_i \frac{u_i^T b}{\sigma_i} \right) \frac{u_i^T b}{\sigma_i} f'_i(\lambda) \\ &= \sum_{i=1}^n 2 \left(\frac{u_i^T b}{\sigma_i} \right)^2 f_i f'_i, \end{aligned}$$

por (6.7), segue que

$$\begin{aligned} \xi' &= \sum_{i=1}^n 2 \left(\frac{u_i^T b}{\sigma_i} \right)^2 \frac{-2}{\lambda} f_i^2 (1 - f_i) \\ \implies \xi' &= \frac{-4}{\lambda} \sum_{i=1}^n 2 \left(\frac{u_i^T b}{\sigma_i} \right)^2 f_i^2 (1 - f_i). \end{aligned}$$

Analogamente,

$$\begin{aligned}
 \rho' &= \frac{d\rho}{d\lambda} = \frac{4}{\lambda} \sum_{i=1}^n (u_i^T b) f_i (1 - f_i)^2 \\
 &= \frac{4}{\lambda} \sum_{i=1}^n \frac{(u_i^T b)^2}{\sigma_i^2} f_i^2 (1 - f_i)^2 \left(\frac{\sigma_i^2}{f_i} \right) \\
 &= \frac{4}{\lambda} \sum_{i=1}^n \left(\frac{u_i^T b}{\sigma_i} \right)^2 f_i^2 (1 - f_i) \left(\frac{(1 - f_i)\sigma_i^2}{f_i} \right) \\
 &= \frac{4}{\lambda} \sum_{i=1}^n \left(\frac{u_i^T b}{\sigma_i} \right)^2 f_i^2 (1 - f_i) \left(\frac{\lambda^2}{\sigma_i^2 + \lambda^2} \frac{\sigma_i^2 + \lambda^2}{\sigma_i^2} \sigma_i^2 \right) \\
 \implies \rho' &= \frac{4}{\lambda} \sum_{i=1}^n \left(\frac{u_i^T b}{\sigma_i} \right)^2 f_i^2 (1 - f_i) \lambda^2 = -\lambda \xi'.
 \end{aligned}$$

Portanto, como $\lambda > 0$ então $\xi'(\lambda) < 0$ e por consequência $\rho' > 0$. Pela regra da cadeia:

$$\frac{d\xi}{d\rho} = \frac{d\xi}{d\lambda} \frac{d\lambda}{d\rho} \implies \xi'(\lambda) \frac{1}{\frac{d\rho}{d\lambda}} = \frac{\xi'(\lambda)}{\rho(\lambda)} < 0,$$

isso mostra que $\|x_\lambda\|^2$ é uma função decrescente em relação a $\|Ax_\lambda - b\|^2$.

Porém, para a Curva-L são utilizados os pontos $(\log \|Ax_\lambda - b\|, \log \|x_\lambda\|)$, para isso é necessário definir $\hat{\xi} = \log \|x_\lambda\|^2$ e $\hat{\rho} = \log \|Ax_\lambda - b\|^2$. Veja então que a Curva-L é parametrizada por $r(\lambda) = (\frac{1}{2}\hat{\xi}, \frac{1}{2}\hat{\rho})$. Derivando as duas quantidades em relação a λ , obtemos

$$\hat{\xi}' = \frac{\xi'}{\xi} \quad \text{e} \quad \hat{\rho}' = \frac{\rho'}{\rho}.$$

E derivando novamente, segue que a segunda derivada das componentes $\hat{\xi}$ e $\hat{\rho}$ são dadas por

$$\hat{\xi}'' = \frac{\xi''\xi - (\xi')^2}{\xi^2} \quad \text{e} \quad \hat{\rho}'' = \frac{\rho''\rho - (\rho')^2}{\rho^2}.$$

Seguindo, pela definição de curvatura,

$$\mathcal{K} = \frac{\|r'(\lambda) \times r''(\lambda)\|}{\|r'(\lambda)\|^3}$$

Para a Curva-L, obtemos então:

$$\hat{c}_\lambda = 2 \frac{\hat{\rho}'\hat{\xi}'' - \hat{\rho}''\hat{\xi}'}{((\hat{\rho}')^2 + (\hat{\xi}')^2)^{\frac{3}{2}}}. \tag{6.8}$$

Por fim, a curvatura pode ser expressa por:

$$\hat{c}_\lambda = 2 \frac{\xi\rho}{\xi'} \frac{\lambda^2\xi'\rho + 2\lambda\xi\rho + \lambda^4\xi\xi'}{(\lambda^2\xi^2 + \rho^2)^{\frac{3}{2}}}. \tag{6.9}$$

Mostraremos essa última relação. Do numerador da expressão (6.8):

$$\begin{aligned}\hat{\rho}'\hat{\xi}'' - \hat{\rho}''\hat{\xi}' &= \frac{\rho'}{\rho} \left(\frac{\xi''\xi - (\rho')^2}{\xi^2} \right) - \frac{\xi'}{\xi} \left(\frac{\rho''\rho - (\rho')^2}{\rho^2} \right) \\ &= \frac{1}{\rho^2\xi^2} (\rho\rho'\xi\xi' - \rho\rho'(\xi')^2 - \xi\xi'\rho\rho'' + \xi\xi'(\rho')^2).\end{aligned}$$

Após algumas simplificações algébricas e substituições, tendo em vista que:

$$\rho'(\lambda) = -\lambda^2\xi' \implies \rho''(\lambda) = -2\lambda\xi' - \lambda^2\xi'', \text{ obtemos:}$$

$$\begin{aligned}\hat{\rho}'\hat{\xi}'' - \hat{\rho}''\hat{\xi}' &= \frac{1}{\rho^2\xi^2} (2\lambda\xi\rho(\xi')^2 + \xi\xi'(\rho')^2 + \lambda^2\rho(\xi')^2) \\ &= \frac{\xi'}{\rho^2\xi^2} (2\lambda\rho\xi\xi' + \lambda^2\rho(\xi')^2 + \lambda^4\xi(\xi')^2) \\ &= \frac{(\xi')^2}{\rho^2\xi^2} (\lambda^2\xi'\rho + 2\lambda\xi\rho + \lambda^4\xi\xi').\end{aligned}$$

Do denominador da expressão (6.8):

$$\begin{aligned}\left((\hat{\rho}')^2 + (\hat{\xi}')^2 \right)^{\frac{3}{2}} &= \left(\left(\frac{\rho'}{\rho} \right)^2 + \left(\frac{\xi'}{\xi} \right)^2 \right)^{\frac{3}{2}} \\ &= \left(\frac{(\rho')^2\xi^2 + (\xi')^2\rho^2}{\xi^2\rho^2} \right)^{\frac{3}{2}} \\ &= \frac{1}{(\xi\rho)^3} \left((-\lambda^2\xi')^2\xi^2 + (\xi')^2\rho^2 \right)^{\frac{3}{2}} \\ &= \left(\frac{\xi'}{\rho\xi} \right)^3 (\lambda^2\xi^2 + \rho^2)^{\frac{3}{2}}.\end{aligned}$$

Por fim,

$$\begin{aligned}\hat{c}_\lambda &= 2 \left(\frac{\xi\rho}{\xi'} \right)^3 \left(\frac{\xi'}{\rho\xi} \right)^2 \frac{(2\lambda\xi\rho + \lambda^2\xi'\rho + \lambda^4\xi\xi')}{(\lambda^2\xi^2 + \rho^2)^{\frac{3}{2}}} \\ \implies \hat{c}_\lambda &= 2 \frac{\xi\rho}{\xi'} \frac{\lambda^2\xi'\rho + 2\lambda\xi\rho + \lambda^4\xi\xi'}{(\lambda^2\xi^2 + \rho^2)^{\frac{3}{2}}}.\end{aligned}$$

O critério da Curva-L como método de determinação de parâmetro para a regularização de Tikhonov apresenta algumas limitações. Uma delas é que para soluções muito suaves em que as componentes $|v_i^T x^{\text{exato}}|$ decaem de maneira rápida, fazendo com que a solução seja influenciada pelas primeiras componentes da SVD, a Curva-L irá indicar um parâmetro muito menor do que o ideal buscado. Outro caso em que o método pode indicar um parâmetro inadequado é quando a mudança na norma do resíduo e da solução são muito pequenas para índices k consecutivos. Mais uma vez são adicionadas componentes da SVD com erros antes que

algum canto da Curva-L seja indicado. Porém, esse é um cenário mais facilmente encontrado em problemas de grande escala.

Computacionalmente existem alguns desafios em relação ao uso da Curva-L, note que \hat{c}_λ não é definido para todo λ . Ao realizar a simulação numérica para valores de λ temos pontos discretos, ou seja, é necessário utilizar algum método de interpolação na curva obtida para que assim possa ser avaliado seu ponto de mínimo. De maneira geral, embora a Curva-L seja um método heurístico e não garanta a melhor regularização em todos os casos, ela é uma excelente ferramenta para determinar o parâmetro λ , ainda que, por si só, avaliá-la seja um novo problema.

7 APLICAÇÕES

Neste capítulo os métodos apresentados serão exemplificados para que seja possível comparar os resultados obtidos e avaliar os principais aspectos dos problemas inversos discretos lineares.

7.1 EQUAÇÃO INTEGRAL DE FREDHOLM DE PRIMEIRO TIPO

Recuperando o exemplo introduzido no capítulo 3, observamos o comportamento da integral de primeira espécie: - essa integral se classifica como um exemplo de problema mal condicionado. Vimos ainda, que ao discretizar o problema através da regra do ponto médio, gerando a matriz A do problema $Ax = b$ e aumentando a malha para $n = 5, 10, 20$, etc, a matriz A resultava em problemas cada vez pior condicionados e as soluções se distanciavam ainda mais da solução exata do problema (ver figuras 6 e 7). Isso porque, sendo $cond(A) = \frac{\sigma_{max}}{\sigma_{min}}$, se o número de condição está próximo de 1, o problema é bem condicionado e a solução de $Ax = b$ é estável em relação à variação de ruído nos dados. Porém, se $cond(A)$ é muito grande, como ocorre no exemplo aqui considerado, o problema é instável e pequenas perturbações nos dados do problema geram grandes variações na solução final.

De fato, os valores singulares possuem grande influência na estabilidade da função. E mais do que isso, como eles sempre decaem para zero, a velocidade de convergência dos autovalores define o quão suave é a função que estamos estudando. Para o exemplo de [GROETSCH \(1993\)](#), veja que a medida que os valores singulares diminuem, menos suaves são as curvas geradas pelos vetores singulares associados aos autovalores σ_i , com $i = \{1, 2, 5, 10, 20 \text{ e } 50\}$, conforme observado abaixo na figura 9, onde no eixo vertical temos o valor do elemento na i -ésima posição do vetor e no eixo horizontal temos a posição do elemento. Em outras palavras, o comportamento das curvas demonstra o motivo das baixas frequências serem mais suavizadas comparativamente às frequências mais altas em problemas diretos.

Ainda sobre a suavidade, a condição discreta de Picard estabelece condições favoráveis para o cálculo de soluções aproximadas estáveis de problemas discretos mal postos. Ou seja, se os coeficientes $u_i^T b$ decaem para zero mais rápido do que os valores singulares, então observamos que a aproximação da solução é mais estável, ou seja, menos sensível aos ruídos.

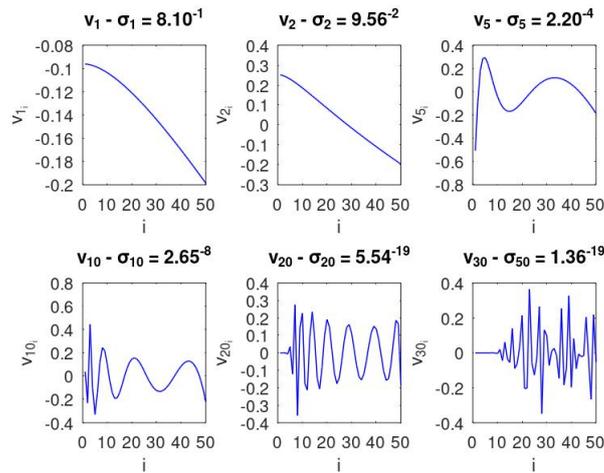


Figura 9 – Comportamento dos vetores singulares v_i associados a σ_i de A .

