

UNIVERSIDADE FEDERAL DE ALFENAS

MARIA CARULINE BAQUIÃO

**ESPAÇOS DE REPRODUÇÃO E APROXIMAÇÃO DE SOLUÇÕES DE EQUAÇÕES
FUNCIONAIS**

Alfenas/MG

2019

MARIA CARULINE BAQUIÃO

**ESPAÇOS DE REPRODUÇÃO E APROXIMAÇÃO DE SOLUÇÕES DE EQUAÇÕES
FUNCIONAIS**

Dissertação apresentada ao Programa de Pós-Graduação em Estatística Aplicada e Biometria, área de concentração em Estatística Aplicada e Biometria da Universidade Federal de Alfenas, MG, como parte dos requisitos para a obtenção do título de Mestre. Linha de Pesquisa: Matemática Aplicada e Modelagem Matemática.
Orientador: Prof. Dr. José Claudinei Ferreira.

Alfenas/MG

2019

Dados Internacionais de Catalogação-na-Publicação (CIP)
Sistema de Bibliotecas da Universidade Federal de Alfenas

Baquião, Maria Caruline.
B222e Espaços de reprodução e aproximação de soluções de equações
funcionais / Maria Caruline Baquião – Alfenas/MG, 2019.
95 f.: il. --

Orientador: José Claudinei Ferreira.
Dissertação (Mestrado em Estatística Aplicada e Biometria) –
Universidade Federal de Alfenas, 2019.
Bibliografia.

1. Teoria da aproximação. 2. Equações diferenciais funcionais. 3.
Equações Integrais. I. Ferreira, José Claudinei. II. Título.

CDD- 515.35



MINISTÉRIO DA EDUCAÇÃO
Universidade Federal de Alfenas / UNIFAL-MG
Programa de Pós-graduação em Estatística Aplicada e Biometria

Rua Gabriel Monteiro da Silva, 700. Alfenas - MG CEP 37130-001

Fone: (35) 3701-9600

<https://www.unifal-mg.edu.br/ppgeab/>



MARIA CARULINE BAQUIÃO

“ESPAÇOS DE REPRODUÇÃO E APROXIMAÇÃO DE SOLUÇÕES DE EQUAÇÕES
FUNCIONAIS”

A Banca Examinadora, abaixo assinada, aprova a
Dissertação apresentada como parte dos requisitos para
a obtenção do título de Mestre em Estatística Aplicada
e Biometria pela Universidade Federal de Alfenas.
Área de Concentração: Estatística Aplicada e
Biometria

Aprovado em: 7 de março de 2019.

Prof. Dr. José Claudinei Ferreira

Instituição: UNIFAL-MG

Assinatura: 

Prof. Dr. José Paulo Carvalho dos Santos

Instituição: UNIFAL-MG

Assinatura: 

Prof. Dr. Nelson Henrique Teixeira Lemes

Instituição: UNIFAL-MG

Assinatura: 

AGRADECIMENTOS

Agradeço primeiramente à Deus por ter iluminado meus caminhos sempre.

À minha família, meus pais, Elógio e Naide, e meu irmão Lucas, bases da minha vida, por todo apoio e incentivo que sempre me deram. Além disso, reconheço e agradeço por todo o sacrifício que sei que fizeram, mesmo que em segredo, para que eu pudesse chegar onde cheguei. Agradeço aos meus colegas de turma, que fizeram meus dias mais iluminados e divertidos.

Agradeço à todos os professores do Departamento de Matemática e do Departamento de Estatística da UNIFAL-MG, que desde o curso de Licenciatura em Matemática participaram e colaboraram em minha vida acadêmica, em especial ao Prof. Dr. Claudinei, pela orientação neste trabalho, compreendendo cada uma das minhas dificuldades e tendo toda a paciência do mundo para me explicar várias vezes a mesma coisa quando necessário.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

RESUMO

O objetivo deste trabalho é apresentar, implementar e propor métodos numéricos com espaços de Hilbert de reprodução, aplicando-os a equações funcionais como equações diferenciais e integrais. Apresentamos uma breve introdução sobre núcleos positivos definidos, espaços de Hilbert de reprodução e sobre alguns métodos de projeção. Tratamos separadamente dos métodos numéricos com espaços de Hilbert de reprodução aplicados a equações funcionais lineares e não lineares. Fornecemos uma nova proposta aos métodos de espaços de reprodução a qual dispensa o uso do produto interno do espaço. As simulações numéricas geraram bons resultados e os códigos implementados em linguagem R estão disponíveis no apêndice do trabalho. Por fim, discorreremos sobre os resultados obtidos.

Palavras-chave: Teoria da aproximação. Equações diferenciais funcionais. Equações Integrais.

ABSTRACT

The objective of this work is to present, implement and propose numerical reproducing kernel methods to linear and non-linear functional equations, like differential and integral ones. We also have presented a brief introduction to positive definite kernels and reproducing kernel Hilbert spaces theory, and some projection methods to functional equations. We provide a new proposal for the methods of reproduction spaces which dispenses the use of the inner product of the space. Numerical simulations generated good results and the implemented R-language codes are available at the appendix of this work. Finally, we discuss the results obtained.

Keywords: Theory of approximation. Functional differential equations. Integral Equations.

LISTA DE FIGURAS

Figura 5.1 – Solução aproximada obtida e erro cometido na aproximação.	60
Figura 5.2 – Solução aproximada e comportamento do erro.	61
Figura 5.3 – Solução aproximada e comportamento do erro.	62
Figura 5.4 – Solução aproximada obtida e erro cometido na aproximação.	63
Figura 5.5 – Solução aproximada obtida e erro cometido na aproximação.	64
Figura 5.6 – Estimativas dos coeficientes de Fourier e erro cometido.	64
Figura 5.7 – Solução aproximada obtida e erro cometido.	66
Figura 5.8 – Solução aproximada obtida e erro cometido na aproximação.	67
Figura 5.9 – Comparação dos coeficientes de Fourier e erro.	67
Figura 5.10 – Solução aproximada obtida e erro cometido na aproximação.	68
Figura 5.11 – Solução aproximada obtida e erro cometido na aproximação.	69
Figura 5.12 – Soluções baseadas no Corolário 4.22 (a) e no Corolário 4.24 (b).	70
Figura 5.13 – Erros dos métodos baseados no Corolário 4.22 (a) e no Corolário 4.24 (b).	70
Figura 6.1 – Aproximação da derivada.	77
Figura 6.2 – Ilustração para a regra de três pontos.	78
Figura 0.1 – Divisão da área sob a curva em trapézios.	81

SUMÁRIO

1	INTRODUÇÃO	8
2	CONCEITOS PRELIMINARES	10
2.1	ESPAÇOS COM PRODUTO INTERNO E ORTOGONALIDADE	10
2.2	BASES ORTONORMAIS	15
2.3	PROCESSO DE ORTOGONALIZAÇÃO DE GRAM-SCHMIDT	19
2.3.1	Decomposição QR	21
3	NÚCLEOS E ESPAÇOS DE HILBERT DE REPRODUÇÃO	25
3.1	NÚCLEOS POSITIVOS DEFINIDOS OU DE REPRODUÇÃO	25
3.2	ESPAÇOS DE HILBERT DE REPRODUÇÃO	31
3.3	EXEMPLOS	35
4	APROXIMAÇÃO POR MÉTODOS DE PROJEÇÃO	38
4.1	MÉTODO DE COLOCAÇÃO	39
4.2	MÉTODO DE ESPAÇOS DE HILBERT DE REPRODUÇÃO	40
4.2.1	Método de Espaços de Hilbert de Reprodução para Equações Não Lineares	45
4.2.1.1	Aproximação sem Ortogonalização	51
4.2.2	Método de Espaços de Hilbert de Reprodução para Equações Lineares	56
5	SIMULAÇÕES NUMÉRICAS	59
5.1	MÉTODO DE NYSTROM	60
5.2	MÉTODO DE COLOCAÇÃO	60
5.3	MÉTODO DE ESPAÇOS DE HILBERT DE REPRODUÇÃO	62
5.3.1	Equações Não Lineares	62
5.3.2	Equações Lineares	66
6	RESULTADOS E DISCUSSÕES	71
	REFERÊNCIAS	75
	APÊNDICE	76

1 INTRODUÇÃO

A modelagem matemática é a arte de transformar problemas do mundo real em problemas matemáticos e resolvê-los, geralmente com o uso da computação e da tecnologia, interpretando suas soluções na linguagem usada na realidade [1]. O uso da modelagem apresenta vantagens e avanços obtidos em diversas áreas como Física, Química, Biologia, Astrofísica, dentre outras. A modelagem é eficiente a partir do momento em que há a consciência de que se trabalha com aproximações da realidade. Além disso, é um processo dinâmico usado na obtenção e validação de modelos matemáticos. Um modelo matemático, por sua vez, é um conjunto de símbolos ou relações matemáticas que representam, de alguma forma, o objeto ou fenômeno estudado [1].

A maioria dos modelos são formados por equações diferenciais, integrais, integro diferenciais, dentre outras. De modo geral, as soluções de tais tipos de equações não podem ser obtidas por métodos analíticos [2]. Dessa forma, diversos autores propõem métodos numéricos para encontrar soluções aproximadas de tais equações por meio de alguma linguagem de programação, porém, estes estudos são, na maioria das vezes, desenvolvidos para um tipo específico de equação e nem sempre são precisos.

Um método que tem se mostrado eficiente é o método de espaços de Hilbert de reprodução. Em [3] é utilizado este método para encontrar uma aproximação para a solução analítica de equações integrais não lineares, ao passo que [2] o aplica em equações integro diferenciais não lineares de Fredholm-Volterra. Além disso, [4] faz uso do método na teoria de equações integrais de Volterra, em [5] o usa em problemas de valores de contorno. Em [6] o método é aplicado o método em equações diferenciais de Riccati. [7] faz uma aplicação do método na área financeira. Segundo [8], diversas pesquisas têm sido desenvolvidas aplicando o método de espaços de Hilbert de reprodução em problemas envolvendo equações diferenciais fracionárias e, tal método tem se mostrado eficiente. Nessa mesma linha, [9] utiliza o método para aproximar as soluções de equações diferenciais de Riccati, porém de ordem fracionária.

Nesse sentido, o objetivo deste trabalho é estudar e implementar o método de espaços de Hilbert de reprodução, aplicando-o a quaisquer equações funcionais que possam ser escritas como

$$L(f)(x) = G(x, f), \quad (1.1)$$

em que $x \in X \subset \mathbb{R}$, $f \in \mathcal{H}_K$, $G(\cdot, f) \in \mathcal{H}_K$, $L : \mathcal{H}_R \longrightarrow \mathcal{H}_K$ é um operador linear limitado, \mathcal{H}_R e \mathcal{H}_K são espaços de Hilbert de reprodução com núcleos de reprodução R e K , respectivamente.

Dessa forma, antes de estudarmos a teoria de espaços de Hilbert de reprodução serão apresentados alguns conceitos preliminares, como espaços normados, espaços com produto interno, ortogonalidade, bases, processo de ortogonalização de Gram-Schmidt e decomposição QR. Tais conceitos estão dispostos no segundo capítulo deste trabalho.

No terceiro capítulo fazemos um estudo sobre espaços de Hilbert de reprodução e também sobre núcleos de reprodução. A teoria tratada servirá de base para o estudo do método de resolução de interesse que será abordado mais adiante.

O quarto capítulo abrange os métodos de colocação e de espaços de Hilbert de reprodução, os quais são os métodos de projeção que utilizamos para aproximar a solução de equações que podem ser escritas como a Equação (1.1).

No quinto capítulo estão dispostas as simulações numéricas que foram feitas com base na teoria estudada nos capítulos anteriores.

No sexto capítulo discutimos os resultados obtidos.

Por fim, no Apêndice estão os códigos dos algoritmos utilizados durante o desenvolvimento do trabalho. Implementamos todos os algoritmos em linguagem R [10]. Os cálculos necessários para os métodos implementados, como derivadas, integrais, produto interno, decomposição QR com a ortogonalização de Gram-Schmidt modificada, dentre outros, foram implementados. O único pacote pronto utilizado foi o `pracma`, que através do comando `pinv` calcula a inversa generalizada de matrizes por meio da decomposição S.V.D. e a implementação deste tipo de cálculo é um projeto mais abrangente do que o escopo deste trabalho. Optamos por não fazer uso de pacotes prontos do software, exceto o `pracma`, com o objetivo de contribuir com a formação da autora, ampliando suas possibilidades do uso de tecnologias e em análise numérica ao exercer sua profissão.

2 CONCEITOS PRELIMINARES

Como trataremos de espaços de Hilbert de reprodução, primeiramente abordaremos alguns conceitos básicos relevantes ao trabalho. São apresentados a definição de espaços com produto interno, incluindo aqui espaços de Hilbert, operadores lineares e resultados de ortogonalidade. Também é feito um estudo sobre bases e bases ortonormais. Por fim, abordamos o processo de ortogonalização de Gram-Schmidt e a decomposição QR. As principais referências utilizadas neste capítulo foram [11–14].

2.1 ESPAÇOS COM PRODUTO INTERNO E ORTOGONALIDADE

Primeiramente, vamos definir espaços com produto interno, espaços normados, operadores lineares e, em seguida espaços de Hilbert. Por fim, trataremos alguns conceitos de ortogonalidade que serão relevantes para a próxima seção.

Definição 2.1. *Seja V um espaço vetorial sobre um corpo \mathbb{K} (\mathbb{R} ou \mathbb{C}). Um produto interno em V é uma função $\langle \cdot, \cdot \rangle : V \times V \rightarrow \mathbb{K}$, tal que*

$$(i) \quad \langle x + \alpha y, z \rangle = \langle x, z \rangle + \alpha \langle y, z \rangle, \text{ para } x, y, z \in V;$$

$$(ii) \quad \langle x, y \rangle = \overline{\langle y, x \rangle}, \text{ para } x, y \in V;$$

$$(iii) \quad \langle x, x \rangle > 0, \text{ se } x \neq 0.$$

Definição 2.2. *Seja V um espaço vetorial sobre um corpo \mathbb{K} (\mathbb{R} ou \mathbb{C}). Uma norma em V é uma aplicação $\|\cdot\|: V \rightarrow [0, \infty)$ que satisfaz, para quaisquer $v, w \in V$, as seguintes propriedades*

$$(i) \quad \|v\| > 0, \text{ se } v \neq 0;$$

$$(ii) \quad \|\alpha v\| = |\alpha| \|v\|, \text{ para } \alpha \in \mathbb{K};$$

$$(iii) \quad \|v + w\| \leq \|v\| + \|w\|.$$

Dizemos que $(V, \|\cdot\|)$ é um espaço vetorial normado.

É possível mostrar que todo espaço V , com produto interno $\langle \cdot, \cdot \rangle$, tem uma norma definida por $\|u\| = \sqrt{\langle u, u \rangle}$, para todo $u \in V$. Esta recebe o nome de norma induzida pelo produto interno (Veja [11] para mais detalhes).

Será apresentada agora a definição de operadores lineares limitados. Esta teoria será útil ao estudarmos os métodos de projeção. Primeiramente, vamos definir um operador linear.

Sejam U e V espaços vetoriais e T um subespaço vetorial de U . Um operador linear é uma aplicação $P : T \rightarrow V$ tal que $P(w + \alpha z) = P(w) + \alpha P(z)$, para todo $w, z \in T$ e $\alpha \in \mathbb{K}$.

Para alguns autores, o nome dado à aplicação P da definição anterior é transformação linear e o termo operador apenas é usado para o caso em que $U = V$. Todavia, aqui utilizaremos o nome operador para ambos os casos, como feito por [14]. Além disso, para o caso em que $V = \mathbb{K}$ é um corpo, P recebe o nome de funcional linear.

Se P for contínuo, então P também é chamado de limitado. Além disso, $B(U, V)$ denotará o conjunto de todos os operadores lineares limitados de U em V e, para o caso em que $U = V$, denotaremos simplesmente por $B(U)$.

Operadores lineares limitados serão úteis em diversos resultados importantes para o método de espaços de Hilbert de reprodução, os quais serão vistos no Capítulo 4. O próximo teorema nos dará condições de verificar quando um operador linear é contínuo e, conseqüentemente, limitado.

Teorema 2.3. *Sejam U e V espaços normados e $P : U \rightarrow V$ um operador linear. As seguintes afirmações são equivalentes*

- (i) $\sup_{\|w\| \leq 1} \|P(w)\| < \infty$, para $w \in U$;
- (ii) existe $c > 0$ de modo que $\|P(w)\| \leq c\|w\|$, para todo $w \in U$;
- (iii) P é uniformemente contínuo;
- (iv) P é contínuo;
- (v) P é contínuo em $w = 0$.

Demonstração. (i) \implies (ii) Vamos tomar $c = \sup_{\|w\| \leq 1} \|P(w)\|$, para todo $w \in U$. Se $w \neq 0$,

então $\left\| P\left(\frac{w}{\|w\|}\right) \right\| \leq c$ e, conseqüentemente, $\|P(w)\| \leq c\|w\|$.

(ii) \implies (iii) Sejam $w, z \in U$, então $\|P(w) - P(z)\| = \|P(w - z)\| \leq c\|w - z\|$.

(iii) \implies (iv) e (iv) \implies (v) são imediatas.

(v) \implies (i) Como P é contínuo em zero, temos que existe $\delta > 0$ de modo que, se $\|w - 0\| = \|w\| \leq \delta$, então $\|P(w) - P(0)\| = \|P(w)\| \leq 1$. Dessa forma, se $\|w\| \leq 1$, segue que $\|\delta w\| \leq \delta$ e $\|P(\delta w)\| \leq 1$. Portanto, $\|P(w)\| \leq \frac{1}{\delta}$, e (i) vale. \square

Definiremos em seguida espaços de Hilbert, pois estes são fundamentais para a teoria estudada neste trabalho. Lembremos que um espaço normado V é completo se, dada $u_n \in V$ tal que $\|u_n - u_m\| \rightarrow 0$, quando $n, m \rightarrow \infty$, implicam que $u_n \rightarrow u \in V$.

Seja V um espaço vetorial normado sobre um corpo \mathbb{K} com produto interno $\langle \cdot, \cdot \rangle$ e $\|u\| = \sqrt{\langle u, u \rangle}$, para $u \in V$. Se $(V, \|\cdot\|)$ for um espaço completo, dizemos que V é um espaço de Hilbert.

Trataremos agora de definições e resultados sobre ortogonalidade, pois estes serão importantes no estudo de bases ortonormais.

Sejam \mathcal{H} um espaço de Hilbert com produto interno $\langle \cdot, \cdot \rangle$ e $W \subset \mathcal{H}$. Então, chamamos o conjunto

$$W^\perp = \{u \in \mathcal{H}; \langle w, u \rangle = 0, \text{ para todo } w \in W\}$$

de conjunto ortogonal a W .

Lema 2.4. *Seja \mathcal{H} um espaço de Hilbert. Se $W \subset \mathcal{H}$, então W^\perp é subespaço fechado de \mathcal{H} .*

Demonstração. Seja (v_n) um sequência de pontos de W^\perp que converge para v . Queremos mostrar que $v \in W^\perp$, ou seja, $\langle w, v \rangle = 0$, para todo $w \in W$.

De fato, seja $w \in W$, então

$$\begin{aligned} |\langle w, v \rangle| - |\langle w, v_n \rangle| &= |\langle w, v - v_n \rangle| \\ &\leq \|w\| \|v - v_n\| \quad (\text{Cauchy-Schwarz}). \end{aligned}$$

Assim,

$$\begin{aligned} \langle w, v \rangle &= \langle w, \lim_{n \rightarrow \infty} v_n \rangle \\ &= \lim_{n \rightarrow \infty} \langle w, v_n \rangle \\ &= 0. \end{aligned}$$

Portanto, $v \in W^\perp$. □

Apresentamos agora um resultado útil para o conceito de base ortogonal e para a demonstração do teorema da representação de Riesz. A sua demonstração usa um argumento técnico envolvendo sequências e pode ser encontrada em [14].

Lema 2.5. *Sejam \mathcal{H} um espaço de Hilbert e E um subespaço fechado de \mathcal{H} . Então,*

$$\mathcal{H} = E \oplus E^\perp.$$

Apresentamos a seguir o Teorema da Representação de Riesz. É um resultado importante na teoria de espaços de Hilbert, fornecendo a forma geral de um funcional linear limitado em um espaço de Hilbert.

Teorema 2.6 (Teorema da Representação de Riesz). *Sejam \mathcal{H} um espaço de Hilbert e $f : \mathcal{H} \rightarrow \mathbb{K}$ um funcional linear contínuo. Existe um único $w \in \mathcal{H}$ tal que $f(u) = \langle u, w \rangle$, para todo $u \in \mathcal{H}$.*

Demonstração. Sejam \mathcal{H} um espaço de Hilbert e $f : \mathcal{H} \rightarrow \mathbb{K}$ um funcional linear contínuo. Se f é identicamente nula, basta tomarmos $w = 0$. Caso contrário, considere o conjunto não vazio $Nuc(f) = \{u \in \mathcal{H}; f(u) = 0\}$. Temos que $Nuc(f)$ é um subespaço fechado de \mathcal{H} . Assim,

$$Nuc(f) \oplus (Nuc(f))^\perp = \mathcal{H}.$$

Seja $r \in (Nuc(f))^\perp$ e $u \in \mathcal{H}$ de modo que $\|r\| = 1$. Definimos $w = f(u)r - f(r)u$, assim

$$\begin{aligned} f(w) &= f(f(u)r - f(r)u) \\ &= f(u)f(r) - f(r)f(u) \\ &= 0. \end{aligned}$$

Logo, $w \in Nuc(f)$.

Por outro lado, note que

$$\begin{aligned} 0 &= \langle w, r \rangle \\ &= \langle f(u)r - f(r)u, r \rangle \\ &= \langle f(u)r, r \rangle - \langle f(r)u, r \rangle \\ &= f(u) \langle r, r \rangle - f(r) \langle u, r \rangle \\ &= f(u) \|r\|^2 - f(r) \langle u, r \rangle \\ &= f(u) - \langle u, \overline{f(r)}r \rangle. \end{aligned}$$

Assim, $f(u) = \langle u, \overline{f(r)}r \rangle$. Tomando $v = \overline{f(r)}r$, temos que $f(u) = \langle u, v \rangle$.

Resta mostrarmos a unicidade. Suponhamos que $v_1, v_2 \in \mathcal{H}$ são tais que

$f(u) = \langle u, v_1 \rangle = \langle u, v_2 \rangle$, para todo $u \in \mathcal{H}$. Assim,

$$\begin{aligned}
 f(v_1 - v_2) &= \langle v_1 - v_2, v_1 \rangle \\
 &= \langle v_1 - v_2, v_2 \rangle \\
 \iff \langle v_1 - v_2, v_1 - v_2 \rangle &= 0 \\
 \iff \|v_1 - v_2\|^2 &= 0 \\
 \iff \|v_1 - v_2\| &= 0 \\
 \iff v_1 &= v_2.
 \end{aligned}$$

□

Um conceito importante para este trabalho é o de operador adjunto. Este conceito será importante no método de espaços de Hilbert de reprodução. A definição de operadores adjuntos decorre do teorema que segue.

Teorema 2.7. *Sejam \mathcal{H}_1 e \mathcal{H}_2 dois espaços de Hilbert. Se $P \in B(\mathcal{H}_1, \mathcal{H}_2)$ então existe um único operador $P^* \in B(\mathcal{H}_2, \mathcal{H}_1)$ de modo que*

$$\langle P(x), y \rangle_{\mathcal{H}_2} = \langle x, P^*(y) \rangle_{\mathcal{H}_1} \quad x \in \mathcal{H}_1, \quad y \in \mathcal{H}_2. \quad (2.1)$$

Daremos aqui apenas uma ideia da demonstração. Para mais detalhes veja em [14].