A questão com as integrais de Fredholm de primeira espécie é que os dados com ruído geralmente não irão satisfazer a condição de Picard. Para observar esse comportamento, vamos analisar o comportamento dos coeficientes $|u_i^T b|$ e $\frac{|u_i^T b|}{\sigma_i}$ para problema discretizado para $n = 20$ e considerando o vetor b_{exato} e com adição de ruído e , gerado pela função `rand` do Octave, normalizado e multiplicado por um fator de 10^{-3} .

$$b = b_{\text{exato}} + 10^{-3} \|b_{\text{exato}}\| \frac{e}{\|e\|}. \quad (7.1)$$

Na figura 10, a condição discreta de Picard é satisfeita até σ_{14} onde $|u_i^T b| < \sigma_i$, após $i = 14$ os coeficientes $|u_i^T b|$ começam a superar os valores singulares, isso considerando b exato. Para a figura 11, com b perturbado, a condição não é satisfeita a partir de $i = 5$ fazendo com que os coeficientes $\frac{|u_i^T b|}{\sigma_i}$ aumentem com maior velocidade.

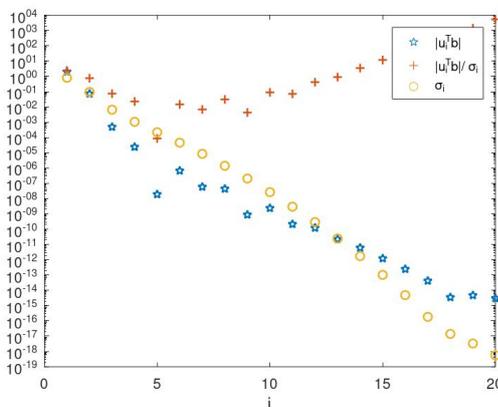


Figura 10 – Condição Discreta de Picard sem adição de ruído. (Fonte: Autora).

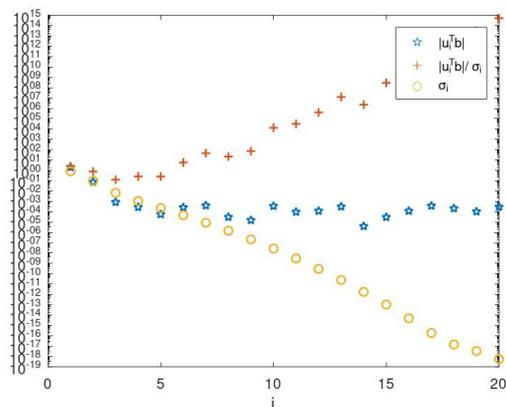


Figura 11 – Condição Discreta de Picard com adição de ruído. (Fonte: Autora).

Avaliar a condição discreta de Picard, proporciona uma ideia intuitiva de qual parâmetro de truncamento escolher para a SVD e obter uma solução com menos perturbações geradas pelas baixas frequências dos σ_i . Veja por exemplo, partindo do método de regularização TSVD, ao utilizar uma malha de discretização de $n = 20$, se forem consideradas todas as componentes da SVD a solução sofre grande influência dos valores singulares σ_i de baixa frequência, prejudicando a solução encontrada. Porém, como visto na figura 11, se realizarmos um truncamento da SVD no índice 5 – ou seja, para $i > 5$ o filtro irá zerar os valores singulares de S (método de regularização TSVD) – então o resultado é uma aproximação muito melhor (veja nas figuras 12, 13 e 14).

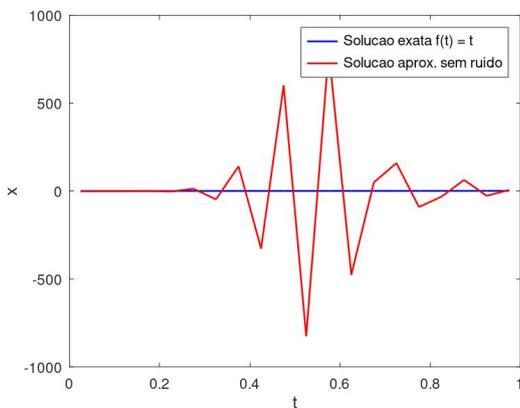


Figura 12 – Solução do problema com $n = 20$ e sem truncamento (b sem perturbação) (Fonte: Autora).

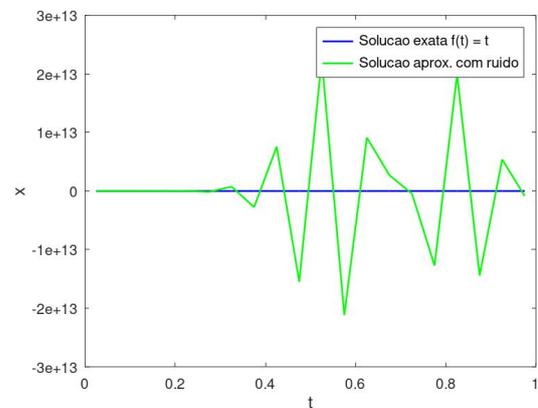


Figura 13 – Solução do problema com $n = 20$ e sem truncamento (b com perturbação) (Fonte: Autora).

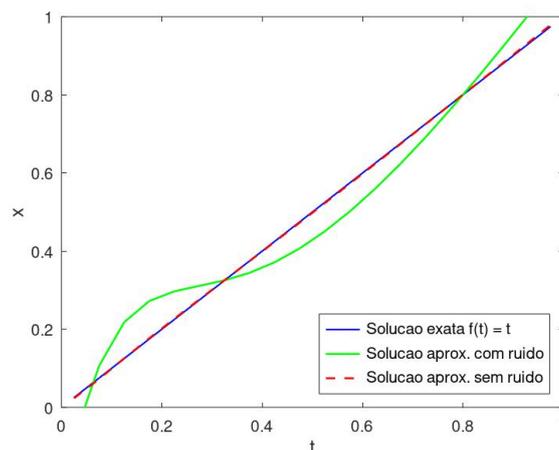


Figura 14 – Solução do problema através do método TSVD com $n = 20$ e $k = 5$

Como já mencionado a chave da determinação do parâmetro de regularização é o balanceamento entre o erro de perturbação e o erro de regularização da aproximação. Observar a condição de Discreta de Picard oferece uma opção intuitiva boa para a escolha do parâmetro de truncamento do método da TSVD, porém é possível que esteja sendo adicionado um dos tipos de erros. Para avaliar essa situação graficar o erro de perturbação ($\|\Delta x_{\text{pert}}\|$) e o erro de regularização ($\|\Delta x_{\text{reg}}\|$) juntamente do erro da solução ($\|t - x_k\|$) possibilitará entender essa dinâmica.

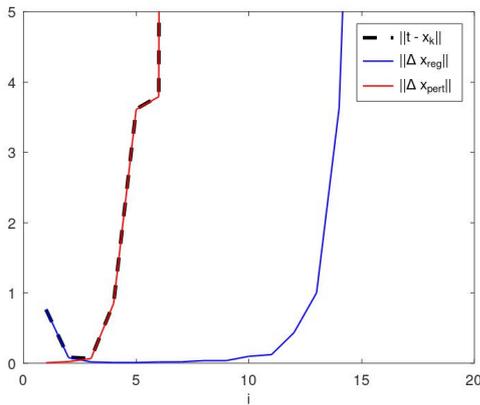


Figura 15 – Norma do erro de regularização e perturbação como função do parâmetro de truncamento k pelo método da TSVD (Fonte: Autora).

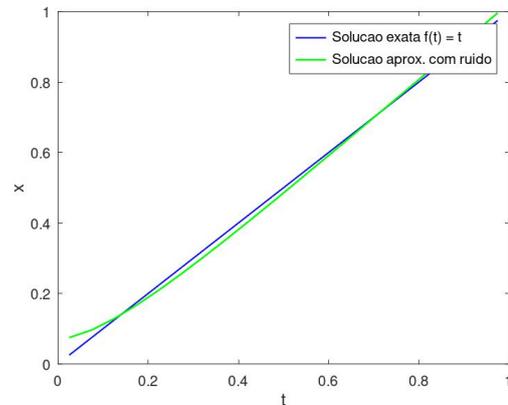


Figura 16 – Solução do problema através do método da TSVD com $n = 20$ e $k = 3$ (Fonte: Autora).

Tanto o erro de perturbação quanto o erro de regularização estão presentes na solução regularizada, isso é visível pelo gráfico da figura 15. Veja que ambos os erros se equilibram e o erro da solução ($\|t - x_k\|$) é mínimo quando $i = 3$, portanto ao escolher $k = 3$ a solução regularizada pelo método da TSVD é mais assertivo. A solução para o problema com a escolha do parâmetro de truncamento $k = 3$ pode ser visto na figura 16. Outra observação a ser feita é que para $k = 5$ escolhido anteriormente, o erro de regularização era minimizado enquanto o erro de perturbação se tornava dominante na solução.

Abaixo, é possível observar o resíduo da solução antes e após a regularização TSVD com $k = 3$:

n	$\ Ax_k - b_{\text{exato}}\ $	$\ t - x_k\ $	$\ Ax_{\text{com ruído}} - b_{\text{exato}}\ $	$\ t - x_{\text{com ruído}}\ $
10	1.1831×10^{-3}	2.9169	1.4140×10^{-3}	8.4284×10^4
20	1.9694×10^{-3}	5.7984	4.0025	4.7306×10^{14}

Tabela 2 – Resíduo e erro das soluções com e sem adição de ruído – TSVD com $k = 3$.

Veja também os resultados obtidos pelo método da TSVD com $k = 5$ e $k = 10$, esses dados proporcionam visualizar a deterioração da solução ao aumentar a malha de discretização e ao aumentar o parâmetro de truncamento:

	$k = 5$		$k = 10$	
n	$\ Ax_k - b_{\text{exato}}\ $	$\ t - x_k\ $	$\ Ax_k - b_{\text{exato}}\ $	$\ t - x_k\ $
10	1.3578×10^{-3}	3.1178	1.3566×10^{-3}	2.7794×10^5
20	1.8458×10^{-3}	5.8554	1.8244×10^{-3}	6879.3

Tabela 3 – Resíduo e erro das soluções – TSVD com $k = 5$ e $k = 10$.

Portanto, aumentar a malha de discretização não oferece melhora na solução, isso porque a integral é mal posta e a matriz A possui número de condição muito grande o que piora o resultado do problema inverso através da solução ingênua (como visto no capítulo 3). Veja que pela teoria clássica de perturbação temos:

$$\frac{\|x_{\text{exato}} - x\|}{\|x_{\text{exato}}\|} \leq \text{cond}(A) \frac{\|e\|}{\|b_{\text{exato}}\|}.$$

Portanto, quando $\text{cond}(A)$ é muito grande, a solução aproximada pode ficar muito longe da solução exata.

Foi possível obter um bom resultado através do método da TSVD, a seguir será avaliado se é possível melhorar ainda mais o desempenho da aproximação da solução. Com o objetivo de amenizar o impacto dos valores singulares na aproximação do problema, e não simplesmente anulá-los a partir de determinado índice, vamos avaliar agora o desempenho do método de Tikhonov, juntamente com os métodos de determinação de parâmetros da Curva-L e discrepância de Morozov.

O método de regularização de Tikhonov, de maneira resumida, consiste em adicionar um filtro do tipo $f_i = \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2}$ e portanto a solução fornecida para o método é $x_\lambda = \sum_{i=1}^r f_i \frac{u_i^T b}{\sigma_i} v_i$, com r sendo o posto de A . Essa estratégia espera obter melhores aproximações para a solução para algum λ adequado.

Para determinar o λ adequado, foram realizadas simulações com valores distintos para λ . Para tal, foi construído um vetor $\text{Lambda} \in \mathbb{R}^{100}$, onde os elementos desse vetor foram gerados de maneira equidistante dentro do intervalo $(10^{-10}, 1)$. Dessa forma, a partir daqui ao citar o índice i o que está sendo indicado é a posição do valor de λ dentro do vetor Lambda ; outra informação relevante é que para i pequeno o valor de λ_i está mais próximo de 0, do contrário se i aumenta, então os valores de λ_i se aproximam de 1.

Assim, é possível graficar em escala logarítmica os gráficos da norma do resíduo em função de λ e também a norma da solução. Ressaltando mais uma vez que os gráficos em função de i indicam a posição de λ_i . A figura 17 mostra a norma do resíduo que apresenta uma condição estacionária até $i = 82$, após esse índice o resíduo aumenta, ou seja, nos distanciamos da solução. Agora, a figura 18 mostra o norma da solução $\|x_\lambda\|$ decrescendo a medida que os λ_i aumentam (se aproximam de um), é possível associar a informação desse gráfico juntamente ao comportamento dos erros de regularização e perturbação da figura 15 e entender que a solução estará mais próxima da solução exata ao considerar a escolha do parâmetro λ grande.