Seja $P : \mathcal{H}_1 \rightarrow \mathcal{H}_2$ um operador linear limitado. Vamos definir $f_y : \mathcal{H}_1 \rightarrow \mathbb{K}$ de modo que $f_y(x) = \langle P(x), y \rangle_{\mathcal{H}_2}$. Pela linearidade de P e do produto interno temos que f_y é um funcional linear.

Assim, pelo Teorema 2.6 segue que existe um único $w \in \mathcal{H}_1$ tal que

$$f_y(x) = \langle x, w \rangle_{\mathcal{H}_1} = \langle P(x), y \rangle_{\mathcal{H}_2}, \quad \text{para todo } x \in \mathcal{H}_1.$$

Como w é definido de modo único por y , basta definirmos $P^*(y) = w$. Então, por construção, temos que

$$\langle P(x), y \rangle_{\mathcal{H}_2} = \langle x, P^*(y) \rangle_{\mathcal{H}_1}, \quad \text{para todo } x \in \mathcal{H}_1 \text{ e } y \in \mathcal{H}_2.$$

O operador P^* , definido no Teorema 2.7, é chamado de operador adjunto do operador

P .

Observação 2.8. *Note que essa definição não deixa claro como calcular $P^*(y)$.*

2.2 BASES ORTONORMAIS

Nesta seção, primeiramente trataremos de alguns conceitos fundamentais para o estudo de bases de um modo geral, como independência linear. Em seguida, abordaremos alguns conceitos de base e, por fim, falaremos de bases ortonormais.

Sejam V um espaço vetorial sobre um corpo \mathbb{K} e B um subconjunto de V . Dizemos que B é linearmente independente se dados $\alpha_1, \alpha_2, \dots, \alpha_n \in \mathbb{K}$ e $v_1, v_2, \dots, v_n \in B$, a igualdade

$$\alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_n v_n = 0$$

implicar que $\alpha_1 = \alpha_2 = \dots = \alpha_n = 0$.

Caso contrário, B é dito ser linearmente dependente.

Seja V um espaço vetorial sobre um corpo \mathbb{K} . Uma base para V é um conjunto B tal que:

- (i) B é linearmente independente.
- (ii) B é um conjunto gerador de \mathcal{H} , isto é, se $v \in V$, então existem $\beta_1, \dots, \beta_m \in \mathbb{K}$, com $m \in \mathbb{N}$, tais que $v = \sum_{l=1}^m \beta_l v_l$, com $v_l \in B$, para $l = 1, \dots, m$.

Lembramos agora o Lema de Zorn, que é essencial para demonstrar a existência de uma base ortonormal para espaços de Hilbert.

Lema 2.9 (Lema de Zorn). *Seja B um conjunto não vazio parcialmente ordenado. Se todo subconjunto totalmente ordenado de B possui uma cota superior, então existe um maior elemento de B .*

Por meio do Lema de Zorn é possível mostrar que, sendo V um espaço vetorial, sempre existe uma base B de V , esta base é chamada de base de Hamel (Para detalhes da demonstração veja [15]). Pode-se mostrar ainda que B é não enumerável, quando possuir infinitos elementos, por isso, vamos introduzir a partir de agora a ideia de base ortonormal.

Definição 2.10. *Seja \mathcal{H} um espaço de Hilbert sobre um corpo \mathbb{K} . Uma base ortogonal de \mathcal{H} é um conjunto B tal que:*

(i) Se $u, v \in B$, com $u \neq v$, então $u \perp v$. Além disso, se $u \in B$, então $u \neq 0$.

(ii) Dados $v \in \mathcal{H}$ e $\varepsilon > 0$, existem $\alpha_1, \dots, \alpha_m \in \mathbb{K}$, com $m \in \mathbb{N}$, e $v_1, \dots, v_m \in B$ tais que

$$\left\| v - \sum_{i=1}^m \alpha_i v_i \right\| < \varepsilon.$$

Se além das condições (i) e (ii) da Definição 2.10, para todo $u \in B$ tivermos que $\|u\| = 1$, então B é uma base ortonormal de \mathcal{H} .

Teorema 2.11. *Seja \mathcal{H} um espaço de Hilbert não trivial. Então, existe uma base B , ortonormal de \mathcal{H} .*

Demonstração. Sejam \mathcal{H} um espaço de Hilbert e $u \in \mathcal{H}$, com $u \neq 0$. Então, o conjunto $B = \left\{ \frac{u}{\|u\|} \right\}$ é ortonormal.

Seja A o conjunto de todos os subconjuntos ortonormais de \mathcal{H} . Temos que $B \in A$ e que A é parcialmente ordenado pela inclusão, isto é, se $C, D \in A$ então, caso $C \subset D$, temos $C < D$.

Se T é um subconjunto totalmente ordenado, então $M = \bigcup_{C \in T} C$ é o elemento maximal de T . Logo, pelo Lema de Zorn, existe $N \in A$ que é maximal, ou seja, não existe $P \in A$ tal que $P > N$.

Resta mostrar que N é uma base ortonormal de \mathcal{H} . Suponha que N não é uma base ortonormal de \mathcal{H} . Então, $\overline{N} \neq \mathcal{H}$ e existe $v \in N^\perp$, com $\|v\| = 1$, pois $\mathcal{H} = \overline{N} \oplus N^\perp$. Logo, $P = N \cup \{v\} > N$. Absurdo, pois N é maximal.

Portanto, N é uma base ortonormal de \mathcal{H} . □

Corolário 2.12. *Seja \mathcal{H} um espaço de Hilbert. Se $B \subset \mathcal{H}$ é um conjunto ortonormal, então existe uma base ortonormal de \mathcal{H} que contém B .*

Dizemos que um subconjunto D de V é denso em V se $D \cap U \neq \emptyset$, para todo aberto não vazio U de V , ou que $\overline{D} = V$.

Um espaço métrico V é separável se existe um subconjunto $M \subset V$ enumerável e denso em V . Nosso trabalho trata, em sua maioria, desses espaços. Visto que trabalharemos com ortogonalização de bases através do processo de Gram-Schmidt, descrito mais adiante.

O próximo teorema fornece uma maneira de verificar se um espaço é separável. Sua demonstração pode ser encontrada em [14].

Teorema 2.13. *Um espaço normado V é separável se, e somente se, existe um subconjunto M enumerável linearmente independente que gera V .*

Veremos agora a Desigualdade de Bessel. Este lema será importante na demonstração do próximo teorema, que fornece condições para verificar se um conjunto ortonormal é uma base de um espaço de Hilbert.

Lema 2.14 (Desigualdade de Bessel). *Sejam \mathcal{H} um espaço de Hilbert e B um conjunto ortonormal de \mathcal{H} . Se $v \in \mathcal{H}$ então*

$$\sum_{i=1}^n |\langle v, v_i \rangle|^2 \leq \|v\|^2,$$

sempre que $v_1, \dots, v_n \in B$, com $n \in \mathbb{N}$.

Demonstração. Sejam \mathcal{H} um espaço de Hilbert e B um conjunto ortonormal de \mathcal{H} . Tomemos $u = v - \sum_{i=1}^n \langle v, v_i \rangle v_i \in \mathcal{H}$, então,

$$\begin{aligned} \|u\|^2 &= \left\langle v - \sum_{i=1}^n \langle v, v_i \rangle v_i, v - \sum_{i=1}^n \langle v, v_i \rangle v_i \right\rangle \\ &= \|v\|^2 - \left\langle v, v - \sum_{i=1}^n \langle v, v_i \rangle v_i \right\rangle - \left\langle v - \sum_{i=1}^n \langle v, v_i \rangle v_i, v \right\rangle + \sum_{i=1}^n |\langle v, v_i \rangle|^2 \\ &= \|v\|^2 - \sum_{i=1}^n |\langle v, v_i \rangle|^2. \end{aligned}$$

Como $\|u\|^2 \geq 0$, segue que

$$\|v\|^2 - \sum_{i=1}^n |\langle v, v_i \rangle|^2 \geq 0$$

□

Dentre as condições do próximo teorema, temos um resultado importante na teoria de espaços de Hilbert, chamado de identidade de Parseval. Este resultado fornece uma representação de qualquer elemento do espaço em função dos elementos da base ortonormal deste espaço.

Teorema 2.15. *Sejam \mathcal{H} um espaço de Hilbert e B um conjunto ortonormal de \mathcal{H} . Então, B é base de \mathcal{H} se, e somente se,*

(i) $[B]^\perp = \{0\}$.

(ii) (*Identidade de Parseval*). Se $v \in \mathcal{H}$, então

$$v = \sum_{u \in B} \langle v, u \rangle u. \quad (2.2)$$

$$\text{Em particular, } \|v\|^2 = \sum_{u \in B} |\langle v, u \rangle|^2.$$

Demonstração. (\implies) Sejam \mathcal{H} um espaço de Hilbert e B uma base ortonormal de \mathcal{H} . Por definição, segue que $[B]^\perp = \{0\}$. Além disso, se $v \in \mathcal{H}$ e $\varepsilon > 0$, existem $\alpha_1, \alpha_2, \dots, \alpha_n \in \mathbb{K}$ e $v_1, v_2, \dots, v_n \in B$ tais que

$$\left\| v - \sum_{i=1}^n \alpha_i v_i \right\|^2 < \varepsilon.$$

Vamos mostrar que podemos escolher $\alpha_i = \langle v, v_i \rangle$. Pela desigualdade de Bessel, para todo $n \in \mathbb{N}$ e $v_1, v_2, \dots, v_n \in B$, segue que

$$\sum_{i=1}^n |\langle v, v_i \rangle|^2 \leq \|v\|^2.$$

Logo, se B é enumerável, por indução temos que

$$\sum_{i=1}^{\infty} |\langle v, v_i \rangle|^2 \leq \|v\|^2. \quad (2.3)$$

Se B for não enumerável, então $\langle v, u \rangle \neq 0$ exceto para $u \in B_1$, em que $B_1 \subset B$ é enumerável. De fato, seja $C_j = \{u \in B; \langle v, u \rangle \geq \frac{1}{j}\}$ para $j \in \mathbb{N}$. Suponha que C_j é infinito. Então existem $u_1, u_2, \dots, u_n \in C_j$, para todo $n \in \mathbb{N}$. Assim,

$$\sum_{j=1}^n |\langle v, u_j \rangle|^2 \geq \frac{n}{j^2}.$$

Logo,

$$\sum_{j=1}^{\infty} |\langle v, u_j \rangle|^2 \longrightarrow \infty,$$

o que contradiz a Desigualdade (2.3). Portanto, C_j é finito.

Dessa forma, $B_1 = \bigcup_{j=1}^{\infty} C_j$ é enumerável e

$$\sum_{u \in B} |\langle u, v \rangle|^2 = \sum_{u \in B_1} |\langle u, v \rangle|^2 \leq \|v\|^2$$

e a série

$$\begin{aligned} \sum_{u \in B} \langle v, u \rangle u &= \sum_{u \in B_1} \langle v, u \rangle u \\ &= \sum_{i=1}^{\infty} \langle v, v_i \rangle v_i \end{aligned}$$

é convergente. Segue então que a sequência $w_n = v - \sum_{i=1}^n \langle v, v_i \rangle v_i$ converge para 0, pois $0 \leq \|w_n\|^2 = \|v\|^2 - \sum_{i=1}^n |\langle v, v_i \rangle|^2$ e $\langle w_n, v_j \rangle = 0$ para todo $1 \leq j \leq n$. Portanto, (w_n) é convergente e $(\lim w_n) \in B^\perp$. Assim, $v = \sum_{u \in B} \langle v, u \rangle u$.

(\Leftarrow) Suponhamos agora que B é um conjunto ortonormal de \mathcal{H} tal que $v = \sum_{u \in B} \langle v, u \rangle u$, para todo $v \in \mathcal{H}$. Assim, se $v \in B^\perp$ segue que $v = 0$ e que $B^\perp = \overline{[B]}^\perp = \mathcal{H}$. Portanto, B é base de \mathcal{H} . \square

Os coeficientes $\langle v, u \rangle$ dados na Equação (2.2) recebem o nome de coeficientes de Fourier de v .

2.3 PROCESSO DE ORTOGONALIZAÇÃO DE GRAM-SCHMIDT

Nesta seção estudaremos o processo de ortogonalização de Gram-Schmidt. Geralmente é utilizado para construir conjuntos ortonormais a partir de conjuntos conhecidos. Este procedimento será de grande utilidade no método de espaços de Hilbert de reprodução, pois permitirá construir bases ortonormais para tais espaços.

Primeiramente, seja V um espaço vetorial sobre um corpo \mathbb{K} , com produto interno $\langle \cdot, \cdot \rangle$. Partindo de um conjunto B enumerável e linearmente independente em V podemos construir um conjunto linearmente independente e ortonormal S de tal forma que $[B] = [S]$. Este procedimento é feito de maneira construtiva, como será explicado a seguir.

Considere um conjunto $B = \{v_1, v_2, \dots, v_n\} \subseteq V$ linearmente independente. Vamos construir um conjunto $S = \{w_1, w_2, \dots, w_n\} \subseteq V$ ortonormal e tal que $[B] = [S]$, como segue.

Tomemos

$$\begin{aligned}
 w_1 &= v_1 \\
 w_2 &= v_2 - \frac{\langle v_2, w_1 \rangle}{\|w_1\|^2} w_1 \\
 &\vdots \\
 w_k &= v_k - \frac{\langle v_k, w_1 \rangle}{\|w_1\|^2} w_1 - \dots - \frac{\langle v_k, w_{k-1} \rangle}{\|w_{k-1}\|^2} w_{k-1}, \quad 2 \leq k \leq n.
 \end{aligned}$$

Como $\{v_1, v_2\}$ é um conjunto linearmente independente, temos que $w_2 \neq 0$, pois v_2 não pode ser escrito como combinação linear de v_1 . Note ainda que,

$$\begin{aligned}
 \langle w_2, w_1 \rangle &= \left\langle v_2 - \frac{\langle v_2, w_1 \rangle}{\|w_1\|^2} w_1, w_1 \right\rangle \\
 &= \langle v_2, w_1 \rangle - \frac{\langle v_2, w_1 \rangle}{\|w_1\|^2} \langle w_1, w_1 \rangle \\
 &= \langle v_2, w_1 \rangle - \frac{\langle v_2, w_1 \rangle}{\|w_1\|^2} \|w_1\|^2 \\
 &= 0.
 \end{aligned}$$

Portanto, w_2 é ortogonal a w_1 . De forma semelhante observamos que o conjunto $S = \{w_1, w_2, \dots, w_n\}$ da maneira como foi construído é ortogonal. Agora, mostraremos que S é linearmente independente. Sejam $w \in S$ e $\alpha_1, \dots, \alpha_n \in \mathbb{K}$. Assim, podemos escrever:

$$w = \sum_{i=1}^n \alpha_i w_i.$$

Como S é um conjunto ortogonal, temos que $\langle w_i, w_j \rangle = 0$ para todo $i \neq j$, então segue que

$$\begin{aligned}
 \langle w, w_j \rangle &= \left\langle \sum_{i=1}^n \alpha_i w_i, w_j \right\rangle \\
 &= \sum_{i=1}^n \alpha_i \langle w_i, w_j \rangle \\
 &= \alpha_j \langle w_j, w_j \rangle, \quad \text{para } j = 1, \dots, n.
 \end{aligned}$$

Portanto, $\alpha_j = \frac{\langle w, w_j \rangle}{\|w_j\|^2}$, para todo $j = 1, \dots, n$, e $w = \sum_{i=1}^n \frac{\langle w, w_i \rangle}{\|w_i\|^2} w_i$.

Vimos anteriormente que $\alpha_j = \frac{\langle w, w_j \rangle}{\|w_j\|^2}$, para todo $j = 1, 2, \dots, n$. Em particular, para

$w = 0$ temos que

$$\alpha_j = \frac{\langle 0, w_j \rangle}{\|w_j\|^2} = 0, \quad j = 1, 2, \dots, n.$$

Portanto, S é um conjunto linearmente independente. Note ainda que $\dim[B] = \dim[S] = n$ e que w_i é dado por uma combinação linear dos elementos de B , para todo $i = 1, 2, \dots, n$. Assim, $[B] = [S]$.

O processo de ortogonalização de Gram-Schmidt pode ser usado em um conjunto enumerável, como será formalizado no teorema a seguir.

Teorema 2.16. *Seja \mathcal{H} um espaço de Hilbert e $\{v_n\}$ uma coleção enumerável linearmente independente em \mathcal{H} . Existe uma coleção ortonormal $\{w_n\}$ em \mathcal{H} com a mesma cardinalidade de $\{v_n\}$. Além disso, para todo $m \in \mathbb{N}$, temos que $\{v_1, v_2, \dots, v_m\}$ e $\{w_1, w_2, \dots, w_m\}$ geram o mesmo subespaço vetorial.*

Demonstração. Seja $\{v_n\}$ uma coleção enumerável linearmente independente em \mathcal{H} . Vamos construir a sequência $\{w_n\}$ por recorrência.

Defina $w_1 = \frac{v_1}{\|v_1\|}$. Note que $\{v_1\}$ e $\{w_1\}$ geram o mesmo subespaço vetorial. Semelhante ao que foi feito anteriormente, dado um conjunto ortonormal $\{\varphi_1, \dots, \varphi_k\}$, o vetor $\varphi - \sum_{j=1}^k \langle \varphi_j, \varphi \rangle \varphi_j$ é ortogonal a cada φ_j , $1 \leq j \leq k$.

Desta forma, seja $v'_{k+1} = v_k - \sum_{j=1}^k \langle v_j, v_k \rangle v_j$ e $w_{k+1} = \frac{v'_{k+1}}{\|v'_{k+1}\|}$. Segue que pela construção, o conjunto $\{w_1, w_2, \dots, w_{k+1}\}$ é ortonormal e gera o mesmo subespaço que $\{v_1, v_2, \dots, v_{k+1}\}$. \square

2.3.1 Decomposição QR

Agora estudaremos a decomposição QR, conforme [11, 16]. Para decomposição QR em dimensão infinita, sugerimos ver [17].

Dada uma matriz $A_{m \times n}$ de posto n , então podemos decompor tal matriz em um produto das matrizes Q e R , em que Q é possui colunas ortonormais e R é triangular superior. Esse método será formalizado no Teorema 2.17. Para tal procedimento, utilizamos o processo de ortogonalização de Gram-Schmidt, porém existem outras formas como rotação de Householder, por exemplo.

Teorema 2.17. *Seja a matriz $A_{m \times n}$ de posto n . Então,*

$$A = QR$$

em que Q é uma matriz $m \times n$ com colunas ortonormais e R é uma matriz $n \times n$ triangular superior com elementos diagonais positivos.

Demonstração. Sejam $A_{m \times n}$ uma matriz de posto n e v_1, v_2, \dots, v_n as colunas da matriz A . Como A tem posto n , temos que o conjunto de vetores $\{v_i\}_{i=1}^n$ é linearmente independente em \mathbb{K}^m . Desta forma, aplicando o processo de ortogonalização de Gram-Schmidt, descrito na Seção 2.3, obtemos o conjunto de vetores ortonormais $\{q_i\}_{i=1}^n \subset \mathbb{K}^m$, dados por

$$q_j = \frac{1}{r_{jj}} \left(v_j - \sum_{i=1}^{j-1} r_{ij} q_i \right), \quad j = 1, 2, \dots, n,$$

em que $r_{ij} = \langle v_j, q_i \rangle$, para todo $i = 1, 2, \dots, j-1$ e $r_{jj} = \left\| v_j - \sum_{i=1}^{j-1} r_{ij} q_i \right\|$. Assim,

$$\begin{aligned} v_1 &= r_{11} q_1 \\ v_2 &= r_{12} q_1 + r_{22} q_2 \\ &\vdots \\ v_n &= r_{1n} q_1 + \dots + r_{nn} q_n. \end{aligned} \tag{2.4}$$

Tomemos Q como a matriz que tem como colunas os vetores q_1, q_2, \dots, q_n e R como a matriz triangular superior dada por

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{nn} \end{bmatrix}.$$

Dessa forma, obtemos que

$$\begin{aligned}
 QR &= \begin{bmatrix} q_1 & q_2 & \cdots & q_n \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{21} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{nn} \end{bmatrix} \\
 &= \begin{bmatrix} q_1 r_{11} & (q_1 r_{12} + q_2 r_{21}) & \cdots & (q_1 r_{1n} + \cdots + q_n r_{nn}) \end{bmatrix} \\
 &= \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} \\
 &= A.
 \end{aligned}$$

Portanto, $A = QR$. □

Corolário 2.18. *Seja a matriz $A_{m \times n}$ de posto $p \leq n$. Então, podemos escrever $A = QR$ em que Q é uma matriz $m \times p$ com colunas ortonormais, tal que $Q^T Q = Id_{p \times p}$, e R é uma matriz $p \times n$ triangular superior.*

Para mais detalhes veja [11]. Vejamos um exemplo.

Exemplo 2.19. *Considere a matriz A a seguir:*

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 2 \\ 1 & 0 & -1 \\ 0 & 1 & 3 \end{bmatrix}.$$

Utilizando o Algoritmo 5, dado no Apêndice, ou Gram-Schmidt, obtemos

$$Q = \begin{bmatrix} 0.7071068 & 0.3162278 & -0.5070926 \\ 0.0000000 & 0.6324555 & -0.1690309 \\ 0.7071068 & -0.3162278 & 0.5070926 \\ 0.0000000 & 0.6324555 & 0.6761234 \end{bmatrix} \text{ e } R = \begin{bmatrix} 1.414214 & 0.7071068 & -0.7071068 \\ 0.000000 & 1.5811388 & 3.4785054 \\ 0.000000 & 0.000000 & 1.1832160 \end{bmatrix}.$$

Para calcular o erro cometido pelo arredondamento em ponto flutuante, basta calcular-

mos $A = QR$. Assim, obtemos

$$A - QR = \begin{bmatrix} 0 & 0 & -1.110223 \cdot 10^{-16} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Logo, o erro cometido pelo algoritmo é da ordem de 10^{-16} .

Podemos usar este mesmo método para funções, utilizando o mesmo procedimento. Porém, o produto escalar e a norma usados no caso da matriz devem ser trocados pelo produto interno e pela norma do espaço a ser tratado. O Algoritmo 6 está adaptado para o Exemplo 2.20, dado a seguir.

Exemplo 2.20. O conjunto $W_2^1[a, b]$ formado por funções $u : [a, b] \rightarrow \mathbb{R}$ contínuas, tais que:

$$(i) \quad u(x) = u(a) + \int_a^x u'(s) ds \text{ e}$$

$$(ii) \quad \int_a^b |u'(x)|^2 dx < \infty,$$

com produto interno definido por

$$\langle u, v \rangle = u(a)v(a) + \int_a^b u'(s)v'(s) ds \quad u, v \in W_2^1[a, b],$$

é um espaço de Hilbert (de reprodução, como veremos mais adiante, e possui núcleo R dado por $R_x(t) = 1 - a + \min(t, x)$).

Tomando $a = 0$ e $b = 1$, trabalhamos com o espaço $W_2^1[0, 1]$, que é um caso particular do espaço descrito no Exemplo 2.20. Tomando $x_1 = 0, x_2 = 0.1, \dots, x_{21} = 1$ e $\varphi_1(x) = R(x, x_1), \varphi_2(x) = R(x, x_2), \dots, \varphi_{21}(x) = R(x, x_{21})$, pode-se mostrar que $A = \{\varphi_1, \varphi_2, \dots, \varphi_{21}\}$ é um conjunto linearmente independente em $W_2^1[0, 1]$.

Pelo que vimos, podemos ortogonalizar os elementos de A e obter $Q = \{\Psi_1, \Psi_2, \dots, \Psi_{21}\}$ em que A e Q se comportam como matrizes e

$$A = QR.$$

Os cálculos foram feitos usando a discretização do Algoritmo 6 e o erro foi da ordem de 10^{-16} .

3 NÚCLEOS E ESPAÇOS DE HILBERT DE REPRODUÇÃO

Antes de estudar o método de espaços de Hilbert de reprodução, trataremos da teoria de espaços de Hilbert de reprodução. Esta teoria está ligada aos núcleos positivos definidos ou núcleos de reprodução, como veremos no decorrer do capítulo, motivando o estudo deste conteúdo.

Primeiramente estudaremos alguns resultados sobre núcleos positivos definidos. Em seguida, passaremos pela teoria de espaços de Hilbert de reprodução. Por fim, serão apresentados alguns exemplos de espaços de Hilbert de reprodução relevantes para o nosso trabalho. As principais referências utilizadas foram [4, 14, 15, 18].

3.1 NÚCLEOS POSITIVOS DEFINIDOS OU DE REPRODUÇÃO

Nesta seção discutiremos algumas definições e resultados relacionados aos núcleos positivos definidos.

Seja X um conjunto não vazio. Um núcleo R é uma função da forma

$$\begin{aligned} R : X \times X &\longrightarrow \mathbb{C} \\ (x, y) &\longmapsto R(x, y). \end{aligned}$$

Um tipo particular de núcleo, denominado núcleo positivo definido será importante para nosso estudo. Para tal, é necessário antes conhecer a definição de matriz não negativa definida.

Uma matriz $A_{n \times n}(\mathbb{C})$ é não negativa definida quando a forma quadrática

$$B(y) = \bar{y}Ay^t, \quad y \in \mathbb{C}^n,$$

é não negativa, isto é,

$$\bar{y}Ay^t \geq 0, \quad y \in \mathbb{C}^n.$$

Além disso, dizemos que a matriz A é positiva definida se a desigualdade anterior é estrita, para y não nulo.

Definição 3.1. *Seja X um conjunto não vazio. Um núcleo $R : X \times X \longrightarrow \mathbb{C}$ é positivo definido quando a matriz $A = [R(x_i, x_j)]_{n \times n}$ é não negativa definida, para qualquer $n \geq 1$ e qualquer n -upla $(x_1, x_2, \dots, x_n) \in X^n$.*

Denotaremos o conjunto dos núcleos positivos definidos com domínio $X \times X$ por $PD(X)$.

Observe que satisfazer a Definição 3.1 é equivalente a satisfazer a seguinte desigualdade

$$\sum_{i,j=1}^n \bar{c}_i c_j R(x_i, x_j) \geq 0. \quad (3.1)$$

Se a desigualdade dada na Equação (3.1) for estrita, sempre que algum c_i for não nulo e os x'_i s forem distintos, dizemos que R é um núcleo estritamente positivo definido.

De fato,

$$\begin{aligned} [\bar{c}][R(x_i, x_j)][c]^t &= \begin{bmatrix} \bar{c}_1 & \bar{c}_2 & \dots & \bar{c}_n \end{bmatrix} \begin{bmatrix} R(x_1, x_1) & \dots & R(x_1, x_n) \\ R(x_2, x_1) & \dots & R(x_2, x_n) \\ \vdots & \ddots & \vdots \\ R(x_n, x_1) & \dots & R(x_n, x_n) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} \\ &= \sum_{j=1}^n \bar{c}_1 c_j R(x_1, x_j) + \dots + \sum_{j=1}^n \bar{c}_n c_j R(x_n, x_j) \\ &= \sum_{i=1}^n \sum_{j=1}^n \bar{c}_i c_j R(x_i, x_j) \\ &= \sum_{i,j=1}^n \bar{c}_i c_j R(x_i, x_j). \end{aligned}$$

para qualquer $n \geq 1$ e qualquer n-upla $(x_1, \dots, x_n) \in X^n$ e $(c_1, \dots, c_n) \in \mathbb{C}^n$.

Em seguida, veremos alguns exemplos de núcleos positivos definidos.

Exemplo 3.2. *Sejam X um conjunto qualquer não vazio e $f : X \rightarrow \mathbb{C}$ uma função qualquer, o núcleo $R : X \times X \rightarrow \mathbb{C}$ dado por $R(x, y) = f(x)\overline{f(y)}$, para todo $x, y \in X$, é positivo definido.*

De fato, sejam $n \in \mathbb{N}$, $(x_1, x_2, \dots, x_n) \in X^n$ e $(c_1, c_2, \dots, c_n) \in \mathbb{C}^n$, então

$$\begin{aligned}
\sum_{i,j=1}^n \bar{c}_i c_j R(x_i, x_j) &= \sum_{i,j=1}^n \bar{c}_i c_j f(x_i) \overline{f(x_j)} \\
&= \sum_{i,j=1}^n \bar{c}_i f(x_i) \overline{c_j f(x_j)} \\
&= \left| \sum_{j=1}^n \bar{c}_j f(x_j) \right|^2 \\
&\geq 0.
\end{aligned}$$

Podemos generalizar a ideia apresentada no exemplo anterior com o seguinte lema.

Lema 3.3. *Sejam \mathcal{H} um espaço de Hilbert com produto interno $\langle \cdot, \cdot \rangle$, X um conjunto não vazio e $f : X \rightarrow \mathcal{H}$ uma função qualquer. Então,*

- (i) *o núcleo $R : X \times X \rightarrow \mathcal{H}$ dado por $R(x, y) = \langle f(y), f(x) \rangle$, para todo $x, y \in X$, é positivo definido;*
- (ii) *o núcleo $R : X \times X \rightarrow \mathcal{H}$ dado por $R_1(x, y) = \langle f(x), f(y) \rangle$, para todo $x, y \in X$, é positivo definido.*

Demonstração. (i) Observe que

$$\begin{aligned}
\sum_{i,j=1}^n \bar{c}_j c_i R(x_j, x_i) &= \sum_{i,j=1}^n \bar{c}_j c_i \langle g(x_i), g(x_j) \rangle \\
&= \left\langle \sum_{i=1}^n c_i g(x_i), \sum_{j=1}^n \bar{c}_j g(x_j) \right\rangle \\
&= \left\| \sum_{i=1}^n c_i g(x_i) \right\|^2 \\
&\geq 0
\end{aligned}$$

Logo, $R(x, y) = \langle f(y), f(x) \rangle$, para todo $x, y \in X$, é um núcleo positivo definido.

(ii) Temos que

$$\begin{aligned}
R_1(x, y) &= \langle g(x), g(y) \rangle \\
&= \overline{\langle g(y), g(x) \rangle} \\
&= \overline{R(x, y)}.
\end{aligned}$$

Logo, R_1 também é um núcleo positivo definido. \square

O próximo teorema é importante pois nos fornece meios de criar núcleos positivos definidos a partir de outros já conhecidos. Sua demonstração pode ser encontrada em [15].

Teorema 3.4. *Sejam $R_1, R_2, \dots, R_l \in PD(X)$ e $c_1, c_2, \dots, c_l \geq 0$. Então:*

(i) *a soma $\sum_{i=1}^l c_i R_i$ está em $PD(X)$;*

(ii) *o produto $R_i R_j$ está em $PD(X)$;*

(iii) *se (R_n) converge pontualmente para R , então R está em $PD(X)$.*

No próximo exemplo veremos que, além de criar novos núcleos a partir de soma e produto de outros conhecidos, podemos também calcular a exponencial destes.

Exemplo 3.5. *Seja $K \in PD(X)$. Então,*

$$R(x, y) = e^{K(x, y)}, \quad x, y \in X,$$

é um núcleo positivo definido.

Note que podemos reescrever o núcleo R da seguinte forma

$$R(x, y) = e^{K(x, y)} = \sum_{i=0}^{\infty} \frac{(K(x, y))^i}{i!}, \quad x, y \in X.$$

Logo, pelo Teorema 3.4 segue que R é positivo definido.

Exemplo 3.6. *O núcleo $R : X \times X \rightarrow \mathbb{C}$ dado por*

$$R(x, y) = \cos(|x| - |y|).$$

é um núcleo positivo definido.

De fato, temos que $R(x, y) = \cos(|x| - |y|) = \cos |x| \cos |y| + \sin |x| \sin |y|$.

Exemplo 3.7. *O núcleo Gaussiano, também conhecido como função distribuição normal, dado por*

$$R(x, y) = e^{-\varepsilon^2 |x-y|^2}, \quad x, y \in \mathbb{R} \text{ e } \varepsilon > 0,$$

é positivo definido.

Antes de mostrar que o núcleo Gaussiano é positivo definido, considere o exemplo.

Exemplo 3.8. Considere o núcleo dado por

$$R_1(x, y) = e^{2\varepsilon^2 xy} = \sum_{j=0}^{\infty} \frac{(2\varepsilon^2)^j}{j!} x^j y^j, \quad x, y \in X \subset \mathbb{R}.$$

O núcleo R_1 é positivo definido (estrito).

De fato, note que

$$\begin{aligned} \sum_{i,j=0}^n \bar{c}_j c_i R_1(x_j, x_i) &= \sum_{i,j=0}^n \bar{c}_j c_i \sum_{k=0}^{\infty} \frac{(2\varepsilon^2)^k}{k!} x_i^k x_j^k \\ &= \sum_{k=0}^{\infty} \frac{(2\varepsilon^2)^k}{k!} \sum_{i,j=0}^n \bar{c}_j c_i x_i^k x_j^k \\ &= \sum_{k=0}^{\infty} \frac{(2\varepsilon^2)^k}{k!} \left| \sum_{i=0}^n c_i x_i^k \right|^2 \\ &\geq 0. \end{aligned}$$

Observe ainda que

$$\sum_{i,j=0}^n \bar{c}_j c_i x_i^k x_j^k = \left| \sum_{i=0}^n c_i x_i^k \right|^2 = 0, \quad k = 0, 1, \dots, \quad (3.2)$$

implica no sistema linear

$$\begin{cases} c_0 + c_1 + \dots + c_n = 0 \\ c_0 x_1 + c_1 x_2 + \dots + c_n x_2 = 0 \\ \vdots \\ c_0 x_1^n + c_1 x_2^n + \dots + c_n x_n^n = 0 \end{cases}$$

ou

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_1^n & x_2^n & \dots & x_n^n \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

A matriz do sistema é a matriz de Vandermonde, cujo determinante é não nulo se x'_i s são distintos. Isto é, o núcleo R_1 é estritamente positivo definido.

Agora, vamos mostrar que o núcleo Gaussiano, dado no Exemplo 3.7, é positivo definido (estrito).

Observe que podemos reescrever o núcleo da forma

$$\begin{aligned} R(x, y) &= e^{-\varepsilon^2|x-y|^2} \\ &= e^{-\varepsilon^2x^2 - \varepsilon^2y^2 + 2\varepsilon^2xy} \\ &= \sum_{j=0}^{\infty} \frac{(2\varepsilon^2)^j}{j!} e^{-\varepsilon^2x^2} e^{-\varepsilon^2y^2} x^j y^j, \quad x, y \in \mathbb{R}. \end{aligned}$$

Mais ainda,

$$\begin{aligned} \sum_{i,j=0}^n \bar{c}_j c_i R(x_j, x_i) &= \sum_{i,j=0}^n \bar{c}_j c_i \sum_{k=0}^{\infty} \frac{(2\varepsilon^2)^k}{k!} e^{-\varepsilon^2x_j^2} e^{-\varepsilon^2x_i^2} x_j^k x_i^k \\ &= \sum_{k=0}^{\infty} \frac{(2\varepsilon^2)^k}{k!} \sum_{i,j=0}^n \bar{c}_j c_i e^{-\varepsilon^2x_j^2} e^{-\varepsilon^2x_i^2} x_j^k x_i^k \\ &= \sum_{k=0}^{\infty} \frac{(2\varepsilon^2)^k}{k!} \sum_{i,j=1}^n \bar{d}_j d_i x_j^k x_i^k \\ &= \sum_{k=0}^{\infty} \frac{(2\varepsilon^2)^k}{k!} \left| \sum_{i=1}^n d_i x_i^k \right|^2 \\ &\geq 0, \end{aligned}$$

em que $d_i = c_i e^{-\varepsilon^2x_i^2}$, para $i = 1, 2, \dots, n$. Neste caso, $d_i = 0$ apenas quando $c_i = 0$. Assim, segue que este núcleo é estritamente positivo definido.

Para o caso em que $x, y \in \mathbb{R}^n$ procederemos de modo semelhante. Nesse sentido, considere o núcleo

$$R_1(x, y) = e^{2\varepsilon^2xy}, \quad x, y \in X^n \subset \mathbb{R}^n,$$

em que xy é o produto escalar entre x e y . Observe que

$$\begin{aligned} e^{2\varepsilon^2xy} &= e^{2\varepsilon^2(x_1y_1 + \dots + x_ny_n)} \\ &= e^{2\varepsilon^2x_1y_1} \times e^{2\varepsilon^2x_2y_2} \times \dots \times e^{2\varepsilon^2x_ny_n}. \end{aligned}$$

Segue que R_1 é positivo definido (estrito).

Assim, escrevendo o núcleo Gaussiano para $x, y \in \mathbb{R}^n$, temos que

$$\begin{aligned} R(x, y) &= e^{-\varepsilon^2|x-y|^2} \\ &= \left(e^{-\varepsilon^2\|x\|^2} e^{-\varepsilon^2\|y\|^2} \right) \times e^{2\varepsilon^2xy}, \end{aligned}$$

é positivo definido (estrito).

Pode-se mostrar ainda que o núcleo Laplaciano $K(x, y) = e^{-\varepsilon\|x-y\|}$, $x, y \in \mathbb{R}^n$ e $\varepsilon > 0$ é positivo definido. O núcleo $K(x, y) = e^{\|x-y\|}$, $x, y \in \mathbb{R}^n$ também é positivo definido [19].

Mais alguns exemplos serão dados na próxima seção ao falarmos de espaços de Hilbert de reprodução.

3.2 ESPAÇOS DE HILBERT DE REPRODUÇÃO

Segundo [15], a teoria de núcleos positivos definidos e espaços de Hilbert de reprodução possuem relações. Os espaços de Hilbert de reprodução são espaços que possuem um núcleo positivo definido. Em contrapartida, a partir de um núcleo positivo definido podemos gerar um espaço vetorial com produto interno, o qual pode ser completado formando um espaço de Hilbert de reprodução.

Nesta seção, estudaremos conceitos importantes sobre os espaços de Hilbert de reprodução. Primeiramente definiremos o conceito de função avaliação, pois este é necessário para definir um espaço de Hilbert de reprodução.

Definição 3.9. *Sejam X um conjunto não vazio, \mathbb{K} um corpo e \mathcal{H} um espaço de funções $f : X \rightarrow \mathbb{K}$. Para cada $x \in X$, a função $\varphi_x : \mathcal{H} \rightarrow \mathbb{K}$ dada por $\varphi_x(f) = f(x)$ é denominada função avaliação em x .*

Definição 3.10. *Seja X um conjunto não vazio. Um espaço de funções $f : X \rightarrow \mathbb{K}$ é um espaço de Hilbert de reprodução se φ_x é contínua para todo $x \in X$.*

O próximo teorema nos dá uma caracterização dos espaços de Hilbert de reprodução.

Teorema 3.11. *Seja \mathcal{H} um espaço de Hilbert de funções com produto interno $\langle \cdot, \cdot \rangle$. Então, \mathcal{H} será um espaço de Hilbert de reprodução se, e somente se, existe um núcleo positivo definido R , chamado de núcleo de reprodução, tal que:*

- (i) *para todo $x \in X$, temos que $R_x \in \mathcal{H}$, em que $R_x(y) = R(y, x)$, para todo $y \in X$;*
- (ii) *(Propriedade de reprodução). para todo $x \in X$ e para toda função $f \in \mathcal{H}$, temos que*

$$\langle f, R_x \rangle = f(x).$$

Além disso, o núcleo R é único e $R(y, x) = R_x(y) = \langle R_x, R_y \rangle$.

Demonstração. Seja \mathcal{H} um espaço de Hilbert com produto interno $\langle \cdot, \cdot \rangle$.

Primeiramente, suponhamos que φ_x é limitado (contínuo). Então, segue do Teorema da Representação de Riesz que existe um único $h_x \in \mathcal{H}$ tal que

$$\varphi_x(f) = \langle f, h_x \rangle, \quad \text{para toda } f \in \mathcal{H}.$$

Vamos definir $R(y, x) = h_x(y)$, para todo $x, y \in X$. Assim,

$$R_x = h_x \in \mathcal{H} \text{ e } \langle f, R_x \rangle = \varphi_x(f) = f(x).$$

Dessa forma, R é um núcleo associado a \mathcal{H} e satisfaz os itens (i) e (ii) do teorema. Além disso,

$$R(x, y) = \langle h_y, h_x \rangle \quad \text{para } x, y \in X,$$

é positivo definido pelo Lema 3.3.

Agora, suponhamos que \mathcal{H} tenha um núcleo associado R . Então, pela desigualdade de Cauchy-Schwarz, para toda $f \in \mathcal{H}$, temos que:

$$\begin{aligned} |\varphi_x(f)| &= |f(x)| \\ &= |\langle f, R_x \rangle| \\ &\leq \|R_x\| \|f\| \\ &= \sqrt{\langle R_x, R_x \rangle} \|f\| \\ &= \sqrt{R(x, x)} \|f\|. \end{aligned}$$

Assim, $\varphi_x : \mathcal{H} \rightarrow \mathbb{K}$ é limitada. Logo, φ_x é contínua e linear e, portanto, \mathcal{H} é um espaço de Hilbert de reprodução. \square

Corolário 3.12. *O espaço \mathcal{H} é um espaço de Hilbert de reprodução se, e somente se, toda função avaliação φ_x , $x \in X$, é limitada.*

Teorema 3.13. *Sejam \mathcal{H}_R um espaço de Hilbert de reprodução sobre o espaço métrico X e R um núcleo positivo definido sobre X . Valem as afirmações a seguir*

$$(i) \quad |f(x) - f(y)| \leq \|f\|_R \|R_x - R_y\|_R, \text{ para } f \in \mathcal{H}_R, x, y \in X;$$

(ii) *Se a sequência $\{f_n\} \subset \mathcal{H}_R$ converge para f em \mathcal{H}_R , $A \subset X$ e $k : X \rightarrow \mathbb{K}$, dada por*

$k(x) = R(x, x)$, é limitada em A , então a convergência é uniforme em A ;

(iii) Se a função k é limitada em subconjuntos compactos de X e cada função R_x é contínua, então $\mathcal{H}_R \subset C(X)$.

Demonstração. Ver em [20]. □

Existem versões mais gerais do Teorema 3.13, como em [4].

Corolário 3.14. Se R é uma função contínua, então $\mathcal{H}_R \subset C(X)$.

Lema 3.15. Se $\mathcal{H}_R \subset C(X)$ e $\{x_i\}$ é um conjunto denso em X . Então, o conjunto $[R_{x_i}]$ é denso em \mathcal{H} . Além disso, $[R_{x_i}]$ é sempre linearmente independente se, e somente se, R for um núcleo estritamente positivo definido.

Demonstração. Seja $f \in \mathcal{H}_K$ uma função contínua e ortogonal ao conjunto $\{R_{x_i}\}_{i \in A}$. Então segue que

$$f(x_i) = \langle f, R_{x_i} \rangle = 0$$

e, conseqüentemente, $f = 0$. Assim, $[\{R_{x_i}\}_{i \in A}]^\perp = \{0\}$ e, então pelo Lema 2.5 segue que $[\{R_{x_i}\}_{i \in A}]$ é denso em \mathcal{H} . Por fim, note que R é estritamente positivo definido se, e somente se,

$$\begin{aligned} g = \sum_{i=1}^n c_i R(x_i, \cdot) = 0 &\iff \langle g, g \rangle = \sum_{i,j=1}^n c_i \bar{c}_j R(x_i, x_j) = 0 \\ &\iff c_i = 0, \quad i = 1, 2, \dots, n. \end{aligned}$$

□

O resultado a seguir fornece uma propriedade de reprodução para derivadas e será útil na demonstração do Corolário 3.17.

Seja Y um conjunto não vazio de \mathbb{R}^n , denotaremos por $C^s(Y)$ o conjunto das funções $h : Y \rightarrow \mathbb{K}$ contínuas em Y tais que existe a derivada de h de ordem α , para $0 \leq |\alpha| \leq s$, para $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{N}^n$, e $|\alpha| = \sum_{i=1}^n \alpha_i$.

Teorema 3.16. Seja R um núcleo de reprodução em $C^{2s}(X \times X)$. Se a função $k : X \rightarrow \mathbb{K}$, dada por $k(x) = R(x, x)$, é tal que $\int_X k(x) dx < \infty$ e $0 \leq |\alpha| \leq s$, então

(i) $\mathcal{H}_R \subset C^s(X)$;

(ii) se $h \in \mathcal{H}_R$, então

$$|h^{(\alpha)}(x)| \leq \left(\frac{\partial^{2\alpha}}{\partial x^\alpha \partial y^\alpha} R(x, x) \right)^{\frac{1}{2}} \|h\|_R, \quad x \in X; \quad (3.3)$$

(iii) para cada derivada $\frac{\partial^\alpha}{\partial y^\alpha} R(\cdot, x) \in \mathcal{H}_R$ vale a propriedade de reprodução

$$h^{(\alpha)}(x) = \left\langle h, \frac{\partial^\alpha}{\partial y^\alpha} R(\cdot, x) \right\rangle_{\mathcal{H}_R}, \quad x \in X, \quad h \in \mathcal{H}_R.$$

Demonstração. Ver em [18]. □

Corolário 3.17. *Sob as condições do Teorema 3.16, se f_n converge para f em \mathcal{H}_K então todas as derivadas $f_n^{(\alpha)}(x)$ convergem uniformemente para $f^{(\alpha)}(x)$, em conjuntos compactos de X , quando $0 \leq |\alpha| \leq s$.*

Demonstração. Segue do Teorema 3.16 que, para qualquer $x \in X$, temos

$$\begin{aligned} |f_n^{(\alpha)}(x) - f^{(\alpha)}(x)| &= \left| \left\langle f_n - f, \frac{\partial^\alpha}{\partial y^\alpha} (R_x) \right\rangle \right| \\ &\leq \|f_n - f\| \left\| \frac{\partial^\alpha}{\partial y^\alpha} (R_x) \right\| \end{aligned}$$

Consequentemente, se f_n converge para f na norma de \mathcal{H} , suas derivadas $f_n^{(\alpha)}(x)$ convergem uniformemente para $f^{(\alpha)}(x)$, se $0 \leq |\alpha| \leq s$. □

Conforme [15], podemos criar um espaço de Hilbert de reprodução a partir de um núcleo positivo definido. Para tanto, é necessário utilizar o processo de completamento de um espaço vetorial. Tal processo é construtivo, garantindo então sua existência, e pela unicidade do completamento, garantimos que este espaço é único.

Seja R um núcleo positivo definido. Considere o espaço vetorial formado pelas funções

$$g(x) = \sum_{i=1}^n c_i R(x, x_i), \quad \text{para } x \in X, \quad (3.4)$$

para todo $n \geq 1$, $\{x_1, x_2, \dots, x_n\} \subset X$ e $\{c_1, c_2, \dots, c_n\} \subset \mathbb{K}$. A definição de núcleo positivo definido nos permite definir um produto interno neste espaço, dado por

$$\langle g, h \rangle = \sum_{i,j=1}^n c_i \bar{d}_j R(x_j, x_i),$$

para

$$h(x) = \sum_{j=1}^n d_j R(x, x_j), \quad \text{para } x \in X. \quad (3.5)$$

Temos, portanto, que o completamento desse espaço é um espaço de Hilbert de reprodução \mathcal{H} , com núcleo de reprodução R , pois $R_x \in \mathcal{H}$ e $\langle f, R_x \rangle = f(x)$, para toda função $f \in \mathcal{H}$ e para todo $x \in X$.