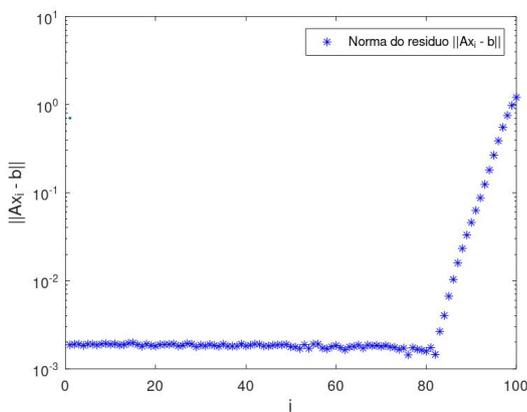


Figura 17 – Norma do resíduo como função de λ_i (Fonte: Autora).

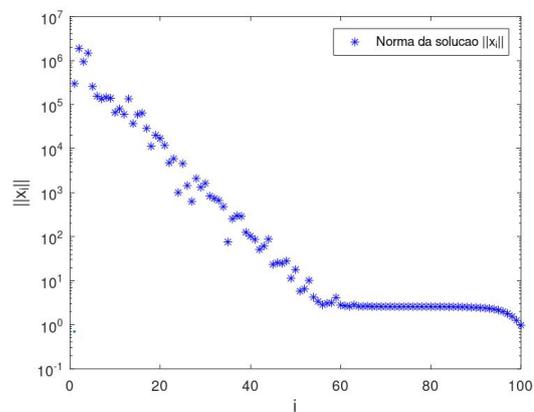


Figura 18 – Norma da solução x_λ como função de λ_i (Fonte: Autora).

Graficando os pares de informações $(\log \|Ax_\lambda - b\|_2, \log \|x_\lambda\|_2)$ observamos o gráfico da Curva-L (figura 19). Onde há indicação no canto superior dos pontos com os menores valores de i e na parte inferior observamos os pontos associados aos maiores valores de i . A solução de quadrados mínimos que corresponde à solução ótima a qual estará exatamente no canto da Curva-L, isso porque aumentar i fará com que a norma da solução seja maior e não necessariamente irá diminuir o resíduo. Outra consideração que pode ser feita é que a norma $\|x_\lambda\|$ aumenta devido ao aumento do ruído adicionado na solução.

De fato a Curva-L é uma ferramenta valiosa para determinar uma aproximação adequada para a solução, diminuindo a influência dos ruídos e perturbações.

Idealmente, o próximo passo seria interpolar os pontos do gráfico da figura 19. Devido às características da curva e às mudanças abruptas locais (como no vértice da curva), a interpolação por splines seria a abordagem mais adequada. Com isso, seria possível usar técnicas de otimização para localizar o ponto de mínimo local da curva e, assim, determinar o melhor valor de λ .

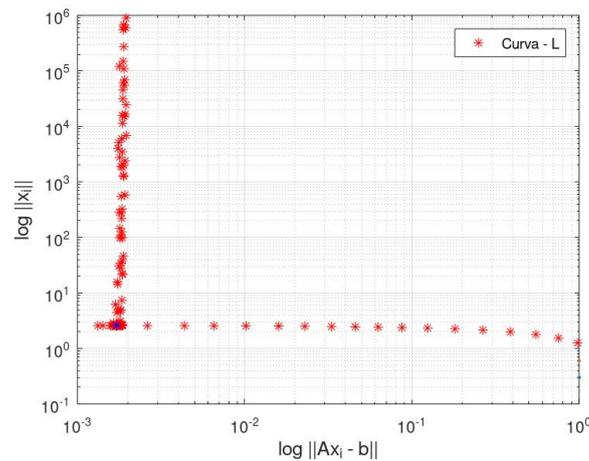


Figura 19 – Curva-L traçada a partir de $(\log \|Ax_\lambda - b\|_2, \log \|x_\lambda\|_2)$ (Fonte: Autora).

Ao invés disso, de maneira intuitiva cabe a avaliação realizada anteriormente observando os gráficos 17 e 18. Na figura 17, os pontos gerados em função de $i = 76$ e $i = 82$ apresentam as melhores aproximações da solução, o que sugere que ambos são candidatos para a solução.

Veja nas figuras abaixo, a reconstrução da solução utilizando: $\lambda_{50}, \lambda_{76}, \lambda_{82}$ e λ_{100} . De fato, a melhor aproximação é a gerada por $\lambda_{76} = 3.7649 \times 10^{-3}$. Para λ pequenos o efeito do filtro é baixo causando soluções oscilantes como na figura 20 (efeito das baixas frequências de σ_i), já para os maiores λ a maior influência vem da perturbação de ruído adicionada e , como mostrado na figura 23. Já nas figuras 21 e 22, o efeito de balanceamento entre Δx_{reg} e Δx_{pert} pode ser notado produzindo boas soluções, observado também avaliando os resíduos na tabela abaixo.

	λ_{50}	λ_{76}	λ_{82}	λ_{100}
$\ Ax_\lambda - b\ $	4.0194×10^{-3}	3.8030×10^{-3}	3.9448×10^{-3}	2.7010

Tabela 4 – Resíduo das soluções geradas através do método de Tikhonov utilizando determinação de parâmetros pela Curva-L.

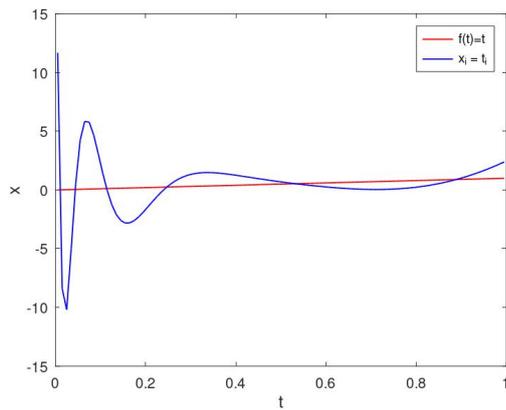


Figura 20 – Solução x_{50} com λ_{50} (Fonte: Autora).

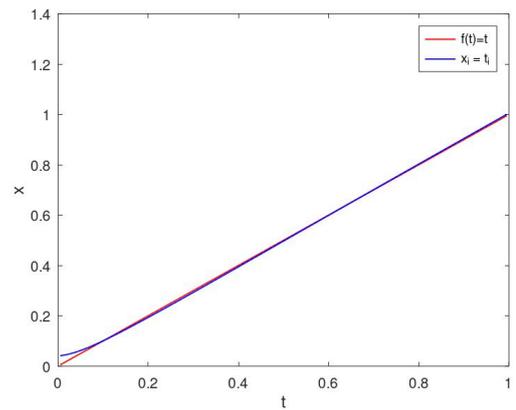


Figura 21 – Solução x_{76} com λ_{76} (Fonte: Autora).

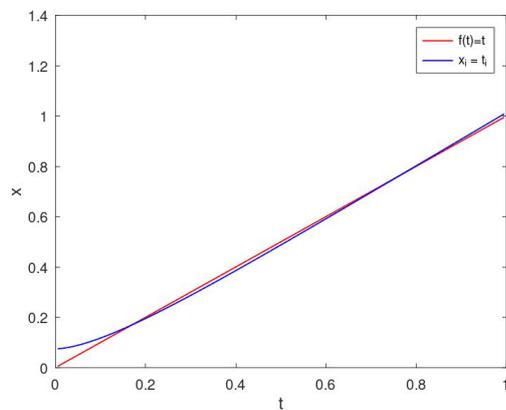


Figura 22 – Solução x_{82} com λ_{82} (Fonte: Autora).

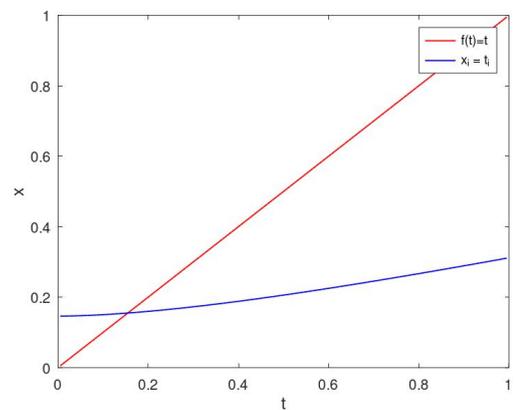


Figura 23 – Solução x_{100} com λ_{100} (Fonte: Autora).

Outra forma de determinar o λ é através do Princípio de Discrepância de Morozov. Porém, é necessário conhecer uma estimativa do nível do erro nos dados, o que faz com que sua aplicação em problemas reais seja mais difícil. Para o caso do problema implementado, a estimativa de erro utilizada foi $\nu_{DP} = 10^{-3}\sqrt{n}$ com $n = 20$. Observando o gráfico da figura 24, o parâmetro indicado é $\lambda_{86} = 0.0385$ o qual aproxima a solução com um resíduo de $\|Ax_\lambda - b\| = 1.0240 \times 10^{-2}$, observado no gráfico da figura 25.

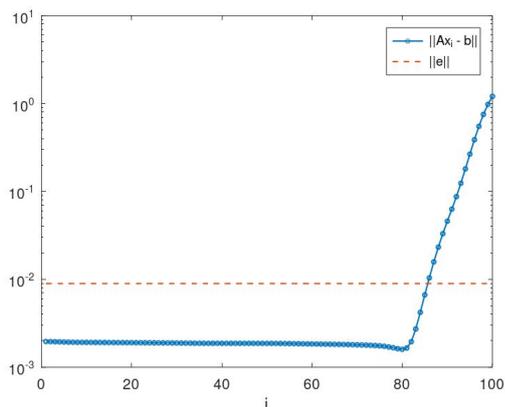


Figura 24 – Princípio de Discrepância de Morozov. (Fonte: Autora).

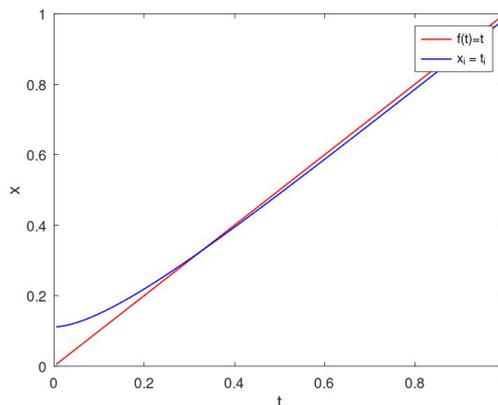


Figura 25 – Solução do problema a partir da escolha $\lambda_{86} = 0.0385$ (Fonte: Autora).

Por fim, comparando as melhores soluções obtidas através dos três métodos: TSVD (figura 14), Regularização de Tikhonov combinado com avaliação da Curva-L (figura 21) e Regularização de Tikhonov combinado com o Princípio de Discrepância de Morozov (figura 25), o que podemos observar é que ocorre maior precisão ao observar mesmo que de forma intuitiva a solução pela Curva-L.

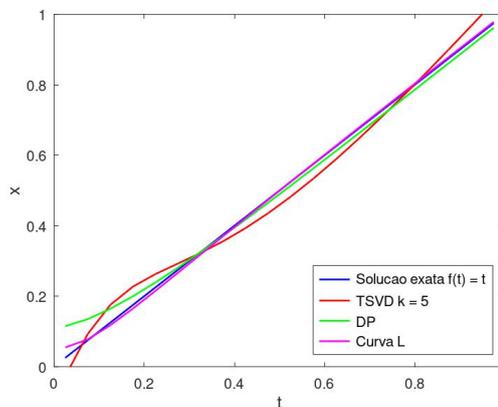
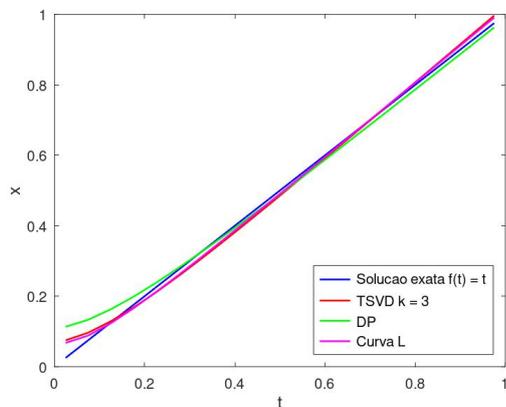


Figura 26 – Soluções obtidas pelos métodos de regularização testados (Fonte: Autora).

	TSVD			Tikhonov
	$\Delta x_{reg} \times \Delta x_{pert}$	Cond. Picard	Curva-L	Princípio da Discrepância
n	20	20	20	20
k	3	5	–	–
λ	–	–	3.7646×10^{-3}	0.0385
$\ Ax - b\ $	1.9694×10^{-3}	2.2002×10^{-2}	3.8030×10^{-3}	1.0240×10^{-2}
$\frac{\ Ax - b\ }{\ b\ }$	9.8127×10^{-4}	1.0963×10^{-2}	1.8949×10^{-3}	5.1021×10^{-3}
$\ t - x\ $	5.7984	5.8317	5.8046	5.7698

Tabela 5 – Resultados obtidos pelos métodos de regularização testados.