Veja [21] para detalhes sobre a construção do espaço de reprodução com núcleo gaussiano.

3.3 EXEMPLOS

Nesta seção serão apresentados alguns exemplos de espaços de Hilbert de reprodução, os quais serão necessários na aproximação de soluções de equações funcionais estudadas mais adiante neste trabalho. Para ver outros espaços de reprodução, sugerimos por exemplo [22].

Exemplo 3.18. *Considere o conjunto $W_2^m[a, b]$, $m \geq 1$, constituído por funções $u : [a, b] \rightarrow \mathbb{R}$ tais que*

(i) $u^{(m-1)}$ é absolutamente contínua em $[a, b]$;

(ii) $\int_a^b |u^{(m)}(x)|^2 dx < \infty$.

Esse conjunto com o produto interno dado por

$$\langle u, v \rangle = \sum_{i=0}^{m-1} u^{(i)}(a)v^{(i)}(a) + \int_a^b u^{(m)}(t)v^{(m)}(t)dt, \quad u, v \in W_2^m[a, b],$$

e norma $\|u\| = \sqrt{\langle u, u \rangle}$, para $u \in W_2^m[a, b]$, é um espaço de Hilbert de reprodução.

A demonstração de que $W_2^m[a, b]$ é um espaço de Hilbert de reprodução pode ser encontrada em [22].

Veremos a seguir alguns casos particulares do Exemplo 3.18 que serão úteis mais adiante.

Exemplo 3.19. *O espaço $W_2^1[a, b]$, dado no Exemplo 2.20, é um espaço de Hilbert de reprodução com núcleo de reprodução R .*

A demonstração de que $W_2^1[a, b]$ é, de fato, um espaço de Hilbert de reprodução pode ser vista em [15] para o caso em que $a = 0$ e $b = 1$. É possível ainda mostrar que o núcleo R , dado no Exemplo 3.19, é estritamente positivo definido [4].

Exemplo 3.20. O conjunto $W_2^1[a, b]$, dado anteriormente, com o produto interno

$$\langle u, v \rangle = \int_a^b (u(t)v(t) + u'(t)v'(t)) dt$$

e norma $\|u\| = \sqrt{\langle u, u \rangle}$, para $u \in W_2^1[a, b]$, é um espaço de Hilbert de reprodução com núcleo de reprodução

$$R_1(x, y) = \frac{1}{2 \sinh(b-a)} [\cosh(x+y-b-a) + \cosh(|x-y|-b+a)].$$

O espaço $W_2^1[a, b]$ com núcleo de reprodução R_1 pode ser encontrado em [2].

Daremos uma ideia de como mostrar que o núcleo R_1 é, de fato, um núcleo de reprodução. Observe que

$$\begin{aligned} \cosh(x+y-b-a) + \cosh(|x-y|-b+a) &= \frac{e^{x+y-(b+a)} + e^{-x-y+(b+a)}}{2} \\ &\quad + \frac{e^{|x-y|-(b-a)} + e^{-|x-y|+(b-a)}}{2} \\ &= \frac{1}{2} (e^x e^y e^{-b} e^{-a} + e^{-x} e^{-y} e^b e^a) \\ &\quad + \frac{1}{2} (e^{|x-y|} e^{-b} e^a + e^{-|x-y|} e^b e^{-a}). \end{aligned}$$

Exemplo 3.21. O conjunto $W_2^2[0, b]$ formado por funções $u : [0, b] \rightarrow \mathbb{R}$ contínuas, tais que:

$$(i) \quad u(x) = u(0) + \int_0^x u'(s) ds;$$

$$(ii) \quad u'(x) = u'(0) + \int_0^x u''(s) ds;$$

$$(iii) \quad \int_0^b |u''(x)|^2 dx < \infty,$$

O conjunto $W_2^2[0, b]$ com o produto interno

$$\langle u, v \rangle = u(0)v(0) + u'(0)v'(0) + \int_0^b u''(s)v''(s) ds, \quad u, v \in W_2^2[0, b]$$

e norma $\|u\| = \sqrt{\langle u, u \rangle}$, para $u \in W_2^2[0, b]$, é um espaço de Hilbert de reprodução com núcleo

de reprodução

$$R_1(x, y) = \begin{cases} 1 + xy + \frac{xy^2}{2} - \frac{y^3}{6}, & y \leq x \\ 1 + xy + \frac{x^2y}{2} - \frac{x^3}{6}, & y > x. \end{cases}$$

A demonstração de que $W_2^2[0, b]$ com núcleo R_1 é, de fato, um espaço de Hilbert de reprodução pode ser vista em [15] para o caso em que $b = 1$.

Exemplo 3.22. O conjunto $W_{2,1}^2[0, b]$ formado por funções $u : [0, b] \rightarrow \mathbb{R}$ contínuas, com $u(0) = 0$ e tais que:

$$(i) \quad u(x) = \int_0^x u'(s) ds;$$

$$(ii) \quad u'(x) = u'(0) + \int_0^x u''(s) ds;$$

$$(iii) \quad \int_0^b |u''(x)|^2 dx < \infty,$$

com produto interno

$$\langle u, v \rangle = u(b)v(b) + \int_0^b u''(s)v''(s) ds, \quad u, v \in W_{2,1}^2[0, b],$$

e norma $\|u\| = \sqrt{\langle u, v \rangle}$, para $u \in W_{2,1}^2[0, b]$, é um espaço de Hilbert de reprodução com núcleo de reprodução

$$R(x, y) = \begin{cases} \frac{y((x-b)by^2 + x(2b^2 - 3xb^2 + x^2b + 6))}{6b^2}, & y \leq x \\ \frac{x((y-b)bx^2 + y(2b^2 - 3yb^2 + y^2b + 6))}{6b^2}, & y > x. \end{cases}$$

Para detalhes veja [6].

O espaço $W_{2,1}^2[0, b]$ é um subespaço de $W_2^2[0, b]$ (Para mais exemplos de subespaços de espaços de reprodução sugerimos [22]).

O Teorema 3.11 nos diz que o núcleo de reprodução de um dado espaço de Hilbert é único. Observe que os Exemplos 3.19 e 3.20 apresentam núcleos diferentes para o mesmo conjunto. Todavia, não contradizem o teorema, pois o conjunto de funções inicial é o mesmo, mas ao trocarmos o produto interno mudamos o espaço de Hilbert. Justificando, assim, a existência de um único núcleo para cada um.

4 APROXIMAÇÃO POR MÉTODOS DE PROJEÇÃO

Segundo [23], métodos de projeção são métodos utilizados para aproximar a solução de problemas usando expansões finitas em termos de elementos da base do espaço em estudo.

Para entender estes métodos, é útil estudar operadores de projeção.

Definição 4.1. *Seja V um espaço normado e U um subespaço não vazio de V . Um operador linear limitado $P : V \rightarrow U$, com a propriedade $P(u) = u$, para todo $u \in U$, é chamado de operador de projeção de V em U .*

Teorema 4.2. *Seja $P : V \rightarrow V$ um operador linear limitado não nulo. Então, P é um operador de projeção se, e somente se, $P^2 = P$. Além disso, operadores de projeção satisfazem $\|P\| \geq 1$.*

Demonstração. Seja $P : V \rightarrow V$ um operador linear limitado e não nulo.

Suponha que P é um operador de projeção. Existe $U \subset V$ tal que $P(u) = u$, para todo $u \in U$. Mais ainda, $Im(P) = U$. Dessa forma, dado $v \in V$ temos que

$$\begin{aligned} P^2(v) &= P(P(v)) \\ &= P(v) \\ &= v. \end{aligned}$$

Logo, $P^2 = P$.

Suponha agora que $P^2 = P$ e considere $U = P(V) \subseteq V$. Assim, para todo $v \in U$ podemos escrever $u = P(v)$ para algum $v \in V$. Como, por hipótese, $P^2 = P$, segue que $u = P(v) = P^2(v) = P(P(v)) = P(u)$. Portanto, P é um operador de projeção. \square

Sejam V um espaço de Banach e $u_n \in V_n$, com $n \geq 1$, uma sequência usada para aproximar u , em que V_n é um subespaço de dimensão finita, para todo n . Consideremos k_n a dimensão de V_n e assumimos que quando $n \rightarrow \infty$, $k_n \rightarrow \infty$.

Assim, procuramos definir o operador de projeção $P_n : V \rightarrow V_n$ para resolver a equação $T(u) = u$. Como $u_n \in V_n$, $P_n(u_n) = u_n \rightarrow u$, quando $n \rightarrow \infty$. Então $P_n(u) \rightarrow u$, $u \in V$.

A ideia é tomar um tal $u_n \in V_n$ de modo que

$$u_n = P_n(T(u_n)), \tag{4.1}$$

que é chamado de ponto fixo aproximado ou simplesmente de solução aproximada do problema de ponto fixo. Ainda, resolver a Equação (4.1) é equivalente a resolver $P_n(u_n - T(u_n)) = 0$.

Em resumo, a ideia para os métodos de projeção é procurar uma aproximação para u da forma $u_n \in V_n$. Na literatura, quando P_n é uma projeção ortogonal, chamamos este procedimento de método de Galerkin (Veja [23, 24]). E, quando P_n é um operador de interpolação, denominamos método de colocação.

4.1 MÉTODO DE COLOCAÇÃO

Consideremos um caso particular. Seja a seguinte equação integral não linear

$$f(t) = \int_X N(t, x, f(x)) dx + g(t) \quad (4.2)$$

em que $N \in \mathcal{C}[X^2 \times \mathbb{R}^n; \mathbb{R}^n]$ é contínua e diferenciável em relação à $f(x)$.

Seja

$$f_n(t) = \sum_{j=1}^{k_n} c_j \Psi_j(t), \quad t \in X, \quad (4.3)$$

em que $\{\Psi_i\}_{i=1}^{k_n}$ é um conjunto linearmente independente em V_n . Consideremos ainda V como o espaço das funções contínuas e $P_n : V \rightarrow V_n$ um operador de projeção interpolatório. Vamos reescrever a Equação (4.2) como segue

$$\varepsilon(f_n(t)) = f_n(t) - \int_X N(t, x, f_n(x)) dx - g(t). \quad (4.4)$$

Note que, se $f(t)$ é uma solução exata da Equação (4.2), então o resíduo $\varepsilon(f(t)) = 0$. Portanto, devemos escolher os coeficientes $\{c_j\}$ de modo que o resíduo seja tão pequeno quanto possível.

Assim, das Equações (4.3) e (4.4), segue que

$$\varepsilon(f_n(t)) = \sum_{j=1}^{k_n} c_j \Psi_j(t) - \int_X N\left(t, x, \sum_{j=1}^{k_n} c_j \Psi_j(x)\right) dx - g(t). \quad (4.5)$$

Pelo método de colocação, precisamos que o resíduo seja igual a zero num dado sistema de pontos de colocação $t_1, t_2, \dots, t_n \in X$, ou seja, $\varepsilon(f_n(t_i)) = 0$, para todo $i = 1, 2, \dots, n$.

Dessa forma, obtemos o seguinte sistema para os coeficientes $\{c_j\}$:

$$\sum_{j=1}^{k_n} c_j \Psi_j(t_i) - \int_X N \left(t_i, x, \sum_{j=1}^{k_n} c_j \Psi_j(x) \right) dx - g(t_i) = 0, \quad i = 1, 2, \dots, n. \quad (4.6)$$

Colocando o problema na forma matricial, obtemos

$$\underbrace{\begin{bmatrix} \Psi_1(t_1) & \dots & \Psi_{k_n}(t_1) \\ \vdots & \ddots & \vdots \\ \Psi_1(t_{k_n}) & \dots & \Psi_{k_n}(t_{k_n}) \end{bmatrix}}_{[\Psi]} \underbrace{\begin{bmatrix} c_1 \\ \vdots \\ c_{k_n} \end{bmatrix}}_{[c]} = \underbrace{\begin{bmatrix} G(t_1, c) \\ \vdots \\ G(t_{k_n}, c) \end{bmatrix}}_{[G_c]},$$

Então, temos

$$[\Psi][c] = [G_c]$$

Logo,

$$\begin{aligned} [c] &= [\Psi]^{-1}[G_c] \\ &= [T(c)]. \end{aligned}$$

Na prática, fornecemos uma aproximação inicial $[c_0]$ e calculamos o processo iterativo

$$[c_j] = [T(c_{j-1})], \quad j = 1, 2, \dots, p. \quad (4.7)$$

Geralmente, são utilizados métodos numéricos para resolver tal sistema, devido a sua complexidade.

No próximo capítulo, serão apresentados exemplos do funcionamento do método de colocação.

4.2 MÉTODO DE ESPAÇOS DE HILBERT DE REPRODUÇÃO

Nesta seção faremos um estudo sobre o método de espaços de Hilbert de reprodução na aproximação da solução de equações funcionais de interesse. O método é baseado em toda a teoria vista no capítulo anterior. Este procedimento se mostra útil para equações integrais, diferenciais e integro diferenciais lineares e não lineares, como será visto mais adiante.

Sejam os espaços de Hilbert de reprodução \mathcal{H}_R e \mathcal{H}_K , com núcleos de reprodução R e

K , respectivamente. Considere a seguinte equação funcional

$$L(f)(x) = G(x, f), \quad (4.8)$$

em que $x \in X$, $G : X \times \mathcal{H}_R \longrightarrow \mathcal{H}_K$, $f \in \mathcal{H}_R$ e $L : \mathcal{H}_R \longrightarrow \mathcal{H}_K$ é um operador linear limitado.

Os próximos resultados têm o objetivo de formalizar os procedimentos de aplicação da teoria de espaços de Hilbert de reprodução em equações funcionais. Tais procedimentos serão tratados mais adiante ainda nesta seção.

O lema enunciado a seguir nos fornece um meio de calcular $L^*K_y(x)$ por meio do operador L , sem encontrar L^* . É um resultado importante pois, além de poupar tempo, nem sempre é trivial encontrar o adjunto de um operador linear.

Lema 4.3. *Seja $L^* : \mathcal{H}_K \rightarrow \mathcal{H}_R$ o operador adjunto de L . Então*

$$Lf(x) = \langle Lf, K_x \rangle_K = \langle f, L^*K_x \rangle_R, \quad f \in \mathcal{H}_R,$$

e

$$L^*K_y(x) = \langle L^*K_y, R_x \rangle_R = \langle K_y, LR_x \rangle_K = \overline{LR_x(y)}, \quad x, y \in X.$$

O próximo lema nos fornece um conjunto linearmente independente no espaço de reprodução tratado.

Lema 4.4. *Seja $L : \mathcal{H}_R \longrightarrow \mathcal{H}_K$ um operador linear limitado, com $\mathcal{H}_R \subset \mathcal{C}(X)$. Se $\{x_i\}_{i \in A}$ é denso em X , então o conjunto gerado por $\{LR_{x_i}\}_{i \in A}$ é denso em \mathcal{H}_K se, e somente se, L tem uma imagem densa. Particularmente, se L é bijetor e R é estritamente positivo definido, então $\{LR_{x_i}\}_{i \in A}$ é linearmente independente.*

Demonstração. Segue do Lema 3.15 que o conjunto gerado por $\{R_{x_i}\}_{i \in A}$ é denso em \mathcal{H}_R . Se L tem imagem densa em \mathcal{H}_K , então para toda função $g \in \mathcal{H}_K$ existe uma função $h = L(f)$, tão perto quanto se queira de g , para algum $f \in \mathcal{H}_R$. Assim, temos que existe q no conjunto gerado por $\{R_{x_i}\}_{i \in A}$, tão perto quanto se queira de f . Dessa forma, $L(q)$ está no gerado por $\{L(R_{x_i})\}_{i \in A}$ e tão perto quanto se queira de h . Logo, o resultado está provado. \square

Corolário 4.5. *Seja $L : \mathcal{H}_R \longrightarrow \mathcal{H}_K$ um operador linear injetivo limitado, com $\mathcal{H}_R \subset \mathcal{C}(X)$. Se $\{x_i\}_{i \in A}$ é denso em X e R é estritamente positivo definido, então $\{L(R_{x_i})\}_{i \in A}$ é linearmente independente em \mathcal{H}_K .*

O corolário a seguir é importante para o método de espaços de Hilbert de reprodução, pois dá a forma de uma base ortonormal para o espaço de Hilbert em estudo, permitindo representar a solução de equações por meio dos termos desta base.

Corolário 4.6. *Seja $L : \mathcal{H}_R \rightarrow \mathcal{H}_K$ um operador linear bijetor limitado, com $\mathcal{H}_K \subset \mathcal{C}(X)$. Se $\{x_i\}_{i \in A}$ é denso em X , então o conjunto gerado por $\{L^* K_{x_i}\}_{i \in A}$ é denso em \mathcal{H}_R . Particularmente, se K é estritamente positivo definido então $\{L^* K_{x_i}\}_{i \in A}$ é linearmente independente. Se $A \subset \mathbb{N}$, então podemos encontrar uma base ortonormal $\{\Psi_i\}$ para \mathcal{H}_R , através do processo de ortogonalização de Gram-Schmidt, em que*

$$\Psi_i(x) = \sum_{j=1}^i \beta_{ji} L^* K_{x_j}(x). \quad (4.9)$$

Observação 4.7. *Note que a hipótese do último resultado nos diz que quando usamos o processo Gram-Schmidt no conjunto linearmente independente $\{L^*(K_{x_i})\}_{i=1}^n$ para obter o conjunto ortonormal $\{\Psi_i\}_{i=1}^n$ estamos fazendo algo semelhante ao processo de decomposição QR [25], em que o primeiro conjunto age como uma matriz $\mathbf{A} = \mathbf{Q}\mathbf{R}$, o segundo conjunto age como \mathbf{Q} e a matriz $\mathbf{B} = [\beta_{ij}]_{n \times n}$ como a matriz inversa de \mathbf{R} . Isto significa que \mathbf{B} é uma matriz triangular superior com elementos positivos na diagonal.*

Para aplicar o último resultado, assumimos a separabilidade de todos os espaços Hilbert de reprodução a partir de agora.

Neste contexto, podemos expressar a solução da Equação (4.8) usando a Identidade de Parseval.

Teorema 4.8. *Considere que a Equação (4.8) possui solução única f . Então, sob as condições do Corolário 4.6, temos que*

$$f_n(x) = \sum_{i=1}^n \alpha_i \Psi_i(x),$$

converge para $f(x)$ com convergência uniforme e na norma de \mathcal{H}_R , em que

$$\begin{aligned}
\alpha_i &= \langle f, \Psi_i \rangle_R \\
&= \left\langle f, \sum_{j=1}^i \beta_{ji} L^*(K_{x_j}) \right\rangle_R \\
&= \left\langle L(f), \sum_{j=1}^i \beta_{ji} K_{x_j} \right\rangle_K \\
&= \sum_{j=1}^i \overline{\beta_{ji}} L(f)(x_j) \\
&= \sum_{j=1}^i \overline{\beta_{ji}} G(x_j, f)
\end{aligned}$$

Demonstração. Do Corolário 4.6, segue que $\{\Psi_i\}_{i=1}^{\infty}$ é uma base ortonormal de \mathcal{H}_R . Observe que da propriedade de reprodução $\langle v, R_{x_i} \rangle = v(x_i)$ para cada $v \in \mathcal{H}_K$, temos que

$$\begin{aligned}
f(x) &= \sum_{i=1}^{\infty} \langle f, \Psi_i \rangle_R \Psi_i(x) \\
&= \sum_{i=1}^{\infty} \sum_{k=1}^i \overline{\beta_{ki}} \langle f, L^* R_{x_k} \rangle_R \Psi_i(x) \\
&= \sum_{i=1}^{\infty} \sum_{k=1}^i \overline{\beta_{ki}} \langle Lf, R_{x_k} \rangle_K \Psi_i(x) \\
&= \sum_{i=1}^{\infty} \sum_{k=1}^i \overline{\beta_{ki}} Lf(x_k) \Psi_i(x) \\
&= \sum_{i=1}^{\infty} \sum_{k=1}^i \overline{\beta_{ki}} G(x_k, f) \Psi_i(x).
\end{aligned}$$

□

O lema seguinte mostra que este procedimento produz algo semelhante a um processo interpolatório. Vemos do Teorema 4.8 que aqui não ocorre o efeito de Runge fora dos pontos interpolados.

Lema 4.9. *Nas condições do Teorema 4.8, temos que*

$$Lf_n(x_j) = G(x_j, f), \quad \text{para } j \leq n. \quad (4.10)$$

Demonstração. Vamos definir o operador de projeção $P : \mathcal{H}_R \longrightarrow \mathcal{H}_R$, dado por

$$P_n(g) = \sum_{i=1}^n \langle g, \Psi_i \rangle \Psi_i, \quad g \in \mathcal{H}_R.$$

Assim, $f_n = P_n(f)$ e vamos mostrar que $Lf_n(x_j) = Lf(x_j)$, para $j \leq n$. De fato,

$$\begin{aligned} Lf_n(x_j) &= \langle Lf_n, K_{x_j} \rangle_K \\ &= \langle f_n, L^* K_{x_j} \rangle_R \\ &= \langle P_n(f), L^* K_{x_j} \rangle_R \\ &= \langle f, P_n(L^* K_{x_j}) \rangle_R \\ &= \langle f, L^* K_{x_j} \rangle_R \\ &= \langle Lf, K_{x_j} \rangle_K \\ &= Lf(x_j). \end{aligned}$$

□

Para o caso em que a equação $Lf = g$ é linear, os resultados anteriores nos dão uma forma direta de aproximar f .

Corolário 4.10. *Considere que, para toda função $f \in \mathcal{H}_R$, $L(f)(x) = g(x)$, possui solução única. Então, sob as condições do Corolário 4.6, temos que*

$$f(x) = \sum_{i=1}^{\infty} \alpha_i \Psi_i(x), \quad x \in X \quad (4.11)$$

com convergência em \mathcal{H}_K e uniforme, em que

$$\alpha_i = \sum_{j=1}^i \overline{\beta_{ji}} g(x_j). \quad (4.12)$$

A seguir é apresentada uma sequência de passos sugeridos para efetuar a implementação desse resultado como algoritmo.

Passo 1. Definir o produto interno e a norma do espaço de reprodução.

Passo 2. Definir n como o número de pontos x_i .

Passo 3. Definir as funções K , g e o operador L .

Passo 4. Definir a função $L^*K_{x_i}$.

Passo 5. Calcular as funções $\Psi_i(x) = \sum_{j=1}^i \beta_{ji} L^*K_{x_j}$, para todo i .

Passo 6. Calcular a solução aproximada $f_n = \sum_{i=1}^n \sum_{j=1}^i \beta_{ji} g(x_j) \Psi_i$.

Passo 7. Plotar o gráfico para teste.

4.2.1 Método de Espaços de Hilbert de Reprodução para Equações Não Lineares

Se $G(x, f)$ é uma função não linear de f então, podemos utilizar um processo iterativo para aproximar f_n dada pelo Teorema 4.8. Uma primeira tentativa, sem garantia formal de convergência, é escolher uma função $\tilde{f}_{n,0} \in \mathcal{H}_R$ e aplicar o processo iterativo

$$\tilde{f}_{n,j} = \sum_{i=1}^n \left(\sum_{j=1}^i \bar{\beta}_{ji} G(x_j, \tilde{f}_{n,j-1}) \right) \Psi_i, \quad j = 1, 2, \dots, \quad (4.13)$$

para aproximar f_n , para $n = 1, 2, \dots$. Este procedimento parece ser algum tipo de método de projeção. Se $\tilde{f}_{n,j}$ converge para f_n , quando $j \rightarrow \infty$, parece que $\tilde{f}_{n+1,0} = f_n$ é uma boa escolha, veja a Equação (14) em [26].