7.2 RESTAURAÇÃO DE SINAIS UNIDIMENSIONAIS

Um dos campos em que os desafios dos problemas inversos são facilmente encontrados é na restauração de sinais e imagens. Essa aplicação tem por objetivo a restauração de um sinal unidimensional através dos métodos de regularização apresentados anteriormente. Um primeiro ponto de atenção é notar que ao trabalhar com um sinal unidimensional estaremos trabalhando novamente com integrais de primeira espécie, porém em um contexto mais específico.

Considere um sinal unidimensional, ou seja, um vetor

$$\mathbf{u} = [0, \dots, 0, \frac{1}{4}, \frac{1}{2}, \frac{1}{4}, 0, \dots, \frac{1}{2}, \dots, 0, 1, 0 \dots, 0]^T \in \mathbb{R}^{200}. \quad (7.2)$$

Aqui, \mathbf{u} (em negrito) será usado para representar o sinal unidimensional definido anteriormente, a fim de evitar confusão com o vetor singular u_i , associado à matriz U da fatoração SVD.

Além disso, considere a definição abaixo de matriz Toeplitz:

Definição 2. *Uma matriz finita, ou matriz infinita dupla (ou seja, matriz com entradas nas posições (m, n) para $m, n \in \mathbb{Z}$) ou uma matriz infinita individual (ou seja, matriz com entradas nas posições (m, n) para m e n inteiros não negativos) é chamada matriz Toeplitz, se suas entradas são constante ao longo de cada diagonal. Isto é, a matriz $(t_{m,n})$ é Toeplitz se*

$$t_{m_1, n_1} = t_{m_2, n_2} \text{ onde } m_1 - n_1 = m_2 - n_2.$$

Dessa forma, seja T , uma matriz Toeplitz e simétrica, dada pela seguinte relação:

$$T = [t_{j,k}] \in \mathbb{R}^{200 \times 200}, \quad t_{j,k} = c \exp\left(-\frac{1}{10}(j-k)^2\right), \quad 1 \leq j, k \leq 200, \quad (7.3)$$

onde $c \approx \frac{1}{5.6}$ foi escolhido para que a soma das entradas de cada coluna seja igual a 1. A figura 27 ilustra em dois gráficos as entradas da matriz, o primeiro à direita em escala normal e o segundo à esquerda em escala logarítmica, veja que a matriz é mais ‘densa’ em sua diagonal.

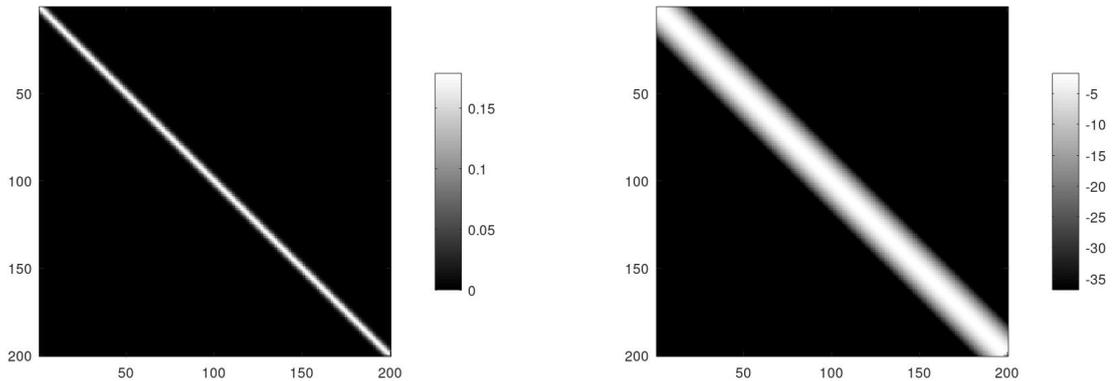


Figura 27 – Entradas da matriz de perturbação T . À direita na escala original e à esquerda as entradas da matriz em escala logarítmica.

Assim, \mathbf{u} é o sinal original (exato) e T é a matriz de dispersão do sinal. Ou seja, T é a responsável pelo efeito ‘blurring’ (desfoque) no problema de restauração de sinal. Note que ao multiplicar as componentes obtemos v , o sinal contaminado. Problemas onde através de v busca-se determinar o sinal original \mathbf{u} são chamados de problemas de ‘deblurring’, ou restauração de sinais.

O motivo de T causar uma perturbação no sinal de \mathbf{u} é devido ao seu mal condicionamento. Ao calcular o número de condição da matriz T é possível observar o quão mal condicionada ela é, veja:

$$\text{cond}(T) = \frac{\sigma_{200}}{\sigma_1} = \frac{3.9865 \times 10^{-11}}{1.0003} = 2.5092 \times 10^{10}.$$

Na figura 28, é possível observar o efeito de T sobre o sinal \mathbf{u} . Em vermelho, o sinal original \mathbf{u} , e, em azul v o sinal desfocado gerado pela produto $T\mathbf{u}$.

Porém, em modelos reais não existe apenas o efeito do desfoque, os instrumentos de medição também podem gerar ruídos na medição do sinal. Portanto, será adicionado ao sinal desfocado um ruído, ou seja, $w = v + e$, onde e é o ruído adicionado. Para esse, exemplo e foi

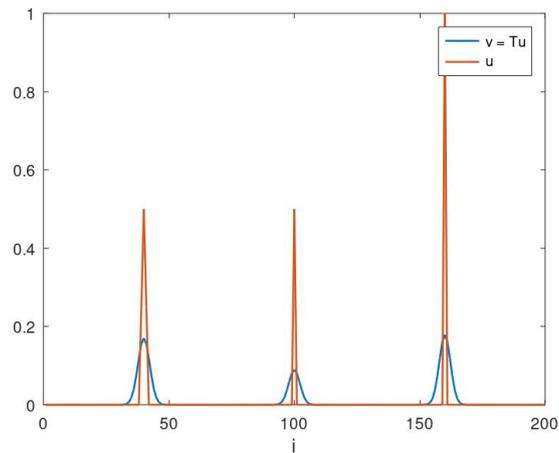


Figura 28 – Sinal original \mathbf{u} e sinal perturbado v .

gerado através da função `rand` e normalizado conforme a seguir:

$$e = \frac{\|\mathbf{u}\|}{\|e\|} 10^{-3}. \quad (7.4)$$

Agora, através de w , o sinal desfocado e com ruídos, vamos restaurar \mathbf{u} . Uma primeira tentativa seria resolver o sistema $\mathbf{u}_{\text{ruído}} = T^{-1}w$, mas como T é mal condicionada, o problema inverso é mal posto e a solução ingênua é ruim, como pode ser observado no gráfico da figura 29.

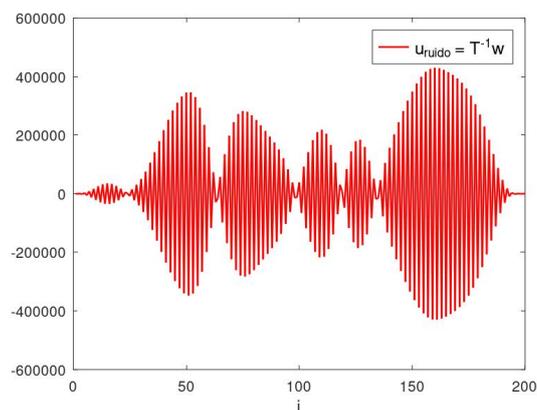


Figura 29 – Restauração $\mathbf{u}_{\text{ruído}} = T^{-1}w$

Essa instabilidade observada está diretamente associada a variação entre as grandezas dos valores singulares, σ_i , associados a T . Sendo assim, é possível observar que a solução fica instável devido aos valores singulares de altas frequência que amplificam pequenas perturbações nos dados (ver a figura 30).

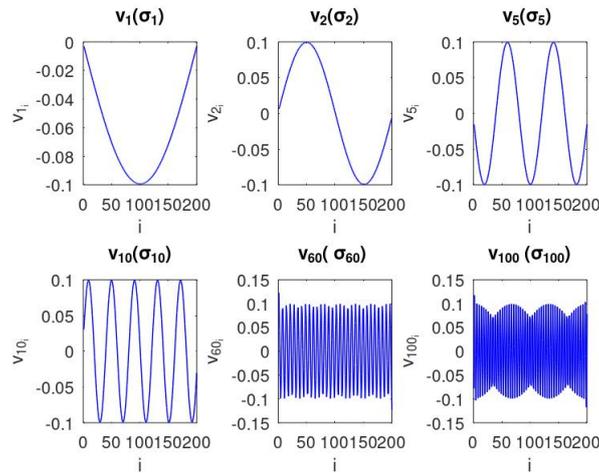


Figura 30 – Comportamento dos autovetores associados a T .

Observando a Condição Discreta de Picard, o efeito oscilatório visto na figura 29, é provocado pela rápida queda dos valores singulares quando comparados aos coeficientes $u_i^T \mathbf{u}_{\text{ruído}}$ (u_i indica o vetor-coluna da matriz U da fatoração SVD de T , e $\mathbf{u}_{\text{ruído}}$ é o sinal após adição da perturbação). Isto é, ao observar o gráfico da figura 31 a Condição Discreta de Picard não é satisfeita para índices maiores do que 125. Portanto, é razoável supor que ao truncar a SVD de T nas proximidades do índice $i = 125$, a restauração de \mathbf{u} pode ser bem-sucedida.

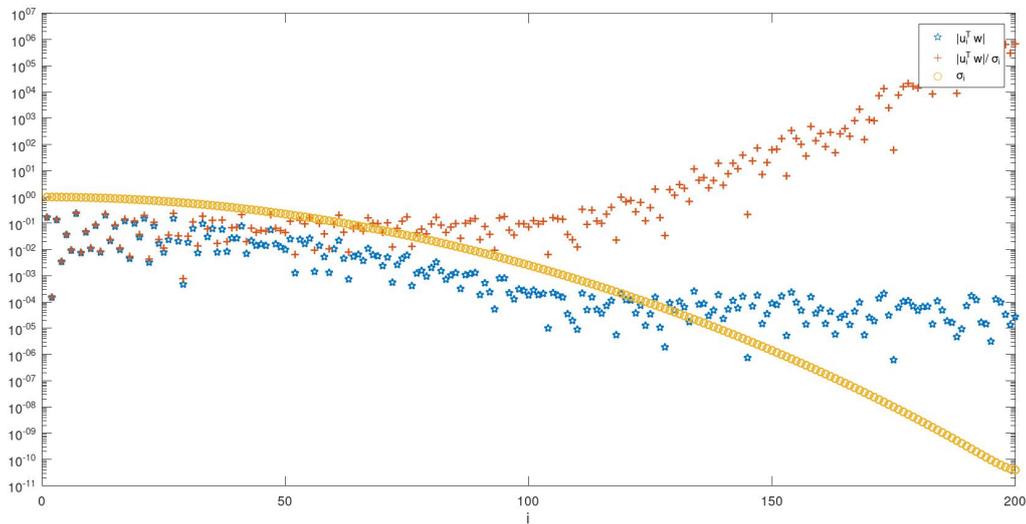


Figura 31 – Condição Discreta de Picard para o problema de restauração de sinal

Através do método da TSVD, com o parâmetro de truncamento $k = 125$, os valores singulares σ_i de maior frequência serão anulados. O resultado obtido é o sinal restaurado $\mathbf{u}_k = T_k^\dagger w$ graficado juntamente com o sinal original \mathbf{u} na figura 32, avaliando o gráfico em conjunto com a norma erro da aproximação, $\|\mathbf{u}_k - \mathbf{u}\| = 1.3101$ e a norma do resíduo $\|T\mathbf{u}_k - w\| = 7.0316$,

conclui-se que \mathbf{u}_{125} não é uma boa restauração do sinal original.