Outro ponto de vista, adaptado de [6, 26] para um caso geral, é escolher uma função $\tilde{f}_0 \in \mathcal{H}_R$ e definir a sequência

$$\tilde{f}_n(x) = \sum_{i=1}^n B_i \Psi_i(x), \quad n = 1, 2, \dots, \quad (4.14)$$

em que os coeficientes B_i são dados por

$$\begin{aligned} B_1 &= \bar{\beta}_{11} G(x_1, \tilde{f}_0) \\ B_2 &= \sum_{k=1}^2 \bar{\beta}_{k2} G(x_k, \tilde{f}_{k-1}) \\ &\vdots \\ B_i &= \sum_{k=1}^i \bar{\beta}_{ki} G(x_k, \tilde{f}_{k-1}) \\ &\vdots \end{aligned} \quad (4.15)$$

Vamos mostrar que $\tilde{f}_n(x)$ dada pelo processo iterativo converge para a solução exata $f(x)$ mesmo em um contexto mais geral que o apresentado por [6].

Lema 4.11. *Suponha que \tilde{f}_n converge para alguma $f \in \mathcal{H}_R$ e que x_n converge para $x \in X$. Se $G(x, f)$ é contínua em $X \times \mathcal{H}_R$, então $G(x_n, \tilde{f}_{n-1}) \rightarrow G(x, f)$ quando $n \rightarrow \infty$.*

Demonstração. Como $\|\tilde{f}_n - \tilde{f}\| \rightarrow 0$ quando $n \rightarrow \infty$, pelo Corolário 3.17 segue que $\tilde{f}_n(x)$ converge uniformemente para $f(x)$ e o resultado segue pela continuidade de G . \square

Para utilizarmos o Lema 4.11, apresentamos o que segue, que se assemelha ao Lema 4.9, mas é diferente. Registramos aqui que discordamos então do item (b) do Teorema 4.1 de [6].

Lema 4.12. *Temos que $L\tilde{f}_n(x_j) = G(x_j, \tilde{f}_{j-1})$, para $j \leq n$.*

Demonstração. Se $j \leq n$, então

$$\begin{aligned} L\tilde{f}_n(x) &= \sum_{i=1}^n B_i L\Psi_i(x) \\ &= \sum_{i=1}^n B_i \langle L\Psi_i, K_x \rangle_K \\ &= \sum_{i=1}^n B_i \langle \Psi_i, L^* K_x \rangle_R \end{aligned}$$

Portanto, obtemos

$$L\tilde{f}_n(x_j) = \sum_{i=1}^n B_i \langle \Psi_i, L^* K_{x_j} \rangle_R. \quad (4.16)$$

Multiplicando ambos os lados da Equação (4.16) por $\bar{\beta}_{kj}$, somando para k de 1 até j e usando a ortogonalidade de $\{\Psi_i\}_{i=1}^\infty$, obtemos que

$$\begin{aligned} \sum_{k=1}^j \bar{\beta}_{kj} L\tilde{f}_n(x_k) &= \sum_{i=1}^n B_i \left\langle \Psi_i, \sum_{k=1}^j \bar{\beta}_{kj} L^* K_{x_k} \right\rangle_R \\ &= \sum_{i=1}^n B_i \langle \Psi_i, \Psi_j \rangle_R \\ &= B_j \\ &= \sum_{k=1}^j \bar{\beta}_{kj} G(x_k, \tilde{f}_{k-1}). \end{aligned}$$

Por outro lado, se $j = 1$, então $L\tilde{f}_n(x_1) = G(x_1, \tilde{f}_0)$. Se $j = 2$, então

$$\bar{\beta}_{12}L\tilde{f}_n(x_1) + \bar{\beta}_{22}L\tilde{f}_n(x_2) = \bar{\beta}_{12}G(x_1, \tilde{f}_0) + \bar{\beta}_{22}G(x_2, \tilde{f}_1),$$

e segue que $L\tilde{f}_n(x_2) = G(x_2, \tilde{f}_1)$.

Assim, por indução temos que

$$L\tilde{f}_n(x_j) = G(x_j, \tilde{f}_{j-1}).$$

□

Apresentamos agora mais um resultado técnico.

Lema 4.13. *A sequência $\{\tilde{f}_n\}_{n=1}^\infty$, dada na Equação (4.14), é monótona não decrescente com relação a norma de \mathcal{H}_R .*

Demonstração. Do Corolário 4.6 segue que $\{\Psi_i\}_{i=1}^\infty$ é uma base ortonormal de \mathcal{H}_R . Consequentemente, temos que

$$\begin{aligned} \|\tilde{f}_n\|^2 &= \langle \tilde{f}_n, \tilde{f}_n \rangle \\ &= \left\langle \sum_{i=1}^n B_i \Psi_i, \sum_{i=1}^n B_i \Psi_i \right\rangle_R \\ &= \sum_{i=1}^n (B_i)^2. \end{aligned}$$

Assim, $\|\tilde{f}_n\|^2$ é monótona não decrescente. □

Podemos agora demonstrar o resultado principal desta seção.

Teorema 4.14. *Suponha que $\|\tilde{f}_n\|$, dada na Equação (4.14), é limitada.*

- (a) *Se $\{x_i\}_{i=1}^\infty$ é denso em X , então a solução aproximada \tilde{f}_n , dada pela Equação (4.14), converge para alguma função \tilde{f} em \mathcal{H}_R ;*
- (b) *Se $G(x, \tilde{f})$ é uma função contínua, então $\tilde{f}(x) = \sum_{i=1}^\infty B_i \Psi_i(x)$ é a solução exata da Equação (4.8).*

Demonstração. (a) Vamos mostrar a convergência de $\tilde{f}_n(x)$. Da Equação (4.14), temos que $\tilde{f}_{n+1}(x) = \tilde{f}_n(x) + B_{n+1}\Psi_{n+1}(x)$. Da ortogonalidade de $\{\Psi_i\}_{i=1}^{\infty}$ segue que

$$\begin{aligned}\|\tilde{f}_{n+1}\|^2 &= \|\tilde{f}_n\|^2 + (B_{n+1})^2 \\ &= \|\tilde{f}_{n-1}\|^2 + (B_n)^2 + (B_{n+1})^2 \\ &\vdots \\ &= \|\tilde{f}_0\|^2 + \sum_{i=1}^{n+1} (B_i)^2.\end{aligned}$$

Pelo Lema 4.13, a sequência $\|\tilde{f}_n\|$ é monotônica crescente. Entretanto, devido ao fato de $\|\tilde{f}_n\|$ ser limitada, segue que $\|\tilde{f}_n\|$ é convergente. Então, existe uma constante z tal que $\sum_{i=1}^{\infty} (B_i)^2 = z$. Isto implica que $(B_i) = \left(\sum_{k=1}^i \beta_{ik} G(x_k, \tilde{f}_{k-1}) \right) \in l^2$, para $i = 1, 2, \dots$, em que $l^2 = \{B; \sum_{i=1}^{\infty} (B_i)^2 < \infty\}$ é um espaço de Hilbert de sequências.

Seja $s > n$. Como $(\tilde{f}_s - \tilde{f}_{s-1}) \perp (\tilde{f}_{s-1} - \tilde{f}_{s-2}) \perp \dots \perp (\tilde{f}_{n+1} - \tilde{f}_n)$ segue que

$$\begin{aligned}\|\tilde{f}_s - \tilde{f}_n\|^2 &= \|\tilde{f}_s - \tilde{f}_{s-1} + \tilde{f}_{s-1} - \dots + \tilde{f}_{n+1} - \tilde{f}_n\|^2 \\ &\leq \|\tilde{f}_s - \tilde{f}_{s-1}\|^2 + \dots + \|\tilde{f}_{n+1} - \tilde{f}_n\|^2.\end{aligned}$$

Como resultado, temos que $\|\tilde{f}_s(x) - \tilde{f}_{s-1}(x)\|^2 = \sum_{i=n+1}^s (B_i)^2$ converge para 0 quando n, s crescem e segue que $\{\tilde{f}_n\}$ é uma sequência de Cauchy.

Da completude de \mathcal{H}_R , segue que existe uma função $f \in \mathcal{H}_R$ tal que \tilde{f}_n converge para f , em relação a norma de \mathcal{H}_R . Portanto, $\tilde{f} = \sum_{i=1}^{\infty} B_i \Psi_i$.

(b) Como $\{x_i\}_{i=1}^{\infty}$ é denso em X , para qualquer $x \in X$, existe uma subsequência $\{x_{n_j}\}$ tal que $x_{n_j} \rightarrow x$ quando $j \rightarrow \infty$. Pelos Lemas 4.9 e 4.12, segue que $L\tilde{f}(x_{n_j}) = G(x_{n_j}, \tilde{f}_{n_j-1})$.

Portanto, quando $j \rightarrow \infty$, pelo Lema 4.11 e pela continuidade de G , temos que $L\tilde{f}(x) = G(x, \tilde{f})$. Isto é, \tilde{f} é a solução da Equação (4.8).

□

Corolário 4.15. Assuma que \tilde{f} é a solução da Equação (4.8) e r_n é o erro cometido pela aproximação $\tilde{f}_n = \sum_{i=1}^n B_i \Psi_i$. Então, a sequência de erros r_n é monótona decrescente em relação a norma de \mathcal{H}_R .

Demonstração. Suponha que $\tilde{f} = \sum_{i=1}^{\infty} B_i \Psi_i$ e $\tilde{f}_n = \sum_{i=1}^n B_i \Psi_i$. Temos que

$$\begin{aligned} \|r_n\|^2 &= \|\tilde{f} - \tilde{f}_n\|^2 \\ &= \left\| \sum_{i=n+1}^{\infty} B_i \Psi_i \right\|^2 \\ &= \sum_{i=n+1}^{\infty} (B_i)^2. \end{aligned} \quad (4.17)$$

Portanto, segue que o erro r_n é monotônico decrescente em relação a norma de \mathcal{H}_R . \square

A seguir é apresentada uma sequência de passos que sugerimos seguir para a implementação do Teorema 4.14 como algoritmo. Note que pode ser necessário aproximar numericamente $\langle f, g \rangle_R$ quando este envolver integrais e derivadas.

Passo 1. Definir o produto interno e a norma do espaço de reprodução.

Passo 2. Definir n como sendo o número de pontos x_i .

Passo 3. Definir as funções K, G , o operador L e $L^* K_{x_i}$.

Passo 4. Criar o conjunto $\{L^* K_{x_i}\}$ para aplicar ortogonalização.

Passo 5. Calcular as funções ortonormais $\Psi_i(x) = \sum_{j=1}^i \beta_{ji} L^* K_{x_j}$, para todo i .

Passo 6. Definir uma aproximação inicial f_0 e estimar os coeficientes $B_i = \sum_{k=1}^i \beta_{ki} G(x_k, \tilde{f}_{k-1})$.

Passo 7. Calcular a solução aproximada $\tilde{f}_n(x) = \sum_{i=1}^n B_i \Psi_i(x)$, para $x \in X$.

Passo 8. Plotar o gráfico para testar a aproximação.

No próximo capítulo apresentamos o Exemplo 5.3 ilustrando a aplicação deste método.

Observação 4.16. Em [6], os autores apresentam um método iterativo, semelhante ao citado anteriormente, para aproximar a solução de equações diferenciais de Riccati, dadas por

$$\begin{aligned} f'(x) &= r(x)f^2(x) + q(x)f(x) + p(x), & 0 \leq x \leq b, \\ f(0) &= 0 \end{aligned} \quad (4.18)$$

em que r, q e p são funções reais contínuas, $\sup_{x \in [0, b]} \{|q'(x)|, |q(x)|\} < \infty$ e $f \in W_2^2[0, b]$. O autor

reescreve tais equações da forma

$$\begin{cases} L(f)(x) = G(x, f), & x \in [0, b] \\ f(0) = 0 \end{cases} \quad (4.19)$$

em que $G(x, f) = p(x) + r(x)f^2(x)$ e $L : W_2^2[0, b] \rightarrow W_2^1[0, b]$, definido por $Lf(x) = f'(x) - q(x)f(x)$, é um operador linear limitado e, então aplica o processo iterativo.

A linearidade do operador L é imediata. Mostraremos que, de fato, ele é limitado. Reescrevendo o operador como uma soma de outros dois, temos que

$$\begin{cases} L_1(f)(x) = f'(x) \\ L_2(f)(x) = -q(x)f(x) \end{cases}$$

Dessa forma, $\|L(f)\| \leq \|L_1(f)\| + \|L_2(f)\|$.

Primeiramente, note que

$$\begin{aligned} \|L_1(f)\|_K^2 &= \langle L_1(f), L_1(f) \rangle_K \\ &= \langle f', f' \rangle_K \\ &= f'(0)f'(0) + \int_0^1 f''(s)f''(s)ds \\ &\leq f(0)^2 + f'(0)^2 + \int_0^1 f''(s)^2 ds \\ &= \|f\|_R^2 \end{aligned}$$

Logo, pelo Teorema 2.3 segue que L_1 é limitado.

Agora, observe que

$$\begin{aligned} \|L_2(f)\|_K^2 &= \langle L_2(f), L_2(f) \rangle_K \\ &= (L_2(f)(0))^2 + \int_0^1 (L_2(f))'(x)dx \\ &= q(0)^2 f(0)^2 + \int_0^1 (q'(x))^2 (f(x))^2 dx + \int_0^1 (q(x))^2 (f'(x))^2 dx \\ &\leq q(0)^2 f(0)^2 + \max_{x \in [0, 1]} |q'(x)|^2 \int_0^1 (f(x))^2 dx + \max_{x \in [0, 1]} |q(x)|^2 \int_0^1 (f'(x))^2 dx \end{aligned}$$

Como $|f(x)| \leq \|f\|_R \|R_x\|$, sejam $c_1 = \max_{x \in [0, 1]} \{|q'(x)|^2, \|R_x\|^2\}$ e $c_2 = \max_{x \in [0, 1]} \left\{ |q(x)|^2, \left\| \frac{\partial R_x}{\partial y} \right\|^2 \right\}$ e temos que

$$\begin{aligned}\|L_2(f)\|_K^2 &\leq c_1\|f\|_R^2 + c_2\|f\|_R^2 \\ &\leq c\|f\|_R^2, \quad \text{para } c = c_1 + c_2.\end{aligned}$$

Assim, L_2 é limitado e, conseqüentemente, L é um operador linear limitado.

4.2.1.1 Aproximação sem Ortogonalização

Alguns autores afirmam que o processo de ortogonalização de Gram-Schmidt é instável numericamente e tem alto custo computacional. Assim, [27] apresenta uma proposta para aplicação do método de espaços de reprodução, em equações integrais lineares, sem precisar de uma base ortonormal para representar a aproximação da solução. Assim, evita-se a instabilidade numérica da ortogonalização. Isso é reforçado em [7]. Visto isso, buscamos uma adaptação para a proposta de [27] para o caso de equações não lineares, conforme apresentado nos próximos resultados.

Antes apresentamos uma versão técnica a fim de evitar a ortogonalização de $\{L^*K_{x_i}\}$.

Lema 4.17. *Seja $Lf(x) = G(x, f)$. A sequência*

$$f_n(x) = \sum_{i=1}^n \alpha_i \overline{L(K_{x_i})(x_i)}, \quad x \in X,$$

com $\alpha = \mathbf{Ba}$, converge para $f(x)$ em \mathcal{H}_K e uniforme.

Demonstração. A independência linear de $\{L^*K_{x_j}\}_{j=1}^n$, o Lema 4.3 e o Teorema 4.8 produzem as igualdades

$$\begin{aligned}f_n &= \sum_{i=1}^n a_i \Psi_i = \sum_{i=1}^n a_i \sum_{j=1}^i \beta_{ji} L^* K_{x_j} \\ &= \sum_{j=1}^n \left(\sum_{i=j}^n \beta_{ji} a_i \right) L^* K_{x_j} \\ &= \sum_{j=1}^n \alpha_j L^* K_{x_j},\end{aligned}$$

em que $L^*K_{x_j}(x) = \overline{LK_x(x_j)}$, $\alpha = \mathbf{Ba}$, e $\mathbf{a} = [a_1 \ a_2 \ \dots \ a_n]^t$. □

Observação 4.18. *Não concordamos com a igualdade $L(f_n) = f_n$ antes da Expressão (33) em [27], também discordamos da igualdade $L(y_n) = f$ depois da Equação (3.9) em [3]. Notamos*

que a Expressão 4.10 parece não implicar que $\langle L(f_n), \Psi_i \rangle_K = \langle G(\cdot, f), \Psi_i \rangle_K$, como [7] afirma na demonstração do seu Teorema 4.4.

Acreditamos que, em alguns casos, podemos apenas dizer que $\langle L(f_n), \Psi_i \rangle_K \approx \langle G(\cdot, f), \Psi_i \rangle_K$, para n grande o suficiente. Como não concordamos com algumas afirmações de [3, 7, 27], tentamos contornar os pontos de discordância usando argumentos de mínimos quadrados.

Apresentamos agora o teorema principal deste trabalho.

Teorema 4.19. *Se a equação funcional $L(f)(x) = G(x, f)$, para $x \in X$ e $f \in \mathcal{H}_R$, possui uma única solução f e $R = K$, então a sequência*

$$\mathbf{f}_n(x) = \sum_{i=1}^n \alpha_i \overline{L(K_x)(x_i)}, \quad x \in X,$$

converge para $f(x)$ em \mathcal{H}_K e uniformemente, em que $\alpha = \alpha(n)$ é dado pelo sistema

$$\Psi \alpha = \mathbf{G},$$

no sentido de mínimos quadrados, em que $\Psi = [\psi_{ij}]_{n \times n}$, $\psi_{ij} = \langle L(L^* K_{x_j}), L^* K_{x_i} \rangle_K$, $\alpha = [\alpha_1 \ \alpha_2 \ \dots \ \alpha_n]^t$, e $\mathbf{G} = [g_1 \ g_2 \ \dots \ g_n]^t$, com $g_i = \langle G(\cdot, f), L^* K_{x_i} \rangle_K$, para $i, j = 1, \dots, n$.

Demonstração. Vamos fixar $n \in \mathbb{N}$. Usaremos a notação $\alpha = \mathbf{B}\mathbf{a}$ e vislumbres do Lema 4.17. Seja $i = 1, \dots, n$, e note que

$$\|G(\cdot, f) - L(f_n)\|_K = \|L(f) - L(f_n)\|_K \leq \|L\| \|f - f_n\|_K. \quad (4.20)$$

Consequentemente, $\langle L(f_n), \Psi_i \rangle_K \approx \langle G(\cdot, f), \Psi_i \rangle_K$, quando n é grande o suficiente.

Por outro lado, se \mathcal{H}_n é o espaço gerado por $\{L^* K_{x_i}\}_{i=1}^n$. Então a solução do problema de mínimos quadrados

$$\min_{h \in \mathcal{H}_n} \|h - G(\cdot, f)\|_K^2$$

é a projeção ortogonal

$$P_n(G(\cdot, f)) = \sum_{i=1}^n \langle G(\cdot, f), \Psi_i \rangle_K \Psi_i$$

de $G(\cdot, f)$ em \mathcal{H}_n . Isto significa que, se pudéssemos encontrar um vetor $\mathbf{a} \in \mathbb{C}^n$ tal que

$$\left\langle L \left(\sum_{j=1}^n a_j \Psi_j \right), \Psi_i \right\rangle_K = \sum_{j=1}^n a_j \langle L(\Psi_j), \Psi_i \rangle_K$$

esteja tão perto quanto possível de $\langle G(\cdot, f), \Psi_i \rangle_K$, para cada $i = 1, \dots, n$, então poderíamos encontrar uma solução aproximada para a Equação $L(f)(x) = G(x, f)$. Para encontrar este vetor precisamos resolver o problema

$$\sum_{j=1}^n a_j \langle L(\Psi_j), \Psi_i \rangle_K \approx \langle G(\cdot, f), \Psi_i \rangle_K, \quad i = 1, \dots, n,$$

no sentido de mínimos quadrados. Segue do Corolário 4.6 que esta expressão é o mesmo que

$$\sum_{j=1}^n a_j \sum_{k=1}^i \overline{\beta_{ki}} \sum_{l=1}^j \langle L(L^* K_{x_l}), L^* K_{x_k} \rangle_K \beta_{lj} \approx \sum_{k=1}^i \overline{\beta_{ki}} \langle G(\cdot, f), L^* K_{x_k} \rangle_K.$$

Observe ainda que esta expressão é equivalente a

$$\sum_{j=1}^n [\mathbf{B}^* \Psi \mathbf{B}]_{ij} a_j \approx \sum_{k=1}^i \beta_{ki} \langle G(\cdot, f), L^* K_{x_k} \rangle_K,$$

que também é equivalente a

$$\mathbf{B}^* \Psi \mathbf{B} \mathbf{a} \approx \mathbf{B}^* \mathbf{G},$$

em que \mathbf{B}^* é a matriz adjunta de \mathbf{B} . Consequentemente, precisamos resolver a equação

$$\Psi \alpha = \mathbf{G},$$

no sentido de mínimos quadrados, e tomar

$$\mathbf{f}_n = \sum_{j=1}^n \alpha_j L^* K_{x_j}.$$

Sabemos do Teorema 4.8 que os termos da expressão da Equação (4.20) tendem para 0. Então, podemos ver do argumento de mínimos quadrados desta demonstração que $\|L(\mathbf{f}_n) - G(\cdot, f)\|_K$ tende a 0 também. \square

Observamos que os resultados da literatura podem ainda ser melhorados. O próximo resultado nos parece novo. É útil para aplicações em que conhecemos o núcleo K mas não conhecemos \mathcal{H}_K . Ele afirma que não precisamos conhecer o produto interno ou mesmo a estrutura de \mathcal{H}_K . Também é útil para o caso em que o produto interno e a estrutura do espaço de reprodução são complicados (veja [21, 28]).

Corolário 4.20. Se $Lf(x) = G(x, f)$ e $R = K$, então

$$\mathbf{f}_n(x) = \sum_{i=1}^n \alpha_i \overline{L(K_x)(x_i)}, \quad x \in X,$$

converge para $f(x)$ em \mathcal{H}_R e uniforme, em que $\alpha = \alpha(n)$ é dado pelo sistema

$$\Psi \alpha = \mathbf{G},$$

no sentido de mínimos quadrados, em que $\Psi = [\psi_{ij}]_{n \times n}$, $\psi_{ij} = L^2(L^*K_{x_j})(x_i)$, $\alpha = [\alpha_1 \ \alpha_2 \ \dots \ \alpha_n]^t$, e $\mathbf{G} = [g_1 \ g_2 \ \dots \ g_n]^t$, com $g_i = LG(x_i, f)$, para $i, j = 1, \dots, n$.

Demonstração. Note que

$$\langle L(L^*K_x), L^*K_y \rangle_K = \langle L^2(L^*K_x), K_y \rangle_K = L^2(L^*K_x)(y),$$

e que

$$\langle G(\cdot, f), L^*K_x \rangle_K = \langle LG(\cdot, f), K_x \rangle_K = LG(x, f).$$

A demonstração segue do teorema anterior. □

Para implementar esses resultados escrevemos o próximo teorema, o qual nos parece ser um resultado mais geral do que os principais resultados de [3, 27].

Teorema 4.21. Suponha que a equação funcional $Lf(x) = G(x, f)$ tem solução única e que $R = K$. Seja

$$\tilde{\mathbf{f}}_n(x) = \sum_{i=1}^n \alpha_i \overline{L(K_x)(x_i)}, \quad x \in X,$$

em que $\alpha = \alpha(n)$ é dado pelo sistema, possivelmente não linear,

$$\Psi \alpha = \mathbf{G},$$

no sentido de mínimos quadrados, em que $\Psi = [\psi_{ij}]_{n \times n}$, $\psi_{ij} = L^2(L^*K_{x_j})(x_i)$, $\alpha = [\alpha_1 \ \alpha_2 \ \dots \ \alpha_n]^t$, e $\mathbf{G} = [g_1 \ g_2 \ \dots \ g_n]^t$, com $g_i = LG(x_i, \tilde{\mathbf{f}}_n)$, para $i = 1, \dots, n$. Se $\tilde{\mathbf{f}}_n$ converge em \mathcal{H}_K para f , então $Lf = G(\cdot, f)$.

Demonstração. Se $\tilde{\mathbf{f}}_n$ converge para f , então $G(x, \tilde{\mathbf{f}}_n)$ converge para $G(x, f)$. A desigualdade

$$|L(\tilde{\mathbf{f}}_n)(x) - G(x, f)| = |L(\tilde{\mathbf{f}}_n)(x) - L(f)(x)| \leq \|L\| \|\tilde{\mathbf{f}}_n - f\|_K, \quad x \in X,$$

tende para 0. O resultado segue. \square

Vamos descrever brevemente como podemos aplicar o Teorema 4.21 em um contexto numérico. Para tal, vamos fixar $n \in \mathbb{N}$ e definir a função

$$\Phi(\alpha) = |\Psi\alpha - \mathbf{G}(\alpha)|^2, \quad \alpha \in \mathbb{R}^n,$$

em que $\mathbf{G}(\alpha)$ é um vetor de \mathbb{R}^n com a i -ésima coordenada dada por

$$G \left(x_i, \sum_{l=1}^n \alpha_l L^* K_{x_l} \right).$$

Não é difícil ver que

$$\Phi'(\alpha) = 2(\Psi - \mathbf{G}'(\alpha))(\Psi\alpha - \mathbf{G}(\alpha)), \quad \alpha \in \mathbb{R}^n,$$

em que $\mathbf{G}'(\alpha)$ é a matriz jacobiana de $G(\alpha)$ (para detalhes sobre matriz jacobiana veja [29]). Sendo assim, para qualquer escolha de $\alpha_0 \in \mathbb{R}^n$, a curva que dá a direção de maior decréscimo de $\Phi(x)$ é a solução do P.V.I.

$$\begin{cases} \alpha'(t) = -\Phi'(\alpha(t)) \\ \alpha(0) = \alpha_0 \end{cases}$$

Uma maneira fácil de tentar encontrar um mínimo para $\Phi(\alpha)$ é usando um método parecido com o do gradiente (veja [30] para referências sobre problemas de otimização). Basta escolher uma aproximação $\alpha_1 \in \mathbb{R}^n$ e construir a sequência

$$\alpha_{j+1} = \alpha_j - t_j \Phi'(\alpha_j), \quad 0 < t_j < 1, \quad j = 1, 2, \dots \quad (4.21)$$

É claro que outros método numéricos, como Runge Kutta, podem apresentar resultados melhores para resolver o P.V.I.

Apresentamos a seguir uma sequência de passos para a implementação do algoritmo.

Passo 1. Definir n como sendo o número de pontos x_i .

Passo 2. Definir as funções K , G , o operador L , $L^* K_{x_i}$ e uma aproximação inicial.

Passo 3. Preencher a matriz Ψ e o vetor \mathbf{G} .

Passo 4. Definir a função Φ para minimização.

Passo 5. Calcular o gradiente da função Φ .

Passo 6. Calcular a solução aproximada do problema de minimização.

Passo 7. Plotar o gráfico para testar a aproximação.

O Exemplo 5.5, apresentado no próximo capítulo, mostra o funcionamento deste método.

4.2.2 Método de Espaços de Hilbert de Reprodução para Equações Lineares

A motivação para esta seção surgiu da seguinte equação integral linear

$$f(x) + p(x)f(h(x)) + \lambda \int_a^{b(x)} N(x, t)f(t)dt = g(x), \quad (4.22)$$

em que $a \leq x, t \leq b$, N é contínua em $[a, b] \times [a, b]$, λ é uma constante e f, p, h e $g \in W_2^1[a, b]$ são funções contínuas, que começamos a estudar em [27].

Assumimos que a Equação (4.22) possui solução única. Então, sua solução exata e aproximada é dada no espaço $W_2^1[a, b]$.

Podemos reescrever a Equação (4.22) da seguinte forma

$$L(f)(x) = g(x), \quad a \leq x \leq b \quad (4.23)$$

e

$$L : W_2^1[a, b] \longrightarrow W_2^1[a, b]$$

$$f(x) \longmapsto f(x) + p(x)f(h(x)) + \lambda \int_a^{b(x)} N(x, t)f(t)dt$$

é um operador linear limitado. Tomemos o conjunto $\{x_i\}_{i=1}^\infty$ denso em $[a, b]$. Em seguida, determinamos $\{L^*K_{x_i}\}_{i=1}^n$ e, então utilizamos a decomposição QR via processo de ortogonalização de Gram-Schmidt para encontrar as constantes β_{ji} . Consequentemente, temos as funções Ψ_i , para todo $i, j = 1, \dots, n$. Dessa forma, truncando a série dada no Corolário 4.10, podemos encontrar a solução aproximada $f_n(x)$, dada por

$$f_n(x) = \sum_{i=1}^n \sum_{j=1}^i \beta_{ji}g(x_j)\Psi_i(x), \quad x \in X. \quad (4.24)$$

O Exemplo 5.6, no próximo capítulo, ilustra o funcionamento deste método.

Apresentamos agora uma versão do Teorema 4.19 para o caso de equações lineares, como corolário. A demonstração para este caso é semelhante ao caso anterior.

Corolário 4.22. *Assumindo que $Lf(x) = g(x)$ tem solução única f e $R = K$, então*

$$\mathbf{f}_n(x) = \sum_{i=1}^n \alpha_i \overline{L(K_x)(x_i)}, \quad x \in X,$$

converge para $f(x)$ na norma de \mathcal{H}_K e uniformemente, em que $\alpha = \alpha(n)$ é dado pelo sistema linear

$$\Psi \alpha = \mathbf{G},$$

no sentido de mínimos quadrados, com $\Psi = [\Psi_{ij}]_{n \times n}$, $\Psi_{ij} = \langle L(L^ K_{x_j}), L^* K_{x_i} \rangle_K$, $\alpha = [\alpha_1 \ \alpha_2 \ \dots \ \alpha_n]^t$ e $\mathbf{G} = [g_1 \ g_2 \ \dots \ g_n]^t$, com $g_i = \langle g, L^* K_{x_i} \rangle_K$, para $i = 1, \dots, n$.*

Observação 4.23. *Note que o vetor α do Lema 4.17 e do Corolário 4.22 podem ser diferentes. Se este é o caso, então $\alpha(n)$ e $\alpha(n+1)$ podem não estar relacionados. Por outro lado, a sequência f_n no Lema 4.17 nos dá uma boa aproximação para f , e a sequência $L(\mathbf{f}_n)$ no Corolário 4.22 nos dá uma boa aproximação para g . Para mais detalhes veja [17] para resultados sobre a solução de problemas de mínimos quadrados.*

Veremos agora uma sequência de passos sugeridos para a implementação do Corolário 4.22 como algoritmo.

Passo 1. Definir o produto interno e a norma do espaço de reprodução.

Passo 2. Definir os valores para n e x_i .

Passo 3. Definir as funções K , g e o operador L .

Passo 4. Definir as funções $L^* K_{x_i}$ e $L(L^* K_{x_i})$.

Passo 5. Fazer $\Psi_{ij} = \langle L(L^* K_{x_j}), L^* K_{x_i} \rangle_K$, para todo $1 \leq i, j \leq n$.

Passo 6. Preencher o vetor \mathbf{G} , em que $G_i = \langle g, L^* K_{x_i} \rangle_K$, para todo $i = 1, \dots, n$.

Passo 7. Encontrar α .

Passo 8. Definir a função $\mathbf{f}_n(x) = \sum_{i=1}^n \alpha_i \overline{L(K_x)(x_i)}$, para $x \in X$.

Passo 9. Plotar o gráfico para testar a aproximação.

O próximo corolário é o último desse capítulo. É também considerado novo e inclui resultados de [7, 27].

Corolário 4.24. *Se $Lf(x) = g(x)$ e $R = K$, então*

$$\mathbf{f}_n(x) = \sum_{i=1}^n \alpha_i \overline{L(K_x)(x_i)}, \quad x \in X,$$

converge para $f(x)$ em \mathcal{H}_K e uniformemente, em que $\alpha = \alpha(n)$ é dada pelo sistema linear

$$\Psi \alpha = \mathbf{G},$$

*no sentido de mínimos quadrados, com $\Psi = [\Psi_{ij}]_{n \times n}$, $\Psi_{ij} = L^2(L^*K_{x_j})(x_i)$, $\alpha = [\alpha_1 \ \alpha_2 \ \dots \ \alpha_n]^t$, e $\mathbf{G} = [g_1 \ g_2 \ \dots \ g_n]^t$, com $g_i = Lg(x_i)$, para $i = 1, \dots, n$.*

Para a implementação do método segundo o Corolário 4.24, basta trocar os Passos 1, 4, 5 e 6, dados anteriormente, pelos que seguem.

Passo 1. Escolher o núcleo de reprodução K .

Passo 4. Definir a função $L^*K_{x_i}$ e $L^2(L^*K_{x_i})$.

Passo 5. Fazer $\Psi_{ij} = L^2(L^*K_{x_j})(x_i)$, para todo $1 \leq i, j \leq n$.

Passo 6. Preencher o vetor G , em que $G_i = Lg(x_i)$, para todo $i = 1, \dots, n$.

Fizemos a implementação dos métodos conforme o Corolário 4.22 e também conforme o Corolário 4.24, o qual não utiliza o produto interno. Os Exemplos 5.7 e 5.8, do próximo capítulo, mostram o funcionamento do método fazendo e não fazendo uso do produto interno, respectivamente.

5 SIMULAÇÕES NUMÉRICAS

Neste capítulo, são apresentadas as simulações numéricas feitas para ilustrar a teoria estudada nos capítulos anteriores.

Como o objetivo do trabalho está relacionado a estudar o método de resolução, para não delongar muito o trabalho tomamos como exemplos equações presentes na literatura. Dessa forma, não nos preocupamos em mostrar condições de existência e unicidade de soluções. Para tais, podem ser usadas técnicas envolvendo o Teorema do Ponto Fixo de Banach, como feito em [14]. Além disso, pelo mesmo motivo já citado, também não nos preocupamos em mostrar a limitação de todos os operadores utilizados. As principais referências utilizadas para buscar equações foram [3, 6, 27].

Os algoritmos foram implementados em linguagem R [10] e estão no Apêndice. Todos os cálculos necessários foram implementados, como derivadas, integrais, produto interno, decomposição QR com a ortogonalização de Gram-Schmidt modificada, dentre outros, com exceção da inversa generalizada de matrizes.

Primeiramente, mostramos um exemplo da aplicação do método de Nystrom. Este método converge rapidamente para o exemplo tratado, e nos retorna uma aproximação da função aplicada a uma tabela de pontos.

Na sequência exibimos o funcionamento do método de colocação, que é um método de projeção.

Por fim, apresentamos exemplos do funcionamento de todas as propostas do método de espaços de Hilbert de reprodução estudadas.

Estão expostas simulações com apenas um valor de n para cada exemplo. Porém, realizamos simulações com aumento no número de pontos x_i e nos cálculos envolvidos em todos os casos, o que mostrou melhora significativa na precisão dos resultados. Em contrapartida, o custo computacional aumenta muito.

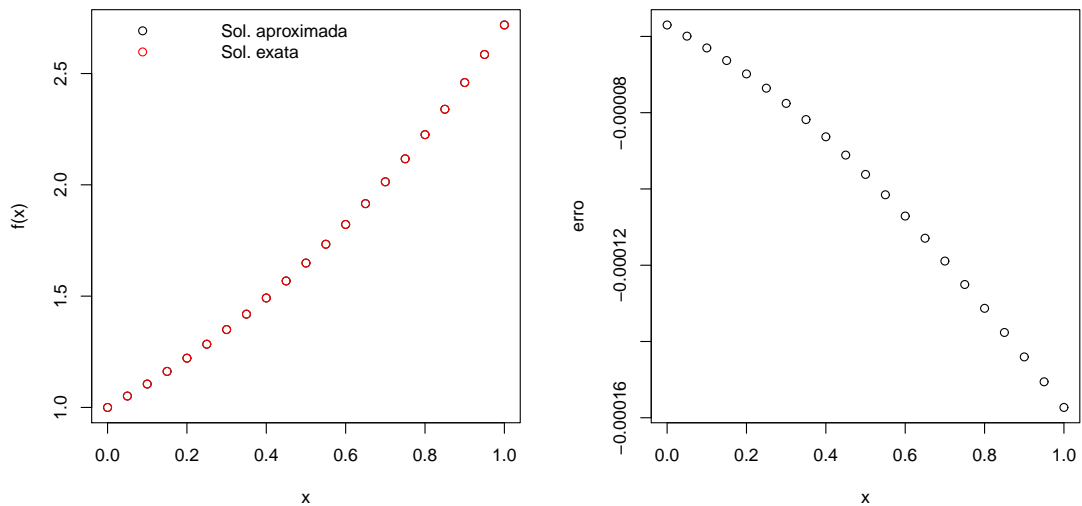
5.1 MÉTODO DE NYSTROM

Apresentamos agora um exemplo da aplicação do método de Nystrom. A descrição do método está exposta no Apêndice. O Algoritmo 7 está adaptado para a equação integral

$$f(x) + \int_0^1 e^{x-2t}(f(t))^3 dt = e^{x+1}, \quad x \in [0, 1]. \quad (5.1)$$

A solução exata desta equação é $f(x) = e^x$ e a aproximação numérica é apresentada na Figura 5.1.

Figura 5.1 – Solução aproximada obtida e erro cometido na aproximação.



Fonte: Da autora.

Foram utilizados 20 pontos e 50 iterações. O erro máximo cometido foi de 0.0001572985 e podemos ver que o método converge rapidamente para este caso. Porém, ele fornece como resposta uma aproximação da função aplicada em alguns pontos específicos, neste exemplo, 20 pontos igualmente espaçados entre 0 e 1.

5.2 MÉTODO DE COLOCAÇÃO

Veremos agora exemplos do método de colocação, descrito na Seção 4.1.

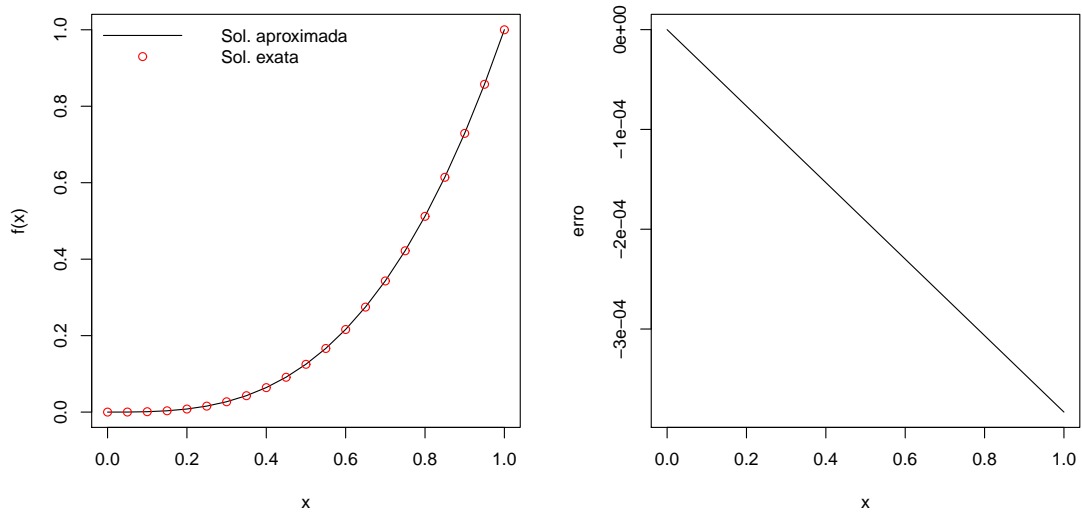
Exemplo 5.1. Considere a equação integral não linear

$$f(x) - \frac{1}{2} \int_0^1 xt^2 \sin(f(t)) dt = x^3 + \frac{x}{6}(\cos(1) - 1), \quad x \in [0, 1], \quad (5.2)$$

que tem solução exata dada por $f(x) = x^3$, $x \in [0, 1]$.

Utilizamos $R(x, y) = 1 + \min(x, y)$, então $\Psi_j(t) = R(x_j, t)$, com 20 pontos $x_j \in [0, 1]$ e fizemos 20 iterações, obtendo um erro máximo de 0.0003832378. Veja a Figura 5.2. Foi utilizado o Código 8, descrito no Apêndice.

Figura 5.2 – Solução aproximada e comportamento do erro.



Fonte: Da autora.

Note que mesmo utilizando um valor baixo para n , os erros cometidos são pequenos. Conforme aumenta-se o valor de n a precisão da solução aproximada aumenta assim como o custo computacional.

Porém, como a maioria dos métodos, o método de colocação não converge sempre. Vejamos o exemplo a seguir.

Exemplo 5.2. Considere a equação integral não linear

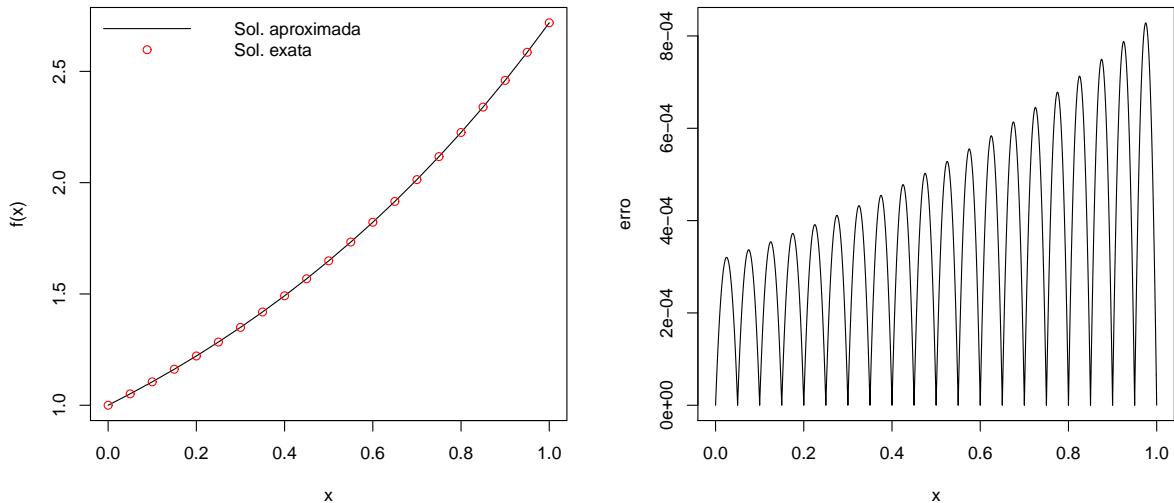
$$f(x) + \int_0^1 e^{x-2y} f^3(y) dy = e^{x+1}, \quad x \in [0, 1] \quad (5.3)$$

que tem solução exata $f(x) = e^x$, $x \in [0, 1]$.

Fazendo as adaptações necessárias no Algoritmo 8, utilizado no Exemplo 5.1, para a Equação (5.3), mantendo o núcleo $R(x, y)$ e as funções $\Psi_j(t)$, o método de colocação diverge.

Porém, adicionando a linha $c = c/\text{aux}(0)$ abaixo da linha $c = z\% * \%t$ no algoritmo, forçamos a condição inicial $f(0) = 1$ (Podemos fazer isto porque conhecemos a solução exata para este exemplo). Deste modo, o algoritmo converge para o Exemplo 5.2. O resultado obtido está exposto na Figura 5.3. O erro máximo foi de 0.0008285329.

Figura 5.3 – Solução aproximada e comportamento do erro.



Fonte: Da autora.

Até o momento, não sabemos explicar, de forma geral, o porquê deste fato ocorrer. Mas a ideia pode ser usada em equações em que sabemos o valor de $f(c)$, para algum c no domínio de $f(x)$ em questão. Todavia, encontramos exemplos em que isso não funcionou.

5.3 MÉTODO DE ESPAÇOS DE HILBERT DE REPRODUÇÃO

Veremos separadamente exemplos do método de espaços de Hilbert de reprodução aplicado a equações lineares e não lineares, conforme descrito nas Seções 4.2, 4.2.1 e 4.2.2.

5.3.1 Equações Não Lineares

Veremos agora um exemplo do funcionamento do método proposto por [6] para equações diferenciais de Riccati. O Algoritmo 12, dado no Apêndice, está adaptado para este exemplo.

Exemplo 5.3. Considere a seguinte equação diferencial não linear de Riccati

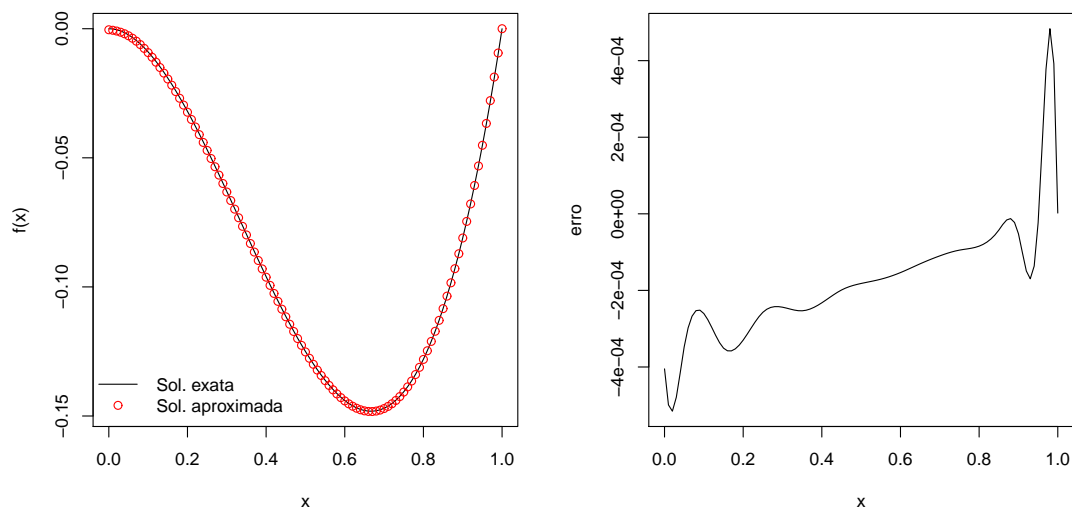
$$\begin{aligned}
 f'(x) &= (x^3 - x^2)f(x) - f^2(x) + 3x^2 - 2x \\
 f(0) &= 0,
 \end{aligned}
 \tag{5.4}$$

que tem solução exata $f(x) = x^3 - x^2$.

Temos, neste caso, que $L(f)(x) = f'(x) - (x^3 - x^2)f(x)$. O núcleo de reprodução usado foi $R_x(y) = y((x-1)y^2 + x(2-3x+x^2+6))/6$, para $1 \geq x \geq y \geq 0$, cujo espaço de Hilbert de reprodução é $W_{2,1}^2[0, 1]$. Usamos $n = 10$ e 20 pontos para os cálculos do produto interno.

Primeiramente, como conhecemos a solução exata da Equação (5.4), usamos esta para calcular os coeficientes de Fourier e então apresentar a aproximação da solução, conforme a Figura 5.4. Desta forma, o erro máximo foi de 0.0005154366.

Figura 5.4 – Solução aproximada obtida e erro cometido na aproximação.