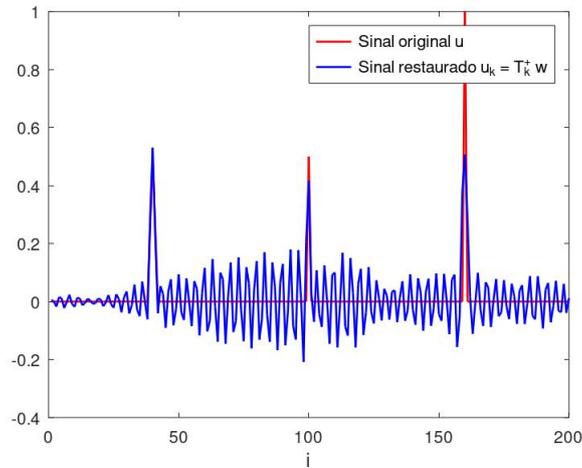


Figura 32 – Solução regularizada pelo método da TSVD com $k = 125$

Logo, visando observar o erro mínimo e equilibrar os erros de perturbação e regularização, com algumas adaptações foi utilizado o método da Curva-L para determinar o parâmetro de truncamento para o método de regularização TSVD. Ou seja, as simulações numéricas realizadas foram através de todas as soluções obtidas ao truncar T , n vezes, a fim de obter todas as soluções de T_k^\dagger no método da TSVD. O passo seguinte foi graficar em escala logarítmica os pontos obtidos através da norma da solução $\|\mathbf{u}_k\|$ e a norma do resíduo $\|T\mathbf{u}_k - w\|$, ou seja, $(\log \|T\mathbf{u}_k - w\|, \log \|\mathbf{u}_k\|)$. A figura 33 indica que $k = 110$ é o melhor índice para o truncamento da TSVD afim de restaurar \mathbf{u} , e avaliando a norma do erro na figura 34 entende-se que escolher um parâmetro k muito grande adiciona ruídos no sinal restaurado.

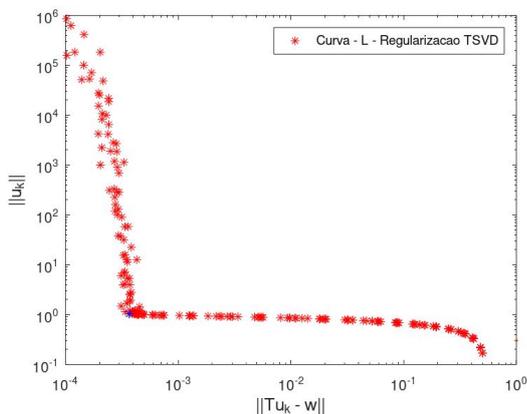


Figura 33 – Determinação do parâmetro de truncamento $k = 110$ através da Curva-L (Fonte: Autora).

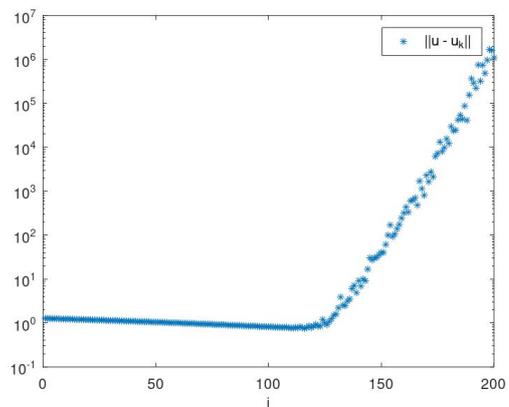


Figura 34 – Norma do erro como função de k na aproximação de \mathbf{u} pelo método TSVD (Fonte: Autora).

Agora se valendo do método de Regularização de Tikhonov, utilizando o mesmo

processo para determinar o λ_i que produzirá a solução ótima, foi graficada a Curva-L através dos pontos $(\log \|T\mathbf{u}_\lambda - w\|, \log \|\mathbf{u}_\lambda\|)$, e ainda, considerando um vetor em \mathbb{R}^{200} com valores equidistantes para λ no intervalo $(10^{-10}, 1)$. Portanto, o gráfico apresenta no eixo horizontal o índice i o qual indica a posição do valor λ_i no vetor utilizado para as simulações (ver as figuras 35 e 36).

Após obter a curva de pontos discretos indicada no gráfico da figura 35, através de uma interpolação por splines a Curva-L estará definida para todos os valores de λ no intervalo $(0, 1)$ e assim é possível através de técnicas de otimização obter o ponto de curvatura máxima da Curva-L plotada. Porém, ao invés disso avaliando o gráfico da figura 36, a escolha do λ deve estar ao redor de 10^{-5} e 10^{-4} , pois quando λ_i é muito pequeno o efeito da regularização é quase nulo, dessa forma ao aumentar o valor λ é possível encontrar um λ ótimo que ocorre em $i = 135$.

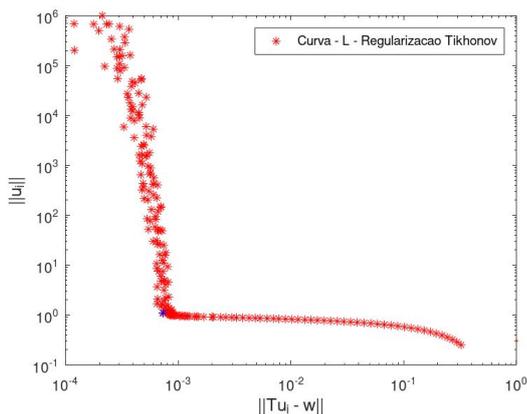


Figura 35 – Determinação $\lambda_{135} = 5.4159 \times 10^{-4}$ através da Curva-L (Fonte: Autora).

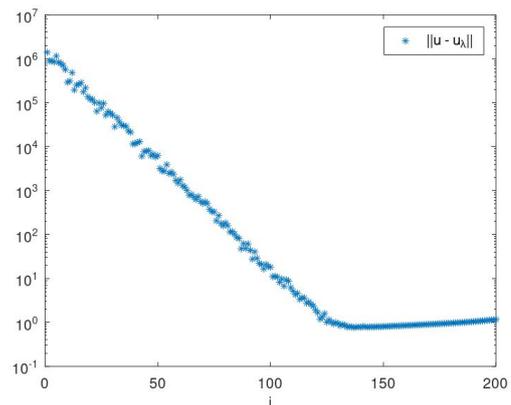


Figura 36 – Norma do erro como função de i na aproximação de \mathbf{u} pelo método Tikhonov (Fonte: Autora).

De fato a Curva-L é uma ferramenta valiosa para realizar a restauração de um sinal sem ruído a partir de um sinal desfocado e com algum erro de medição. Abaixo veja as duas aproximações obtidas através dos métodos de Regularização da TSVD e Tikhonov, com parâmetros de truncamento $k = 110$ e regularização $\lambda_{135} = 5.4159 \times 10^{-4}$, determinados pelo método da Curva-L (figuras 37 e 38).

O efeito das duas regularizações é bastante semelhante. As duas reconstruções reproduzem bem o primeiro e o segundo pulso do sinal \mathbf{u} e igualmente não sejam tão assertivas no último pulso, porém, note que pelo método de regularização TSVD se mostra menos sensível as perturbações se observarmos os pontos nulos. O sinal nulo restaurado no eixo horizontal

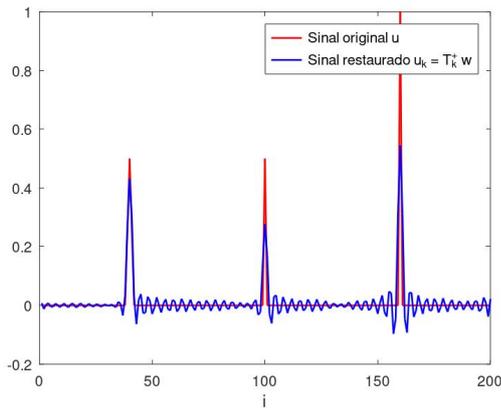


Figura 37 – Restauração de \mathbf{u} através do método da TSVD e Curva-L (Fonte: Autora).

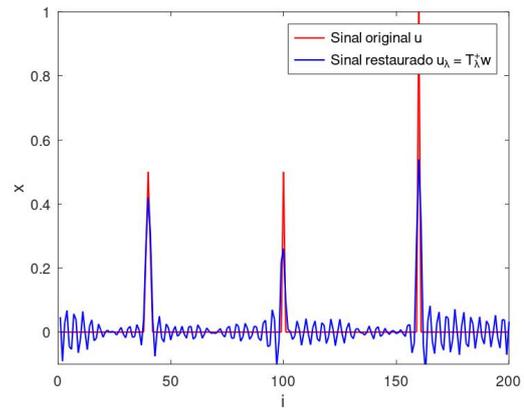


Figura 38 – Restauração de \mathbf{u} através do método de Tikhonov e Curva-L (Fonte: Autora).

recebe menos interferência observado pelo método da TSVD do que pelo método de Tikhonov.

Uma outra forma de determinar um valor para λ para o método de Tikhonov é utilizar a estimativa de erro introduzido no sinal \mathbf{u} e avaliar em que ponto a norma do resíduo $\|T\mathbf{u}_\lambda - w\|$ passa a ser maior do que essa estimativa. Para esse método o erro estimado foi calculado através de um fator de segurança ν_{PD} definido por:

$$\nu_{DP} = 10^{-3} \sqrt{n}.$$

Esse fator de segurança foi multiplicado pela norma de \mathbf{u} , gerando o fator de discrepância $\|e\|$ representado no gráfico da figura 39 com a linha tracejada em vermelho.

O resíduo na figura 39 é uma função crescente associado aos valores de λ que se aproximam de 1. Dessa forma, através do Princípio de Discrepância de Morozov é determinado como parâmetro para a Regularização de Tikhonov $\lambda_{178} = 7.8428 \times 10^{-2}$.

Ao realizar a restauração do sinal de \mathbf{u} pelo método de Tikhonov com λ_{178} , figura 40, veja que existe uma perda de qualidade na reprodução dos três picos do sinal. Isso porque ao aumentar o valor de λ aproximando o mesmo de 1 existe uma perda de informações relevantes dos valores singulares σ_i .

A tabela abaixo mostra de maneira resumida os resultados obtidos na restauração de \mathbf{u} através dos métodos estudados (a notação \mathbf{u}_{reg} foi utilizada para indicar o vetor \mathbf{u} após regularização). Veja que para essa aplicação o melhor desempenho foi obtido através do método de regularização TSVD através da determinação do parâmetro k pela avaliação da Curva-L e o gráfico de erro.

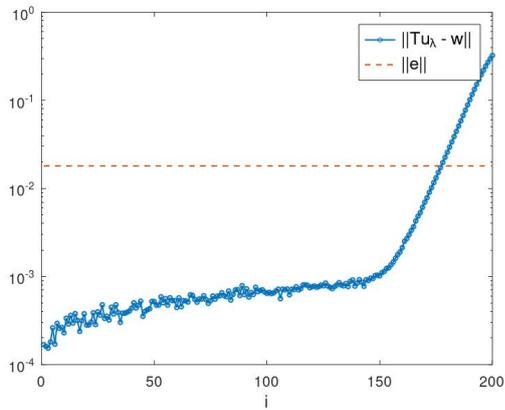


Figura 39 – Determinação de parâmetro λ_{178} através do Princípio de Discrepância de Morozov (Fonte: Autora).

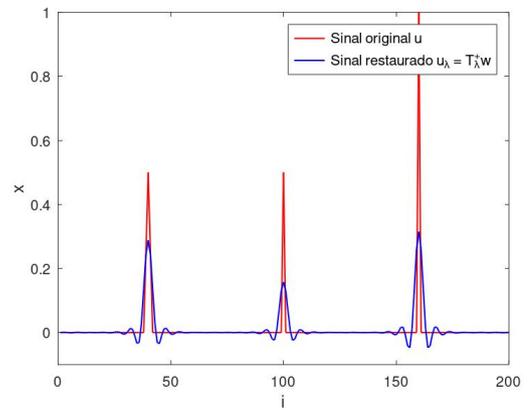


Figura 40 – Restauração de \mathbf{u} através do método de Tikhonov e Princípio de Discrepância de Morozov (Fonte: Autora).

	TSVD		Tikhonov	
	Curva-L	Cond. Picard	Curva-L	Princípio da Discrepância
n	200	200	200	200
k	110	125	–	–
i	–	–	135	178
λ	–	–	5.4159×10^{-4}	7.8428×10^{-2}
$\ T\mathbf{u}_{\text{reg}} - w\ $	7.0380	7.0316	8.7653×10^{-4}	1.9691×10^{-2}
$\frac{\ T\mathbf{u}_{\text{reg}} - w\ }{\ w\ }$	13.340	13.328	1.6614×10^{-3}	3.7168×10^{-2}
$\ \mathbf{u} - \mathbf{u}_{\text{reg}}\ $	0.7703	1.3101	0.8340	0.9501

Tabela 6 – Resultados obtidos para regularizar \mathbf{u} pelos métodos de regularização estudados.

8 CONCLUSÃO

Este estudo teve por objetivo introduzir os métodos de regularização de Tikhonov e TSVD como ferramentas na solução de problemas inversos discretos, buscando estabilizar a solução e diminuir a influência das perturbações e ruídos nos dados iniciais do problema. Nas aplicações práticas foi possível simular resultados com diferentes parâmetros de regularização, o que enfatizou a importância das técnicas de determinação de parâmetros também estudadas.