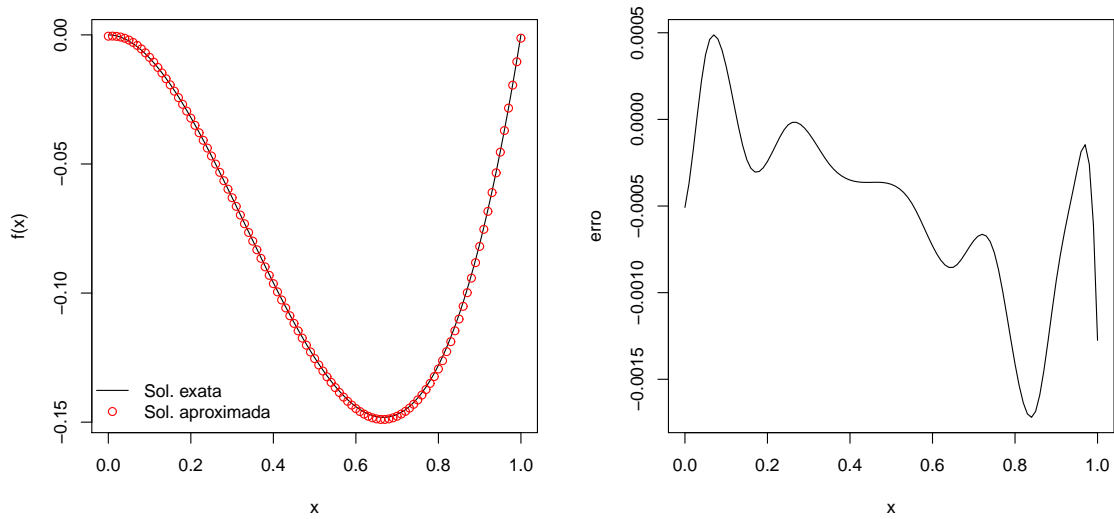


Fonte: Da autora.

Agora, calculando as estimativas dos coeficientes de Fourier, de acordo com a Equação (4.15), obtemos a solução aproximada dada na Figura 5.5. O erro máximo foi de 0.001719727.

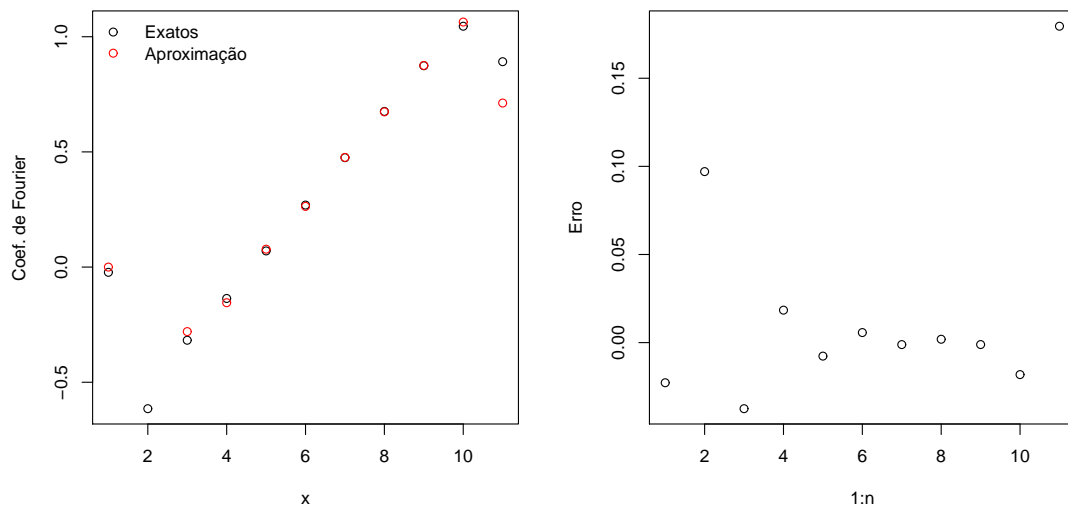
Comparando as estimativas dos coeficientes de Fourier, obtidas por meio da Equação (4.15), com os calculados através da solução exata conhecida, obtemos os resultados da Figura 5.6.

Figura 5.5 – Solução aproximada obtida e erro cometido na aproximação.



Fonte: Da autora.

Figura 5.6 – Estimativas dos coeficientes de Fourier e erro cometido.



Fonte: Da autora.

Temos que o erro máximo foi de 0.1795203. Note que esses coeficientes apresentam menor precisão nos pontos dos extremos, ocorrendo neste caso um efeito parecido com o fenômeno de Runge da interpolação polinomial.

Observação 5.4. *O operador L tratado no exemplo anterior é um operador linear limitado.*

A linearidade de L é imediata. Vamos mostrar que L é limitado.

De fato, vamos reescrever o operador como uma soma de outros dois, da seguinte forma

$$\begin{cases} L_1(f) = f' \\ L_2(f) = -(x^3 - x^2)f \end{cases}$$

Dessa forma, segue que $\|L(f)\| \leq \|L_1(f)\| + \|L_2(f)\|$.

A limitação de L_1 segue de modo análogo ao feito na Observação 4.16.

Agora, observe que

$$\begin{aligned} \|L_2(f)\|_K^2 &= \langle L_2(f), L_2(f) \rangle_K \\ &= \underbrace{(L_2(f)(0))^2}_{=0} + \int_0^1 (L_2(f))'(x) dx \\ &= 0 + \int_0^1 (3x^2 - 2x)^2 (f(x))^2 dx + \int_0^1 (x^3 - x^2)^2 (f'(x))^2 dx \\ &\leq \int_0^1 (f(x))^2 dx + \int_0^1 (f'(x))^2 dx \\ &\leq \max_{x \in [0,1]} \|R_x\|^2 \|f\|_R^2 + \max_{x \in [0,1]} \left\| \frac{\partial R_x}{\partial y} \right\|^2 \|f\|_R^2 \\ &\leq c \|f\|_R^2. \end{aligned}$$

Portanto, temos que L_2 é limitado e, conseqüentemente, L é um operador linear limitado.

Agora, apresentamos um exemplo do funcionamento do método baseado no Teorema 4.21. O Algoritmo 13 foi implementado para este exemplo.

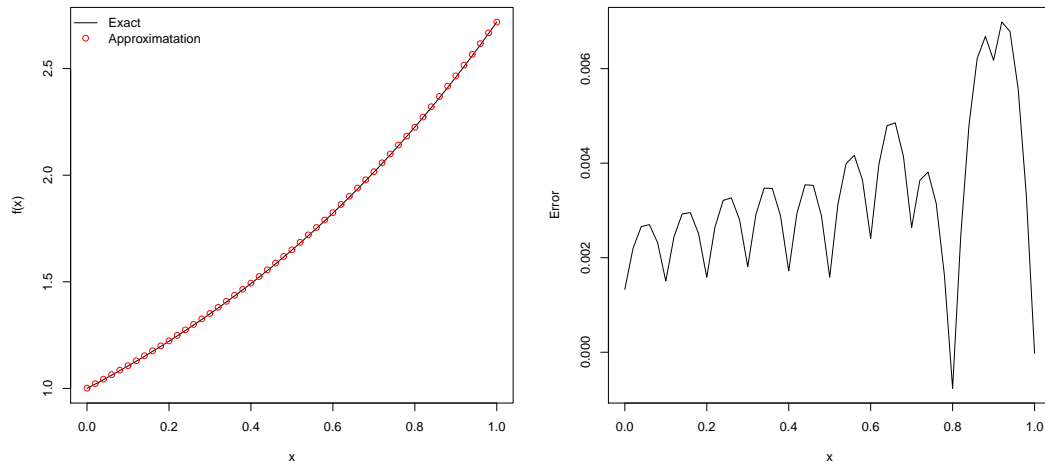
Exemplo 5.5. *Considere a equação integral não linear apresentada no Exemplo 5.2.*

Usamos o núcleo de reprodução $K(x, y) = 1 + \min(x, y)$, e notamos que a ij -ésima coordenada da matriz do jacobiano $\mathbf{G}'(\alpha)$ é

$$-3 \int_0^1 e^{x_i - 2u} \left(\sum_{l=1}^n \alpha_l L^* K_{x_l}(u) \right)^2 L^* K_{x_j}(u) du.$$

A solução exata é dada por $f(x) = e^x$. Com $n = 11$ e $x_i = 0.1i$, para $0 \leq i \leq 10$ obtemos o resultado exposto na Figura 5.7.

Figura 5.7 – Solução aproximada obtida e erro cometido.



Fonte: Da autora.

5.3.2 Equações Lineares

Segue um exemplo da utilização de base ortonormal em um espaço de Hilbert de reprodução. Numericamente, fazemos a decomposição QR por meio do processo de ortogonalização de Gram-Schmidt, como tratado no Corolário 4.10.

Exemplo 5.6. *Considere a equação integral linear*

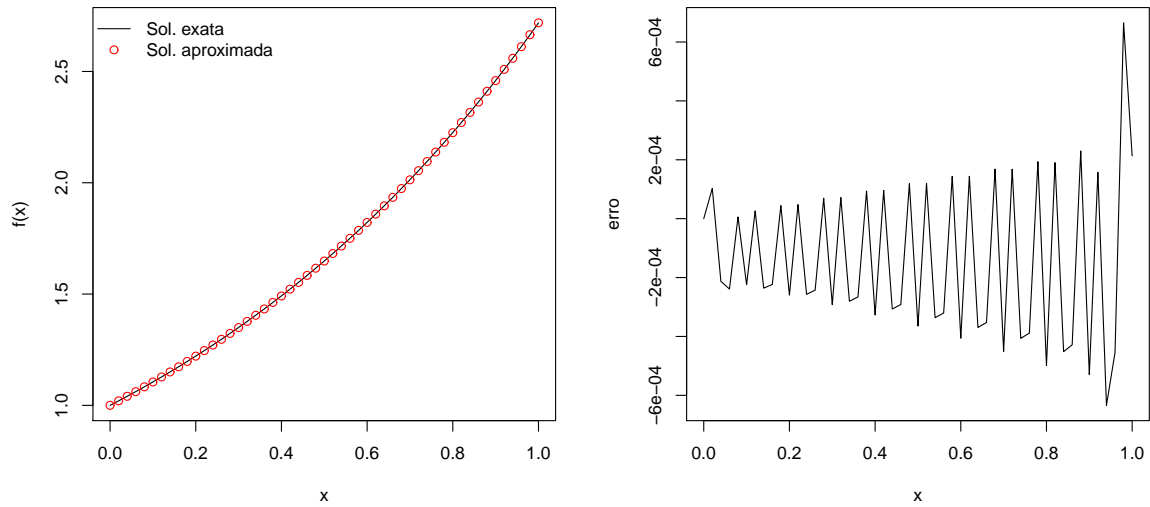
$$f(x) + \int_0^1 xsf(s)ds = e^x + x, \quad x \in [0, 1]. \quad (5.5)$$

que tem solução exata $f(x) = e^x$, para $x \in [0, 1]$.

Utilizando o Algoritmo 9, apresentado no Apêndice, obtemos a Figura 5.8, com $n = 20$ e 50 pontos para a construção do gráfico. O erro máximo cometido na aproximação foi de 0.0006649314. Neste exemplo, temos que $L(f)(x) = f(x) + \int_0^1 xsf(s)ds$ e $g(x) = e^x + x$. O núcleo utilizado foi $R(x, y) = 1 + \min(x, y)$ e o espaço de reprodução tratado é $W_2^1[0, 1]$.

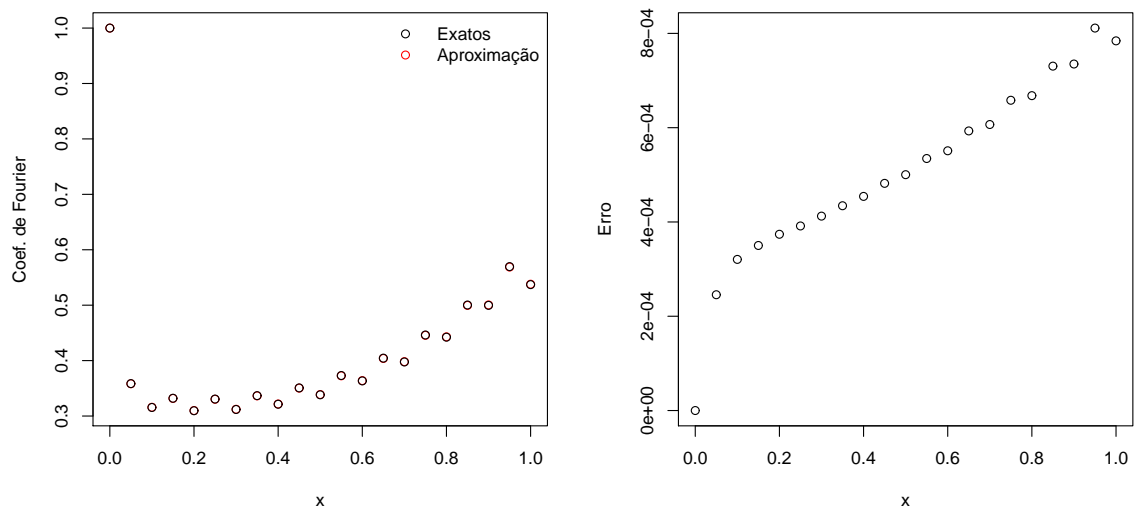
Como conhecemos a solução exata deste exemplo, podemos comparar as estimativas dos coeficientes de Fourier obtidas por este método com os calculados por meio da identidade de Parseval, utilizando a solução exata da equação. Note que, como mostra a Figura 5.9, o erro cometido é pequeno.

Figura 5.8 – Solução aproximada obtida e erro cometido na aproximação.



Fonte: Da autora.

Figura 5.9 – Comparação dos coeficientes de Fourier e erro.



Fonte: Da autora.

Observe que a precisão dos coeficientes de Fourier do caso linear é maior do que do caso não linear. Isso ocorre porque as estimativas utilizadas no método linear não dependem da solução exata da equação, como ocorre no caso das equações não lineares.

Veremos um exemplo do funcionamento do método proposto por [27], conforme o Corolário 4.22, sem o processo de ortogonalização.

Exemplo 5.7. Considere a equação integral linear

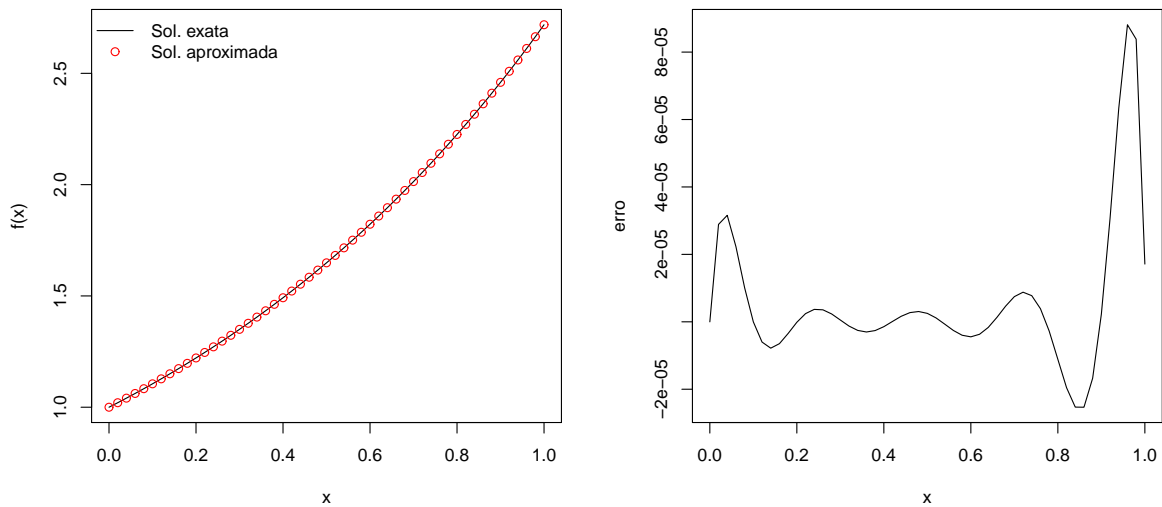
$$f(x) + e^{-x} f(\sin(x)) + \int_0^1 2e^{x-u} f(u) du = 3e^x + e^{\sin(x)-x}, \quad x \in [0,1],$$

cujas soluções exatas são $f(x) = e^x$.

Utilizamos aqui o núcleo gaussiano, $R(x,y) = e^{-4(x-y)^2}$, $x, y \in \mathbb{R}$, e $n = 10$ pontos $x_i \in [0, 1]$ para os cálculos. O resultado obtido está exposto na Figura 5.10.

Com este algoritmo, conseguimos de maneira mais rápida um resultado melhor do que o apresentado por [27], com $n = 100$, mesmo sem determinar \mathcal{H}_K e $\langle \cdot, \cdot \rangle_K$.

Figura 5.10 – Solução aproximada obtida e erro cometido na aproximação.



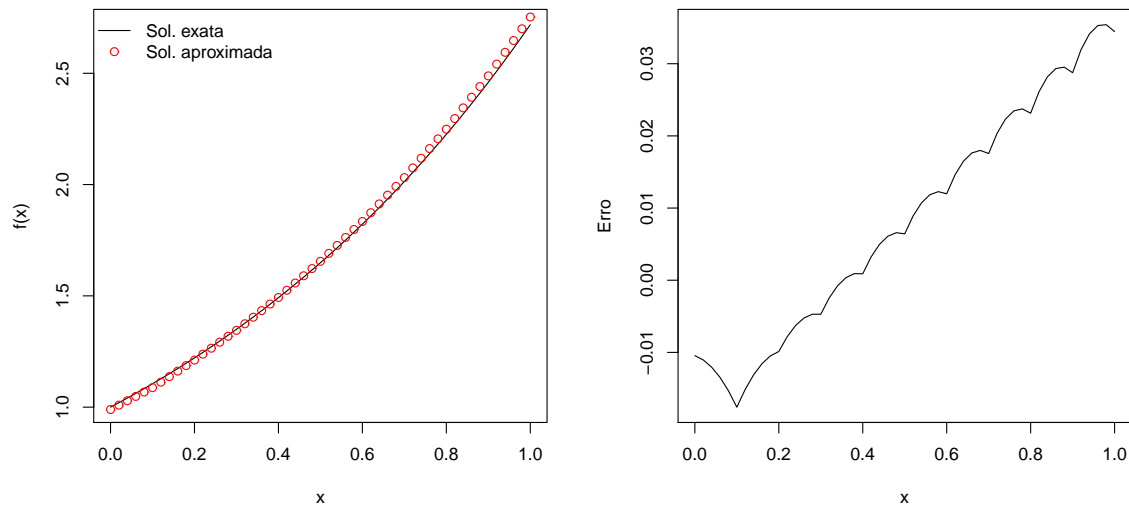
Fonte: Da autora.

Exemplo 5.8. Considere a equação dada no Exemplo (5.6).

O Algoritmo 10, dado no Apêndice, foi implementado para este exemplo, sem utilizar ortogonalização. Utilizamos integração numérica pela regra dos trapézios e o mesmo núcleo dado no Exemplo 5.6. Adotamos $n = 10$ e o erro máximo foi de 0.03548214, como mostra a Figura 5.11.

Note que, embora tenha sido utilizado $n = 10$, podemos apresentar a solução aproximada em quantos pontos quisermos. Isso é possível devido ao fato deste método obter como resposta uma função, a qual é aproximadamente a solução exata do problema, e não uma tabela de pontos, como é o caso do método de Nystrom, que já foi discutido.

Figura 5.11 – Solução aproximada obtida e erro cometido na aproximação.



Fonte: Da autora.

Também implementamos um algoritmo baseado no Corolário 4.24, dado pelo Algoritmo 11 do Apêndice. Este método se mostrou mais rápido, pois não faz uso de produto interno.

Note nas simulações que a suavidade do erro das aproximações está relacionada com a suavidade do núcleo de reprodução utilizado. Isso se deve ao fato de que, conforme o Corolário 3.17, dependendo da suavidade do núcleo, além da aproximação f_n convergir para a solução exata f , as derivadas de f_n também convergem para as derivadas de f .

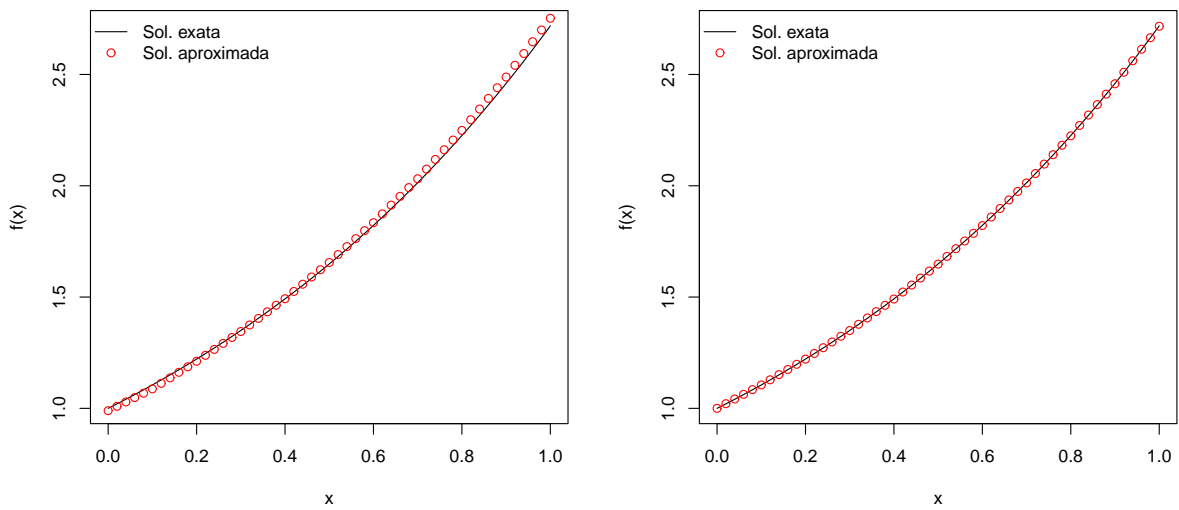
Para comparar o método baseado no Corolário 4.22 com o dado pelo Corolário 4.24, vamos considerar novamente a equação integral linear dada no Exemplo 5.8

$$f(x) + \int_0^1 xsf(s)ds = e^x + x, \quad x \in [0,1]. \quad (5.6)$$

A Figura 5.12 (a) mostra a aproximação feita pelo método utilizando o produto interno do espaço, baseado no Corolário 4.22, e a Figura 5.12 (b) a aproximação calculada sem usar o produto interno, conforme o método dado pelo Corolário 4.24.

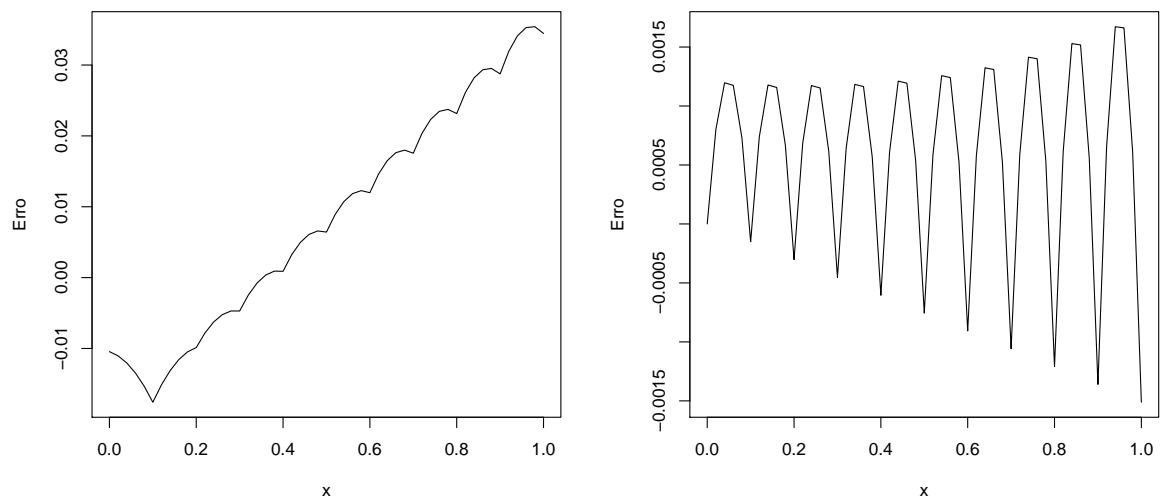
Com um olhar rápido já podemos perceber que a aproximação calculada sem utilizar o produto interno é melhor do que a primeira. A Figura 5.13 apresenta os erros cometidos nas duas aproximações.