Através dos experimentos computacionais foi possível avaliar que o Método Regularização de Tikhonov é um método mais robusto comparado ao Método TSVD, pois seu intuito é balancear o erro da aproximação da solução e penalizar as componentes que causam instabilidade através da adição de um parâmetro de regularização. Já o Método da TSVD exclui tais componentes do problema, dessa forma a escolha do parâmetro de truncamento é bastante crítica, uma vez que pode existir a perda de informações essenciais do problema ou aumentar os erros de perturbação na solução. Quanto às técnicas de determinação de parâmetros, pode-se ressaltar que o Princípio de Discrepância de Morozov apresentou nos exemplos implementados bons resultados, porém em problemas reais possui um limitador em relação à necessidade de conhecer uma estimativa do nível do ruído presente nos dados do problema; e a técnica de regularização através da Curva-L é bastante custosa computacionalmente visto a necessidade de resolver n problemas inversos com diferentes parâmetros de regularização.

Contudo, este trabalho apresenta algumas limitações, principalmente no que diz respeito à determinação dos parâmetros “ótimos”. Seria importante para o estudo integral do método da Curva-L, fazer uso de técnicas de interpolação para obter uma curva contínua e assim, através de ferramentas de otimização, determinar o parâmetro de regularização. Além disso, a análise experimental foi bastante restrita, implicando em uma avaliação não tão profunda dos desafios enfrentados em problemas inversos discretos.

Para trabalhos futuros, seria interessante explorar a classe de métodos de regularização iterativos importantes na resolução de problemas de grande porte, tais como: o método Landweber, que se baseia no princípio de mínimos quadrados, ou até mesmo os métodos ART, que são métodos algébricos de reconstrução; além de discutir os possíveis critérios de parada e a eficiência dos métodos. Além disso, visto que uma das principais aplicações das técnicas de regularização é a reconstrução de imagens seria interessante poder abordar os problemas

inversos discretos nesse contexto como uma sequência natural da aplicação de restauração de sinal unidimensional.

Tendo em vista todos os tópicos abordados até aqui e as muitas possibilidades de aplicações dos problemas inversos discretos, este trabalho contribui como um estudo introdutório aos problemas inversos discretos e à compreensão das técnicas de regularização abordadas. Da mesma forma, os experimentos numéricos aqui explorados foram indispensáveis para a compreensão das dificuldades associadas aos problemas inversos discretos lineares, esses experimentos se mostraram como ferramentas práticas para validar os estudos teóricos realizados a respeito do impacto de erros e perturbações, para explorar diferentes métodos e para avaliar a sensibilidade e estabilidade dos resultados.

REFERÊNCIAS

- FSV BAZÁN and LS BORGES. Métodos para problemas inversos de grande porte. *Notas em Matemática Aplicada – SBMAC*, 39, 2009.
- J D.; BURDEN Annette M. BURDEN, Richard L.; FAIRES. *Análise Numérica*. São Paulo: Cengage Learning Brasil, 2016.
- A DE CEZARO and FT DE CEZARO. Problemas inversos em tomografias. *Notas em Matemática Aplicada – SBMAC*, 59, 2012.
- Charles W GROETSCH. *Inverse problems in the mathematical sciences*. Springer, 1993.
- Per Christian HANSEN. *Discrete Inverse Problems*. Society for Industrial and Applied Mathematics, 2010.
- H KEKKONEN and Y KOROLEV. Inverse problems – lecture notes. *University of Cambridge*, 2019.
- Elon Lages LIMA. *Álgebra linear*. Rio de Janeiro: IMPA, 2014.
- Lila L P PAREDES et al. Métodos de regularização iterativos para problemas discretos mal postos de grande porte. *Universidade Federal de Santa Catarina*, 2014.
- Samuli SILTANEN. Inverse problems. *Lecture Notes – Tampere University of Technology*, 2009.
- Gilbert STRANG. *Álgebra Linear e suas aplicações*. São Paulo: Cengage Learning Brasil, 2014.

A IMPLEMENTAÇÕES. PROBLEMAS MAL POSTOS

Script utilizado para gerar as figuras 2 e 3.

```
%script utilizado para no exemplo de prob mal condicionado
clear all
close all
clc

A = [1 2.1; 2.1 3.99];
b_exato = [3.1 6.09]';
b_ruido = [3.09 6]';

display('Solucao EXATA Q.M')
x_exato = inv(A' * A) * A' * b_exato

display('Cond(A)')
cond(A)
display('Solucao Q.M')
x_min = inv(A' * A) * A' * b_ruido
display('Residuo Q.M')
res_qm = norm(A*x_min - b_ruido)
display('Residuo Relativo Q.M')
res_qm = norm(A*x_min - b_ruido)/norm(b_exato)
display('Erro Q.M')
erro_qm = norm(x_exato - x_min)
display('Erro Relativo Q.M')
erro_qm = norm(x_exato - x_min)/norm(x_exato)

plot(x_exato(1),x_exato(2), 'MarkerSize',25)
hold on
plot(x_min(1),x_min(2), 'MarkerSize', 25)
hold off
xlabel('(x_1)');
ylabel('(x_2)');
legend('Solucao x_{exato}', 'Solucao com perturbacao x_{min}','FontSize', 14)

plot(b_exato(1),b_exato(2), 'MarkerSize',25)
```

```
hold on
plot(b_ruido(1),b_ruido(2), 'MarkerSize', 25)
hold off
xlabel('(b_1)');
ylabel('(b_2)');
legend('Vetor b_{exato}', 'Vetor com perturbacao b_{pert}','FontSize', 14)
```

B IMPLEMENTAÇÕES. ESTABILIDADE

Script utilizado para gerar as figuras 4 e 5.

```
clear all
close all
clc

t = 0:0.01:1;

f1 = @(t) sin(2*pi*t)
f2 = @(t) sin(2*pi*2*t)

f8 = @(t) sin(2*pi*8*t)

plot(t,f1(t),'b','LineWidth',0.5)
hold on
plot(t,f2(t),'r','LineWidth',0.5)

plot(t,f8(t),'g','LineWidth',0.5)
grid on

legend('n = 1','n = 2', 'n = 8','FontSize', 14)

clear all
close all
clc

t = 0:0.01:1;

f1 = @(t) sinh(t)/(2*pi)
f2 = @(t) sinh(t)/(2*pi*2)

f8 = @(t) sinh(t)/(2*pi*8)
```

```
plot(t,f1(t),'b','LineWidth',1.0)
hold on
plot(t,f2(t),'r','LineWidth',1.0)

plot(t,f8(t),'g','LineWidth',1.0)
grid on
xlim([0, 1]);
ylim([0, 0.2]);
legend('n = 1','n = 2', 'n = 8', 'FontSize', 14)
```

C IMPLEMENTAÇÕES. PROBLEMAS INVERSOS DISCRETOS

Scripts utilizados para gerar 6 e 7.

```
function [M, b, x] = GroestchIntegral(n)
t = (2 * (1:n) - 1) / (2 * n)';

M = zeros(n,n);

for i = 1:n
    for j = 1:n
        M(i,j) = (1/n)* sqrt( t(i)^2 + t(j)^2);
    endfor
endfor

M;

b = zeros(1,n);
for i = 1:n
    b(i) = (1/3)*((1+t(i)^2)^(3/2) - t(i)^3);
endfor

b;

x = M\b';

endfunction
```


D IMPLEMENTAÇÕES. APLICAÇÕES: INTEGRAL DE PRIMEIRA ESPÉCIE

Script utilizado para gerar as figuras: [9](#), [10](#), [11](#), [12](#), [13](#), [14](#), [16](#) e [26](#).

```
% REGULARIZACAO DO PROBLEMA INVERSO DISCRETO ATRAVES DA TSVD
```

```
clear all
```

```
close all
```

```
clc
```

```
n= 20;
```

```
[M, b_exato] = GroestchIntegral(n);
```

```
[U, S, V] = svd(M);
```

```
ruido = rand(1,n);
```

```
ruido_norm = ruido/norm(ruido);
```

```
erro = (10^-3)*norm(b_exato) * ruido_norm';
```

```
b_ruido = b_exato' + erro;
```

```
t = (2 * (1:n) -1) / (2 * n);
```

```
Diag_S = diag(S);
```

```
filt = [];
```

```
par_Trunc =20; # declarar parametro de regularizacao TSVD
```

```
for i = 1:n
```

```
    if i < par_Trunc
```

```
        f_i = 1;
```

```
    else
```

```
        f_i = 0;
```

```
    endif
```

```
    filt = [filt f_i];
```

```
endfor
```

```
Filt = diag(filt);
```

```
A = V*Filt*(pinv(S))*U';
```

```
cond(M);
```

```
cond(A);
```

```
x1 = A*b_ruido
```

```

x2 = A*b_exato';
ERRO_ruido = norm(t - x1)
ERRO_exato = norm(t - x2)
residuo_ruido= norm(M*x1-b_exato')
residuo_sem_ruido=norm(M*x2-b_exato')

%PLOT DO COMPORTAMENTO DOS VETORES SINGULARES ASSOCIADOS A MATRIZ M

AutoVetor1 = V(:,1);
S_1 = S(:,1);
AutoVetor2 = V(:,2);
S_2 = S(:,2);
AutoVetor5 = V(:,5);
S_5 = S(:,5);
AutoVetor10 = V(:,10);
S_10 = S(:,10);
AutoVetor20 = V(:,20);
S_20 = S(:,20);
AutoVetor50 = V(:,50);
S_20 = S(:,50)

i = 1:1:n
subplot(2,3,1);
plot(i,AutoVetor1,'b','LineWidth',0.7);
title('v_1 - \sigma_1 = 8.10^{-1}');
subplot(2,3,2);
plot(i,AutoVetor2,'b','LineWidth',0.7);
title('v_2 - \sigma_2 = 9.56^{-2}');
subplot(2,3,3);
plot(i,AutoVetor5,'b','LineWidth',0.7);
title('v_5 - \sigma_5 = 2.20^{-4}');
subplot(2,3,4);
plot(i,AutoVetor10,'b','LineWidth',0.7);
title('v_{10} - \sigma_{10} = 2.65^{-8}');
subplot(2,3,5);
plot(i,AutoVetor20,'b','LineWidth',0.7);
title('v_{20} - \sigma_{20} = 5.54^{-19}');
subplot(2,3,6);
plot(i,AutoVetor50,'b','LineWidth',0.7);

```

```

title('v_{30} - \sigma_{50} = 1.36^{-19}');

% CONDICAO DISCRETA DE PICARD
T1 = [];
T2 = [];

for i = 1:n
    teste1 = norm(U(:,i)) * b_ruido;
    teste2 = teste1 / Diag_S(i);
    T1 = [T1 teste1];
    T2 = [T2 teste2];
endfor

i_valores = 1:n;

h1 = semilogy(i_valores, T1, 'p', 'MarkerSize', 6, 'LineWidth', 1.2);
hold on
h2 = semilogy(i_valores, T2, '+', 'MarkerSize', 6, 'LineWidth', 1.2);
h3 = semilogy(i_valores, Diag_S, 'o', 'MarkerSize', 6, 'LineWidth', 1.2);
hold off;

SizeFontlabel = 12;
xlabel('i', 'fontsize', SizeFontlabel);
legend([h1, h2, h3], {'|u_i^Tb|', '|u_i^Tb|/ \sigma_i', '\sigma_i'}, 'Location', 'northeast');

%Plot das solucoes obtidas atraves do metodo da TSVD com e sem ruido

plot(t,t,'b','LineWidth',1.0);
hold on
plot(t,x1,'-g','LineWidth',1.2);
%plot(t,x2,'--r','LineWidth',1.2);
hold off
SizeFontlabel = 12;
%ylim([0 1.0]);
xlabel('t','fontsize',SizeFontlabel);
ylabel('x','fontsize',SizeFontlabel);
legend('Solucao exata f(t) = t','Solucao aprox. com ruido','FontSize', 10);
%legend('Solucao exata f(t) = t','Solucao aprox. sem ruido','FontSize', 10);

```

```
%legend('Solucao exata f(t) = t','Solucao aprox. com ruido','Solucao aprox. sem ruido','FontSize', 10,  
  
%gera o grafico com todas as solucoes regularizadas  
  
Lambda = logspace(-10,0,100);  
  
[x3, res3] = tikhonovPlotL(n, Lambda(86))  
[x4, res4] = tikhonovPlotL(n, Lambda(76))  
  
plot(t,t,'b','LineWidth',1.2)  
hold on  
plot(t,x1,'r','LineWidth',1.2)  
plot(t,x3,'g','LineWidth',1.2)  
plot(t,x4,'m','LineWidth',1.2)  
hold off  
ylim([0 1.0]);  
xlabel('t','fontsize',SizeFontlabel);  
ylabel('x','fontsize',SizeFontlabel);  
legend('Solucao exata f(t) = t', 'TSVD k = 5', 'DP', 'Curva L','FontSize', 10, 'Location', 'southeast')  
  
erro_TSVD = norm(t - x1)  
erro_DP = norm(t - x3)  
erro_CL = norm(t - x4)
```

Script utilizado para gerar a figura 15.

```
% script para avaliar o erro de regularizacao e perturbacao associado  
% a regularizacao TSVD  
  
clear all  
close all  
clc  
  
n=20;  
[M, b_exato, x] = GroestchIntegral(n);  
[U, S, V] = svd(M);  
Diag_S = diag(S)
```

```

t = ((2 * (1:n) - 1) / (2 * n))';
ruído = rand(1,n);
ruído_norm = ruído;
erro = (10^-3)* norm(b_exato)*ruído_norm';
b_ruído = b_exato' + erro;

error = [];
bias = [];
pert = [];

x_k = zeros(1,n)';
x_k_ruído = zeros(1,n)'

for k = 1: n
    x_ruído = zeros(1,n)';
    x_i = zeros(1,n)';
    for i = 1: k
        i
        Diag_S(i)
        x_i = x_i+ ((U(:,i)'*b_exato')/Diag_S(i))*V(:,i)
        x_ruído = x_ruído + ((U(:,i)'*b_ruído)/Diag_S(i))*V(:,i)
    endfor
    x_k = [x_k x_i];
    x_k_ruído = [x_k_ruído x_ruído];

    error = [error norm(t - x_k_ruído(:,k+1))]    % erro da da solucao
    bias = [bias norm(t-x_k(:,k+1))]              % erro de perturbacao
    pert = [pert norm(x_i - x_k_ruído(:,k+1))]    % erro de regularizacao

endfor
i = 1:n;

plot(i, error, '--k', 'LineWidth', 2.5);
hold on
plot(i, bias, '-b', 'LineWidth', 1.0);
plot(i, pert, '-r', 'LineWidth', 1.0);
hold off
ylim([0 5]);

```

```

SizeFontlabel=10;
xlabel('i','fontsize',SizeFontlabel);
legend('||t - x_k||', '||\Delta x_{reg}||', '||\Delta x_{pert}||', 'FontSize', 10)

```

Script utilizado para gerar as figuras: 17, 18, 19, 20, 21, 22, 22, 23 e 25.

```

% Simulacoes para Curva L e plot solucao regularizada Tikhonov
clear all
close all
clc

Vet_Aprox_x = [];
teste = [];
Vet_Resid = [];
Lambda = logspace(-6, 0,30);
n = 20;
t = (2 * (1:n) -1) / (2 * n)';
erro = [];
[M] = GroestchIntegral(n);

%calcula os n problemas inversos para a determinacao de parametro - Curva L
for i = 1:length(Lambda)
    Lambda(i);
    [x_i, res_i, b_ruido] = tikhonovPlotL(n, Lambda(i));
    Vet_Aprox_x = [Vet_Aprox_x x_i];
    Vet_Resid = [Vet_Resid res_i];
    norma_erro = norm( b_ruido' - M*x_i);
    erro = [erro norma_erro];
endfor

% Plot curva L
X = [];
Y = [];
for i = 1:length(Lambda)
    Norma_x = norm(Vet_Aprox_x(:,i));           %gerando vetor ||x_lambda||
    Norma_Resid = norm(Vet_Resid(:,i));         %%gerando vetor ||Ax_lambda - b||
    X = [X Norma_x];
    Y = [Y Norma_Resid];
endfor
loglog(Y ,X,'r*',0.3);

```

```

hold on
loglog(Y(76), X(76), 'b*', 0.6);
hold off
SizeFontlabel = 12;
xlim ([10^-3 10^0]);
ylim ([10^-1 10^6]);
legend('Curva - L');
xlabel('log ||Ax_i - b||','fontsize',SizeFontlabel);
ylabel('log ||x_i||','fontsize',SizeFontlabel);

#plot residuo e norma solucao
i_valores = 1:length(Lambda)

semilogy(i_valores, X, 'b*',0.7);          ### plot norma da solucao
legend('Norma da solucao ||x_i||');
xlabel('i','fontsize',SizeFontlabel);
ylabel('||x_i||','fontsize',SizeFontlabel);

semilogy(i_valores, Y, 'b*',0.7)          ### plot norma do residuo
legend('Norma do residuo ||Ax_i - b||');
xlabel('i','fontsize',SizeFontlabel);
ylabel('||Ax_i - b||','fontsize',SizeFontlabel);

% plota a solucao regularizada pelo metodo de tikhonov
% atraves da indicacao do parametro lambda determinado (curva -L ou DP Morozov)

[x, res] = tikhonovPlotL(n, Lambda(86));
display('Solucao x =')
x
display('Residuo: ')
res
display('Lambda Minimo: ')
Lambda(86)
plot(t,t, 'r', 'LineWidth',0.9);
hold on
plot(t,x, 'b', 'LineWidth',0.9);
hold off

```

```

SizeFontlabel = 12;
xlabel('t','fontsize',SizeFontlabel);
ylabel('x','fontsize',SizeFontlabel);
legend('f(t)=t', 'x_i = t_i');

function [xx1, b_exato, erro] = tikhonovPD(n, Lambda, ruido)
%METODO DE REGULARIZACAO DE TIKHONOV
%realizadas adaptacoes nas saidas para determinar o parametro por DP Morozov

[M, b_exato] = GroestchIntegral(n);
[U, S, V] = svd(M);

n = length(b_exato);
t = (2 * (1:n) - 1) / (2 * n);
ruido_norm = ruido;
erro = (10^-3)* norm(b_exato)*ruido_norm';
b_ruido = b_exato' + erro;
Diag_S = diag(S);
filt = [];

for i = 1:n
    f_i = ((Diag_S(i))^2)/((Diag_S(i))^2+(Lambda^2));
    filt = [filt f_i];
endfor

Filt = diag(filt);
A = V*Filt*(pinv(S))*U';

cond(M);
cond(A);
xx1=A*b_ruido ;
xx2=M*b_exato';
M*xx1;

residuo_ruido= norm(M*xx1 - b_exato');
residuo_sem_ruido=norm(M*xx2 - b_exato');

```

```

endfunction

function [xx1, residuo_ruido, b_ruido] = tikhonovPlotL(n, Lambda)
% REGULARIZACAO TIKHONOV
% adaptacoes nas saidas da funcao para determinar o parametro pela curva L

[M, b_exato] = GroestchIntegral(n);
[U, S, V] = svd(M);

t = (2 * (1:n) - 1) / (2 * n);
ruido = rand(1,n);
ruido_norm = ruido/norm(ruido);
erro = (10^-3)* norm(b_exato)*ruido_norm';
b_ruido = b_exato' + erro;
Diag_S = diag(S);
filt = [];

for i = 1:n
    f_i = ((Diag_S(i))^2)/((Diag_S(i))^2+(Lambda^2));
    filt = [filt f_i];
endfor

Filt = diag(filt);
A = V*Filt*(pinv(S))*U';

cond(M);
cond(A);
xx1=A*b_ruido ; %%% Alterado
xx2=A*b_exato' ; %%% Alterado
M*xx1;

residuo_ruido= norm(M*xx1-b_exato');
residuo_sem_ruido=norm(M*xx2-b_exato');

endfunction

```

Sript utilizado para gerar as figuras: 24.

```

##PRINCIPIO DA DISCREPANCIA DE MOROZOV
clear all
close all
clc

Lambda = logspace(-10, 0,100);
n = 20;
t = (2 * (1:n) -1) / (2 * n);
ruído = rand(1,n);
ruído_norm = ruído/norm(ruído);
e = norm(ruído_norm);

[M] = GroestchIntegral(n);

Vet_Resid = [];
nu_dp = 10-3*sqrt(n) # fator de segurança definido a partir do nível de ruído
resíduo_sol = [];

%simula n problemas inversos para o cálculo do resíduo
% afim de avaliar ||Ax_lambda - b|| = nu_dp||e||

for i = 1:length(Lambda)
    [x_i, b_exato, erro] = tikhonovPD(n, Lambda(i),ruído_norm);
    Vet_erro = [Vet_Resid erro];
    res_sol_xi = norm(b_exato' - M*x_i);
    resíduo_sol = [resíduo_sol res_sol_xi];
endfor

% Gera o gráfico para avaliação do princípio de discrepância de Morozov

norma_b = norm(b_exato)
nu_dp*norma_b

i = 1:length(Lambda)

Discrepancia = nu_dp*norma_b*ones(1,100);
semilogy(i, resíduo_sol, '-o', 'MarkerSize', 3, 'LineWidth', 1.0);

```

```
hold on
semilogy(i, Discrepancia, '--', 'MarkerSize', 3, 'LineWidth', 1.0);
hold off
xlim ([0 100]);
ylim ([10^-3 10^1]);
SizeFontlabel = 12;
xlabel('i','fontsize',SizeFontlabel);
legend('||Ax_i - b||', '||e||','fontsize',12);
```

```
[x, res] = tikhonovPlotL(n, Lambda(86))
```

```
plot(t,t)
hold on
plot(t,x)
hold off
t
```


E IMPLEMENTAÇÕES. APLICAÇÕES: SINAL UNIDIMENSIONAL

Script utilizado para gerar as figuras: [27](#), [28](#), [29](#), [30](#) e [31](#).

```
clear all
close all
clc

%Definicao do sinal de unidimensional u

n = 200;
u = zeros(1,n);
u(39)= 1/4;
u(40) =1/2;
u(41)= 1/4;
u(100) = 1/2;
u(160) = 1;

T = zeros(n,n);
%Definindo a matriz Toeplitz de perturbacao
T = zeros(n,n);
for j = 1:n
    for k = 1:n
        T(j,k) = (1/5.6)*exp(-1/10*(j-k)^2);
    endfor
endfor

%Plot da matrix Toeplitz
subplot(1,2,1);
imagesc(T);
colormap gray;
colorbar;
axis equal;

T = T - min(T(:)) + 10^-16;
LogT = log(T);
```

```
subplot(1,2,2);
imagesc(LogT);
colormap gray;
colorbar;
axis equal;

% comparando o sinal original e o sinal com perturbacao 'blurring'
v = T*u';
i = 1:n;

plot(i, v, 'LineWidth', 1.2);
hold on
plot(i, u, 'LineWidth', 1.2);
hold off
SizeFontlabel = 12;
xlabel('i', 'fontsize', SizeFontlabel);
legend({'v = Tu', 'u'}, 'Location', 'northeast');

%adicionando ruido
ruido = randn(n,1);
ruido_normalizado = ruido/norm(ruido);
erro = ruido_normalizado*norm(u)*10^(-3);
w = v + erro;

u_ruido = T\w;

plot(i,u_ruido,'r','LineWidth', 1.2)
SizeFontlabel = 12;
xlabel('i', 'fontsize', SizeFontlabel);
legend({'u_{ruido} = T^{-1}w'}, 'fontsize',12,'Location', 'northeast');

[U, S, V] = svd(T);
Diag_S = diag(S);

%avaliacao dos vetores singulares de T
```

```

AutoVetor1 = V(:,1);
S_1 = S(:,1);
AutoVetor2 = V(:,2);
S_2 = S(:,2);
AutoVetor5 = V(:,5);
S_5 = S(:,5);
AutoVetor10 = V(:,10);
S_10 = S(:,10);
AutoVetor60 = V(:,60);
S_60 = S(:,60);
AutoVetor100 = V(:,100);
S_100 = S(:,100);

i = 1:1:n
subplot(2,3,1);
plot(i,AutoVetor1,'b','LineWidth',0.7);
title('v_1(\sigma_1)');
subplot(2,3,2);
plot(i,AutoVetor2,'b','LineWidth',0.7);
title('v_{2}(\sigma_{2})');
subplot(2,3,3);
plot(i,AutoVetor5,'b','LineWidth',0.7);
title('v_{5}(\sigma_{5}) ');
subplot(2,3,4);
plot(i,AutoVetor10,'b','LineWidth',0.7);
title('v_{10}(\sigma_{10})');
subplot(2,3,5);
plot(i,AutoVetor60,'b','LineWidth',0.7);
title('v_{60}( \sigma_{60}) ');
subplot(2,3,6);
plot(i,AutoVetor100,'b','LineWidth',0.7);
title('v_{100} (\sigma_{100})');

%condicao discreta de picard

T1 = [];
T2 = [];

```

```
for i = 1:n
    teste1 = norm(U(:,i)' * w);
    teste2 = teste1 / Diag_S(i);
    T1 = [T1 teste1];
    T2 = [T2 teste2];
endfor

i_valores = 1:n;

h1 = semilogy(i_valores, T1, 'p', 'MarkerSize', 6, 'LineWidth', 1.2);
hold on
h2 = semilogy(i_valores, T2, '+', 'MarkerSize', 6, 'LineWidth', 1.2);
h3 = semilogy(i_valores, Diag_S, 'o', 'MarkerSize', 6, 'LineWidth', 1.2);
hold off;

SizeFontlabel = 12;
xlabel('i', 'fontsize', SizeFontlabel);
legend([h1, h2, h3], {'|u_i^T w|', '|u_i^T w| / \sigma_i', '\sigma_i'}, 'Location', 'northeast');
```

Script utilizado para gerar 32 e 37.

```
clear all
close all
clc

%Definicao do sinal de unidimensional u

n = 200;
u = zeros(1,n);
u(39)= 1/4;
u(40) =1/2;
u(41)= 1/4;
u(100) = 1/2;
u(160) = 1;

T = zeros(n,n);
```

```

%Definindo a matriz Toeplitz de perturbacao
T = zeros(n,n);
for j = 1:n
    for k = 1:n
        T(j,k) = (1/5.6)*exp(-1/10*(j-k)^2);
    endfor
endfor
[U, S, V] = svd(T);
filt = [];
par_Trunc = 50; # declarar parametro de truncamento TSVD

for i = 1:n
    if i < par_Trunc
        f_i = 1;
    else
        f_i = 0;
    endif

    filt = [filt f_i];
endfor

Filt = diag(filt);
A = V*Filt*(pinv(S))*U';

%adicionando ruido
v = T*u';
ruido = rand(n,1);
ruido_normalizado = ruido/norm(ruido);
erro = ruido_normalizado*norm(u)*10^(-3);
w = v + erro;

u_ruido = T\w;

condicao = cond(A)
v2 = A*w; %utilizando pseudoinversa e efeito de ruido + deblurring
v1 = A*u'; %com perturbacao da matriz de toeplitz (efeito blurring)

residuo_ruido= norm(T*v2-v')
residuo_sem_ruido=norm(T*v1-v')

```

```
norm(v2)
erro = norm(u' - v2)
i = 1:n;
plot(i,u,'r','LineWidth',1.2);
hold on
plot(i,v2,'b','LineWidth',1.2);
hold off
SizeFontlabel = 12;
xlabel('i','fontsize',SizeFontlabel);
legend('Sinal original u', 'Sinal restaurado  $u_k = T^{+}_{k} w$ ','FontSize', 10);

clear all
close all
clc

% script para avaliar o erro de regularizacao e perturbacao associado
% a regularizacao TSVD

%Definicao do sinal de unidimensional u

n = 200;
u = zeros(1,n);
u(39)= 1/4;
u(40) =1/2;
u(41)= 1/4;
u(100) = 1/2;
u(160) = 1;

T = zeros(n,n);
%Definindo a matriz Toeplitz de perturbacao
T = zeros(n,n);
for j = 1:n
    for k = 1:n
        T(j,k) = (1/5.6)*exp(-1/10*(j-k)^2);
    endfor
endfor
```

```

endfor

[U, S, V] = svd(T);
Diag_S = diag(S);

%adicionando ruido
v = T*u';
ruido = randn(n,1);
ruido_normalizado = ruido/norm(ruido);
erro = ruido_normalizado*norm(u)*10^(-3);
w = v + erro;

error = [];
bias = [];
pert = [];

x_k = zeros(1,n)';
x_k_ruido = zeros(1,n)'

for k = 1: n
    x_ruido = zeros(1,n)';
    x_i = zeros(1,n)';
    for i = 1: k
        i
        Diag_S(i)

        x_i = x_i + ((U(:,i)'*u')/Diag_S(i))*V(:,i);
        x_ruido = x_ruido + ((U(:,i)'*w)/Diag_S(i))*V(:,i);
    endfor
    %x_k = [x_k x_i];
    %x_k_ruido = [x_k_ruido x_ruido];

    error = [error norm(u' - x_ruido)];
    bias = [bias norm(u' - x_i)];
    pert = [pert norm(x_i - x_ruido)];
endfor

```

```
endfor

[e,i] = min(error)
[reg, i] = min(bias)
[pb, i] = min(pert)
pert(34)
i = 1:n;

plot(i, error, '--k', 'LineWidth', 2.5);
hold on
plot(i, bias, '-b', 'LineWidth', 1.0);
plot(i, pert, '-r', 'LineWidth', 1.0);
hold off
SizeFontlabel=10;
ylim ([10^-3 10^1]);
xlabel('i','fontsize',SizeFontlabel);
legend('||t - x_k||', '||\Delta x_{reg}||', '||\Delta x_{pert}||', 'FontSize', 10)
```

Script utilizado para gerar: 33, 34, 35, 36, 38.

```
clear all
close all
clc

Vet_Aprox_x = [];
teste = [];
Vet_Resid = [];
n = 200;
erro = [];
u = zeros(1,n);
u(39)= 1/4;
u(40) =1/2;
u(41)= 1/4;
u(100) = 1/2;
u(160) = 1;

T = zeros(n,n);
```

```

%Definindo a matriz Toeplitz de perturbacao
T = zeros(n,n);
for j = 1:n
    for k = 1:n
        T(j,k) = (1/5.6)*exp(-1/10*(j-k)^2);
    endfor
endfor

Lambda = logspace(-10,0,n); %para o metodo de Tikhonov

%curva L pelo mEtodo TSVD
for i = 1:n
    [x_i, res_i, E] = BLURTSVDrev(n,T,u,i);
    Vet_Aprox_x = [Vet_Aprox_x x_i];
    Vet_Resid = [Vet_Resid res_i];
    norma_erro = E;
    erro = [erro E];
endfor
i_valores = 1:n;

%curva L pelo mEtodo Tikhonov

for i = 1:n
    [x_i, res_i] = blurTikhonov(n,Lambda(i),T,u);
    Vet_Aprox_x = [Vet_Aprox_x x_i];
    Vet_Resid = [Vet_Resid res_i];
    norma_erro = norm(u' - x_i);
    erro = [erro norma_erro];
endfor
i_valores = 1:length(Lambda);

semilogy(i_valores, erro, '*', 'MarkerSize', 5)
%ylim ([10^-3 10^3]);
xlabel('i', 'fontSize', 10);
legend('||u - u_k||', 'FontSize', 10)

```

```

%Plot curva L
%comentar os trechos 'curva L metodo Tikhonov' ou 'curva L metodo TSVD'
%dependendo da curva L plotada

X = [];
Y = [];
for i = 1:n
    Norma_x = norm(Vet_Aprox_x(:,i));
    Norma_Resid = norm(Vet_Resid(:,i));
    X = [X Norma_x];
    Y = [Y Norma_Resid];
endfor
Lambda(135)
loglog(Y ,X, 'r*',0.3);
hold on
loglog(Y(135) ,X(135), 'b*',0.3);
hold off
SizeFontlabel = 12;
%xlim ([10^-4 10^0]);
%ylim ([10^-1 10^6]);
%legend('Curva - L - Regularizacao TSVD');
legend('Curva - L - Regularizacao Tikhonov');
xlabel('||u_i||', 'fontsize', SizeFontlabel);
ylabel('||Tu_i - w||', 'fontsize', SizeFontlabel);

%plot de solucoes pelo metodo de Tikhonov e TSVD

testes = 178 % indicar parametro de truncamento ou indice do lambda
i_valores = 1:length(Lambda); % comentar se for plotar restauracao por TSVD
[x, res] = blurTikhonov(n, Lambda(testes),T,u); % comentar se for plotar restauracao por TSVD
%i_valores = 1:n; % comentar se for plotar restauracao por Tikhonov
%[x, res] = BLURTSVDrev(n,T,u,testes) % comentar se for plotar restauracao por Tikhonov
display('Solucao x =')

```

```

x
display('Residuo: ')
res
display('Lambda Minimo: ')
Lambda(testes)
display('Erro da solucao')
erro = norm(u'- x)
plot(i_valores,u,'r','LineWidth',1.0);
hold on
plot(i_valores,x,'b','LineWidth',1.0);
hold off
SizeFontlabel = 12;
ylim ([-0.1 1]);
ylabel('x','fontsize',SizeFontlabel);
legend('Sinal original u', 'Sinal restaurado u_\lambda = T^{+}_{\lambda}w','FontSize', 10);

```

```

function [v2, residuo_ruido, erro] = BLURTSVDrev(n, T, u, par_Trunc)
%funcao da regularizacao TSVD
% adaptados as saidas para determinar parametro de truncamento atraves da curva-L

[U, S, V] = svd(T);
filt = [];

for i = 1:n
    if i < par_Trunc
        f_i = 1;
    else
        f_i = 0;
    endif

    filt = [filt f_i];
endfor

Filt = diag(filt);
A = V*Filt*(pinv(S))*U';

%adicionando ruido

```

```
v = T*u';
ruído = rand(n,1);
ruído_normalizado = ruído/norm(ruído);
erro = ruído_normalizado*norm(u)*10(-3);
w = v + erro;

u_ruído = T\w;

condicao = cond(A);
v2 = A*w; %utilizando pseudo inversa e efeito de ruído + deblurring
v1 = A*u'; %com perturbação da matriz de toeplitz (efeito blurring)

resíduo_ruído= norm(T*v2-w);
resíduo_sem_ruído=norm(T*v1-w);
norm(v2);
erro = norm(u' - v2);

endfunction
```

Scripts utilizados para gerar: [39](#) e [40](#).

```
##PRINCIPIO DA DISCREPANCIA DE MOROZOV
clear all
close all
clc

Lambda = logspace(-10, 0,200);
n = 200;
Vet_Resid = [];
nu_dp = (10(-3))*sqrt(n) # valor deveria ser entre 3 e 5
resíduo_sol = [];

u = zeros(1,n);
u(39)= 1/4;
u(40) =1/2;
u(41)= 1/4;
u(100) = 1/2;
u(160) = 1;
```

```

norma_u = norm(u);

T = zeros(n,n);
%Definindo a matriz Toeplitz de perturbacao
T = zeros(n,n);
for j = 1:n
    for k = 1:n
        T(j,k) = (1/5.6)*exp(-1/10*(j-k)^2);
    endfor
endfor

%Avalia o indice em que o residuo e maior que a discrepancia

for i = 1:length(Lambda)
    [x_i, res_i] = blurTikhonov(n, Lambda(i),T,u);
    residuo_sol = [residuo_sol res_i];

    if res_i >= (nu_dp*norma_u)
        display("Maior")
        i
        (res_i)
    else
        display("Menor")
        i
        (res_i)
    endif
endfor

% plot grafico do metodo de discrepancia

buscando = nu_dp*norma_u
i = 1:length(Lambda)
Discrepancia = nu_dp*norma_u*ones(1,200);
semilogy(i, residuo_sol, '-o', 'MarkerSize', 3, 'LineWidth', 1.0);
hold on
semilogy(i, Discrepancia, '--', 'MarkerSize', 3, 'LineWidth', 1.0);
hold off
%xlim ([0 100]);

```

```

%ylim ([10^-3 10^1]);
SizeFontlabel = 12;
xlabel('i','fontsize',SizeFontlabel);
legend('||Tu_\lambda - w||', '||e||','fontsize',12);

function [v2, residuo_ruido, w, u, erro] = blurTikhonov(n, Lambda,T,u)
%Definicao do sinal de unidimensional u

[U, S, V] = svd(T);
%adicionando ruido
v = T*u';
ruido = randn(n,1);
ruido_normalizado = ruido/norm(ruido);
erro = ruido_normalizado*norm(u)*10^(-3);
w = v + erro;

Diag_S = diag(S);
filt = [];

for i = 1:n
    f_i = ((Diag_S(i))^2)/((Diag_S(i))^2+(Lambda^2));
    filt = [filt f_i];
endfor

Filt = diag(filt);
A = V*Filt*(pinv(S))*U';

cond(T);
cond(A);
u_ruido = T\w;
v2 = A*w; %utilizando pseudoinversa e efeito de ruido + deblurring
v1 = A*u'; %com perturbacao da matriz de toeplitz (efeito blurring)
norma_v2 = norm(v2);

residuo_ruido= norm(T*v2 - w);
residuo_sem_ruido=norm(T*v1 - w);

endfunction

```