Figura 5.12 – Soluções baseadas no Corolário 4.22 (a) e no Corolário 4.24 (b).



Fonte: Da autora.

Figura 5.13 – Erros dos métodos baseados no Corolário 4.22 (a) e no Corolário 4.24 (b).



Fonte: Da autora.

Além de mais preciso, o procedimento que não faz uso do produto interno é mais rápido, como mostra a Tabela 5.1.

Tabela 5.1 – Comparação dos métodos com e sem o uso de produto interno.

Método com prod. interno		Método sem prod. interno		
n	Tempo (s)	Erro	Tempo (s)	Erro
10	58.28	0,03540921	29.97	0,001672325
20	236.51	0.05757165	105.29	0.0006929453

Fonte: Da autora.

6 RESULTADOS E DISCUSSÕES

Durante o desenvolvimento do trabalho, tentamos estudar e aplicar diversas propostas do método de espaços de Hilbert de reprodução. Todavia, não conseguimos bons resultados com a feita por [3]. Os autores apresentam uma maneira alternativa para resolver equações integrais não lineares sem utilizar o processo de ortogonalização, pois afirmam que este é instável numericamente. Acreditamos que a falha esteja relacionada com o fato de que os operadores tratados pelos autores não são claramente lineares, como afirmam. O trabalho deles trata de equações integrais não lineares que podem ser escritas da forma

$$L(f)(x) = f(x), \quad (6.1)$$

em que $L(f)(x) = g(x) - \int_a^{b(x)} N(x,t)D(f(t))dt$ e D é uma função contínua. Assim, dependendo da função D , o operador L é não linear, o que não satisfaz as condições da teoria estudada.

Assim como [3], outros trabalhos da literatura afirmam que o processo de ortogonalização de Gram-Schmidt é instável numericamente. Apesar disto, a proposta testada utilizando uma base ortonormal, ilustrada no Exemplo 5.6, mostrou-se mais eficiente e mais rápida do que a de [27] na resolução de equações funcionais lineares. Talvez isso tenha ocorrido pela forma como o processo de ortogonalização foi programado, por particularidades dos exemplos escolhidos e o número de pontos utilizado. Além disso, usamos o método de Gram-Schmidt modificado em nossas implementações e, na literatura, como mostra [3], os autores parecem utilizar o método clássico que é bem mais instável numericamente.

Conseguimos adaptar a proposta feita por [27] para aproximar também soluções de equações não lineares, como descrito na seção 4.2.1.1, e ela forneceu bons resultados, como mostrado no Exemplo 5.5. Além disso, conseguimos melhorar o método tanto para equações lineares quanto para não lineares, simplificando os cálculos e desprezando o uso do produto interno, como afirmado nos Corolários 4.20 e 4.24.

Nossa proposta, dada pelo Corolário 4.24, mostrada no Exemplo 5.7, fornece uma aproximação mais precisa e de maneira mais rápida, visto que não é necessário utilizar o produto interno. Embora, não tenhamos exibido, o mesmo ocorre para o caso não linear.

Os Corolários 4.20 e 4.24 ao nosso ver são resultados novos e irão contribuir para a aplicação dos métodos de espaços de Hilbert de reprodução, pois asseguram que basta partir

de um núcleo positivo definido, ou seja, não precisamos conhecer o produto interno e nem a estrutura do espaço em que estamos trabalhando. Este fato traz vantagens, visto que a maioria dos trabalhos que aplicam os métodos de espaços de Hilbert de reprodução gastam tempo e espaço demasiadamente grandes apenas para descrever os espaços de reprodução em que estão trabalhando e, em alguns casos, descrever a estrutura do espaço é uma tarefa bastante complexa.

Como possíveis desdobramentos deste trabalho, consideramos que as propostas apresentadas tanto para equações lineares, como para equações não lineares (com exceção da feita por [3]) podem ser adaptadas para aproximar a solução de sistemas de equações funcionais.

REFERÊNCIAS

- [1] BASSANEZI, R. C. **Ensino-aprendizagem com modelagem matemática**. 3. ed. São Paulo: Contexto, 2006.
- [2] ARQUB, O. A.; AL-SMADI, M.; MOMANI, S. Application of reproducing kernel method for solving nonlinear Fredholm-Volterra integrodifferential equations. **Abstract and Applied Analysis**, v. 2012, 16 p., 2012.
- [3] JAVAN, S. F.; ABBASBANDY, S.; ARAGHI, M. A. F. Application of Reproducing Kernel Hilbert Space Method for Solving a Class of Nonlinear Integral Equations. **Mathematical Problems in Engineering**, v. 2017, 10 p., 2017.
- [4] FERREIRA, J. C.; FERREIRA, E. C. On Reproducing Kernel and Applications. **Advances in Analysis**, v. 3, n. 1, p. 11-22, 2018.
- [5] AZARNAVID, B.; PARAND, K. An iterative reproducing kernel method in Hilbert space for the multi-point boundary value problems. **Journal of Computational and Applied Mathematics**, v. 328, p. 151-163, 2018.
- [6] SAKAR, M. G. Iterative reproducing kernel Hilbert spaces method for Riccati differential equations. **Journal of Computational and Applied Mathematics**, v. 309, p. 163-174, 2017.
- [7] VAHDATI, S.; FARDI, M.; GHASEMI, M. Option pricing using a computational method based on reproducing kernel, **Journal of Computational and Applied Mathematics**, v. 328, p. 252-266, 2018.
- [8] INC, M.; AKGÜL, A.; KILIÇMAN, A. Explicit solution of telegraph equation based on reproducing kernel method. **Journal of Function Spaces and Applications**, v. 2012, 23 p., 2012.
- [9] SAKAR, M. G.; AKGÜL, A.; BALEANU, D. On solutions of fractional Riccati differential equations. **Advances in Difference Equations**, v. 2017, n. 1, 2017.
- [10] R, Core Team. **R: A language and environment for statistical computing**, R Foundation for Statistical Computing, Vienna, Áustria, 2016. Disponível em: <<http://www.R-project.org/>>. Acesso em: 10 jul. 2018.
- [11] BUENO, H. P. **Álgebra linear**: um segundo curso. Rio de Janeiro: SBM, 2006.

- [12] COELHO, F. U.; LOURENÇO, M. L. **Um curso de álgebra linear**. 2. ed. São Paulo: Edusp, 2013. v. 34.
- [13] FOLLAND, G. B. **Real analysis: modern techniques and their applications**, New York: John Wiley e Sons, 1999.
- [14] OLIVEIRA, C. R. **Introdução à análise funcional**, Rio de Janeiro: IMPA, 2015.
- [15] FERREIRA, E. C. **Espaços de Hilbert de reprodução e aproximação de soluções de equações integrais de Volterra**. 2016. 104f. Dissertação (Mestrado em Estatística Aplicada e Biometria) - Instituto de Ciências Exatas, Universidade Federal de Alfenas, Alfenas, 2016.
- [16] FRANCO, N. B. **Cálculo numérico**. São Paulo: Pearson Prentice Hall, 2006.
- [17] WEBB, M. D. **Isospectral algorithms, toeplitz matrices and orthogonal polynomials**. 2017. 264f. Tese (Doutorado em Filosofia) - Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, 2017.
<https://doi.org/10.17863/CAM.9505>
- [18] FERREIRA, J. C.; MENEGATTO, V. A. Reproducing properties of differentiable Mercer-like kernels. **Mathematische Nachrichten**, v. 285, n. 8-9, p. 959-973, 2012.
- [19] BERG, C.; CHRISTENSEN, J. P. R.; RESSEL, P. **Harmonic analysis on semigroups**, Berlim: Springer Verlag, 1984.
- [20] FERREIRA, J. C. **Operadores integrais positivos e espaços de Hilbert de reprodução**. 2010. 121f. Tese (Doutorado em Ciências Matemáticas) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2010.
- [21] FASSHAUER, G. E. Positive definite kernels: past, present and future. **Dolomites Research Notes on Approximation**, v. 4, p. 21-63, 2011.
- [22] YANG, L.; LIN, Y. Reproducing kernel methods for solving linear initial-boundary-value problems. **Electronic Journal of Differential Equations (EJDE)[electronic only]**, v. 2008, n. 29, 11 p., electronic only-Paper, 2008.
- [23] GUECHI, S. **Méthodes computationnelles pour la résolution des équations intégrales non linéaires**. 2017. 86f. Tese (Doutorado em Matemática) - Université Mohamed Boudiaf, M'Sila, 2017.
- [24] WANG, Y.; ZHU, L. Solving nonlinear Volterra integro-differential equations of fractional order by using Euler wavelet method. **Advances in Difference Equations**, v. 2017, n. 1, p. 27, 2017.
- [25] O'LEARY, D. P.; WHITMAN, P. Parallel QR factorization by householder and modified Gram-Schmidt algorithms. **Parallel Computing**, v. 16, n. 1, p. 99-112, 1990.
- [26] ARQUB, O. A.; AL-SMADI, M.; SHAWAGFEH, N. Solving Fredholm integro-differential

equations using reproducing kernel Hilbert space method. **Applied Mathematics and Computation**, v. 219, n. 17, p. 8938-8948, 2013.

- [27] JAVADI, S.; BABOLIAN, E.; MORADI, E. New implementation of reproducing kernel Hilbert space method for solving a class of functional integral equations. **Communications in Numerical Analysis**, v. 2014, p. 1-7, 2014.
- [28] MINH, H. Q. Some Properties of Gaussian Reproducing Kernel Hilbert Spaces and Their Implications for Function Approximation and Learning Theory, **Constructive Approximation**. v. 32, n. 2, p. 307-338, 2010.
<https://doi.org/10.1007/s00365-009-9080-0>
- [29] LIMA, E. L. Análise real. Rio de Janeiro: IMPA, 2004. v. 2.
- [30] Limei Zhou, Yue Wu, Liwei Zhang, Guang Zhang, Convergence analysis of a differential equation approach for solving nonlinear programming problems, **Applied Mathematics and Computation**, v. 184, n. 2, p. 789-797, 2007.
<https://doi.org/10.1016/j.amc.2006.05.190>.

APÊNDICE

Neste capítulo apresentaremos os algoritmos utilizados nas simulações numéricas. As implementações foram feitas em linguagem R [10]. Todos os cálculos necessários para os métodos implementados, como derivadas, integrais, produto interno, decomposição QR com a ortogonalização de Gram-Schmidt modificada, dentre outros, foram implementados. Utilizamos apenas um pacote para calcular matriz inversa, chamado `pracma` e a função `pinv()`. Esta função calcula a inversa generalizada de matrizes através da decomposição de valores singulares da matriz e, algoritmos deste tipo são bastante sofisticados para o escopo deste trabalho.

Escolhemos não fazer uso de outros pacotes prontos do software com o objetivo de contribuir com a formação da autora em análise numérica, ampliando suas possibilidades do uso de tecnologias ao exercer sua profissão. Além disso, ao implementar cada parte do algoritmo é possível perceber onde aparecem os problemas e temos mais flexibilidade para tentar solucioná-los.

Com objetivo de encurtar o texto, a parte de construção de gráficos foi omitida de alguns algoritmos. Além disso, nos algoritmos em que o comando `pinv()` é utilizado, primeiramente é necessário carregar o pacote `pracma`, através do comando `require(pracma)`.

Algoritmos Básicos

Nesta seção apresentamos uma breve fundamentação teórica e a implementação feita para algoritmos de derivadas, integrais e produto interno. Além disso, apresentamos ainda o algoritmo para a decomposição QR.

As implementações expostas nesta seção são fundamentais para a elaboração dos algoritmos do método de espaços de Hilbert de reprodução, os quais serão apresentados mais adiante.

Derivação e Integração Numéricas

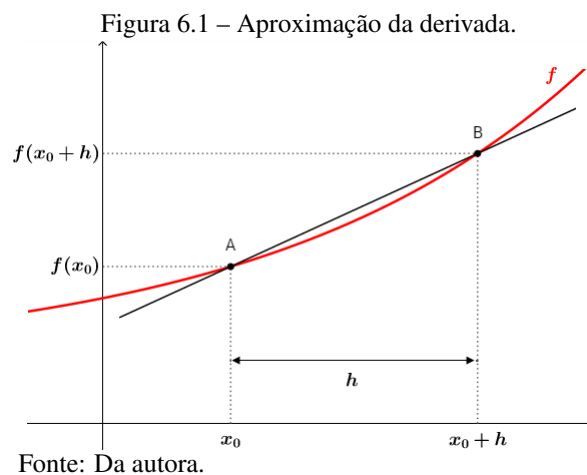
Em muitos casos, necessitamos calcular derivadas e integrais de funções as quais não conhecemos sua forma analítica, ou seja, conhecemos apenas seu valor em um determinado conjunto de pontos. Também ocorre, em certas ocasiões, da forma analítica da função ser extremamente complicada e de difícil manuseio. Nesses casos, torna-se útil conhecer métodos

numéricos de derivação e integração.

Na implementação dos algoritmos foram utilizados métodos de derivação e integração numéricos. Uma vez que os comandos prontos para derivação não funcionam para alguns algoritmos do modo como foram implementados. Além disso, o comando pronto para integração na maioria dos casos mostrou-se inviável, pois tornava os algoritmos muito lentos.

Derivação Numérica

Considere uma função f e um dado ponto A sobre o gráfico de f . A reta tangente em A pode ser aproximada pela reta AB na medida em que B se aproxima de A , onde B é um segundo ponto na curva. Conforme mostra a Figura 6.1.



Temos que o coeficiente da reta AB pode ser calculado como segue.

$$m = \frac{f(x_0 + h) - f(x_0)}{h}.$$

Esse valor é chamado de derivada numérica. Além disso, a derivada de f no ponto A é dada pelo coeficiente da reta tangente ao ponto A . Pela Figura 6.1, podemos concluir que quanto menor h , mais próximo B estará de A , então a reta AB se aproxima da reta tangente à A . Logo,

$$f'(x_0) = \lim_{h \rightarrow 0} \frac{f(x_0 + h) - f(x_0)}{h}.$$

Conforme h diminui, o valor da derivada numérica se aproxima do valor real da derivada. Contudo, por menor que seja h , o método ainda pode apresentar erros grandes de arredondamento. Um modo de tentar reduzir estes erros é utilizar mais pontos.

A ideia para o método é dividir o intervalo $[a,b]$ em n pontos x_1, \dots, x_n distintos e igualmente espaçados por uma distância h . Em seguida, devemos interpolar esse conjunto de pontos, encontrando uma função polinomial $p(x)$ que os representa. Assim, podemos calcular a derivada da função $p(x)$ e aplicá-la a qualquer ponto $x_i \in [a,b]$. Quanto maior o número de pontos, melhor será a aproximação.

Considere n pontos x_1, \dots, x_n distintos em $[a,b]$ e $f \in \mathcal{C}^{n+1}([a,b])$, em que $f(x_1), \dots, f(x_n)$ são conhecidos. Para $k = 1, 2, \dots, n$, dados os seguintes polinômios de grau n :

$$p_i(x) = \frac{(x - x_0) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)}{(x_i - x_0) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)},$$

temos que o polinômio

$$P_n(x) = \sum_{i=1}^n f(x_i)p_i(x) \quad (6.2)$$

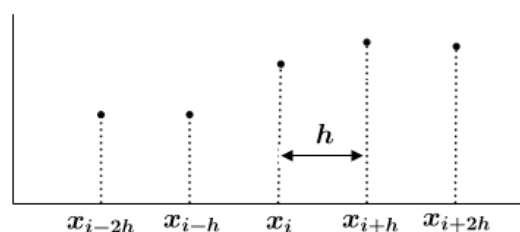
é de grau no máximo n e satisfaz $P_n(x_i) = f(x_i)$, para todo $i = 1, 2, \dots, n$. A fórmula dada na Equação (6.2) é chamada de fórmula de Lagrange do polinômio de interpolação. Para mais detalhes sobre o polinômio de Lagrange veja [16].

Derivando a expressão dada na Equação (6.2), temos que

$$f'(x) \simeq \sum_{i=1}^n f(x_i)p'_i(x). \quad (6.3)$$

Por praticidade, costumamos utilizar fórmulas que dependem de três a cinco pontos. Vejamos a fórmula para três pontos. Sejam os pontos x_1, \dots, x_n distintos e igualmente separados por uma distância h . Se queremos determinar a derivada de f em um certo ponto $x_i \in \{x_1, \dots, x_n\}$, então temos três possibilidades, como ilustra a Figura 6.2.

Figura 6.2 – Ilustração para a regra de três pontos.



Fonte: Da autora.

Podemos escolher três pontos da seguinte forma:

1. diferenças finitas retroativas: x_{i-2h}, x_{i-h}, x_i ;
2. diferenças finitas centrais: x_{i-h}, x_i, x_{i+h} ;
3. diferenças finitas progressivas: x_i, x_{i+h}, x_{i+2h} .

Substituindo os valores para os pontos na Equação (6.3) para $n = 3$, obtemos as fórmulas que seguem.

1. para diferenças finitas retroativas:

$$f'(x_i) \simeq \frac{1}{2h} [-3f(x_i) + 4f(x_i + h) - f(x_i + 2h)]; \quad (6.4)$$

2. para diferenças finitas centrais:

$$f'(x_i) \simeq \frac{1}{2h} [f(x_i + h) - f(x_i - h)]; \quad (6.5)$$

3. para diferenças finitas progressivas:

$$f'(x_i) \simeq \frac{1}{2h} [f(x_i - 2h) - 4f(x_i - h) + 3f(x_i)]. \quad (6.6)$$

O Algoritmo 1 a seguir é um exemplo de como as fórmulas de três pontos foram implementadas e utilizadas em outros algoritmos.

Algoritmo 1. *Fórmulas de três pontos para aproximação da derivada.*

```
D=matrix(0, m, m)
D[1, 1]=-3/2; D[1, 2]=2; D[1, 3]=-1/2
for ( i in 2:(m-1) ) {
D[i, i-1]=-1/2; D[i, i+1]=1/2}
D[m, m-2]=1/2; D[m, m-1]=-2; D[m, m]=3/2; D=D/h
```

Assim, para calcular a derivada numérica de $f(x)$ nos pontos x_1, \dots, x_n definimos o vetor $F = [f(x_1) \dots f(x_n)]^T$ e, então calculamos $DF \simeq f'(x_i)$.

Outra fórmula bastante utilizada é a de cinco pontos, dada por

$$f'(x_0) \simeq \frac{1}{12h} [f(x_0 - 2h) - 8f(x_0 - h) + 8f(x_0 + h) - f(x_0 + 2h)],$$

em que $h > 0$.

O algoritmo para esta fórmula, para função não tabelada e com domínio contendo $s - 2h, s - h, s + h$ e $s + 2h$, foi implementado e é dado a seguir.

Algoritmo 2. *Fórmula de cinco pontos para aproximação da derivada.*

```
D<-function(f, s) { h=10^{-5}
p=(-f(s+2*h)+8*f(s+h)-8*f(s-h)+f(s-2*h))/(12*h); p}
```

Para utilizar o algoritmo dado anteriormente, primeiro é necessário definir a função que desejamos derivar. Veremos a seguir um exemplo.

Exemplo 0.1. *Seja $f(x) = \cos(x) + 1$. Obtenha $f'(5)$.*

Sabemos que $f'(x) = -\text{sen}(x)$. Logo, $f'(5) = -\text{sen}(5) = 0.9589243$.

Utilizando o Algoritmo 2, primeiramente definimos $f <- \text{function}(s)\{\cos(s) + 1\}$ e obtemos $f'(5) = 0.9589243 \simeq -\text{sen}(5)$, por meio do comando $D(f,5)$.

Este método é preciso e rápido computacionalmente. Porém, o Algoritmo 1 nos retorna valores pontuais e não uma função, como o Algoritmo 2. Como em nossos algoritmos precisamos apenas de valores pontuais, ele se mostrou eficiente e mais leve.

Integração Numérica

Nesta seção descrevemos o método de integração conhecido como regra dos trapézios repetida. Considere a seguinte integral definida

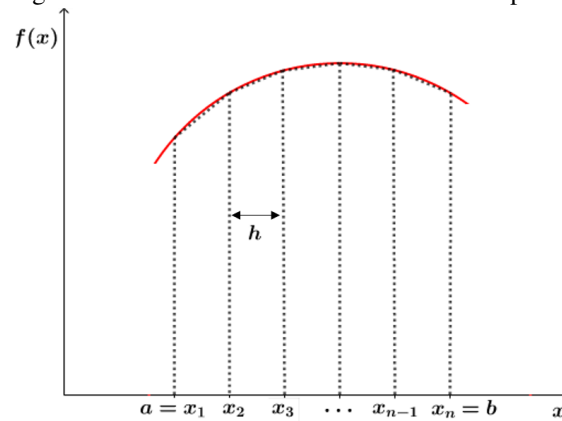
$$\int_a^b f(x)dx. \quad (0.1)$$

A ideia do método é dividir o intervalo de integração $[a, b]$ em n pontos $a = x_1, x_2, \dots, x_{n-1}, x_n = b$ distintos e igualmente espaçados por uma distância h e, calcular a integral dada na Equação (0.1) como

$$\int_a^b f(x)dx = \int_{a=x_1}^{x_2} f(x)dx + \int_{x_2}^{x_3} f(x)dx + \dots + \int_{x_{n-1}}^{x_n=b} f(x)dx. \quad (0.2)$$

Cada parcela $\int_{x_i}^{x_{i+1}} f(x)dx$ da soma dada na Equação (0.2) pode ser aproximada pela área de um trapézio de bases $f(x_i)$ e $f(x_{i+1})$ e altura h , como mostra a Figura 0.1.

Figura 0.1 – Divisão da área sob a curva em trapézios.



Fonte: Da autora.

Dessa forma, temos que

$$\begin{aligned} \int_a^b f(x)dx &= \int_{a=x_1}^{x_2} f(x)dx + \int_{x_2}^{x_3} f(x)dx + \dots + \int_{x_{n-1}}^{x_n=b} f(x)dx \\ &\simeq [f(x_1) + f(x_2)] \frac{h}{2} + [f(x_2) + f(x_3)] \frac{h}{2} + \dots + [f(x_{n-1}) + f(x_n)] \frac{h}{2} \\ &= \frac{h}{2} [f(x_1) + 2f(x_2) + \dots + 2f(x_{n-1}) + f(x_n)] \end{aligned}$$

é a regra dos trapézios repetida. Veja outros métodos de integração numérica em [16].

O algoritmo para este método é apresentado a seguir.

Algoritmo 3. *Integração por regra dos trapézios repetida.*

```
trap <- function(f, a, b, n) {
  h= (b-a)/n                #tamanho dos intervalos
  k = seq(a, b, h)          #vetor com pontos do intervalo
  r = f(a) + f(b); c = length(k) - 1
  for (i in 2:c) {
    r = r + 2*(f((a+(i-1)*h)))
  }
  r = r*(h/2); r }
```

Como exemplo, vamos considerar a integral definida

$$I = \int_0^5 e^x dx.$$

Utilizando uma calculadora científica, obtemos como resultado desta integral o valor 6.389056.

Para utilizar o Algoritmo 3, primeiramente definimos a função f a ser integrada $f < -\text{function}(s)\{\exp(s)\}$. Quanto maior o valor de n , menor o erro cometido. Por exemplo, com $n = 10$, obtemos o valor de $I = 6.410339$, dado pelo comando `trap(f, 0, 5, 10)`, cometendo um erro de aproximadamente 0.02128267. Já com $n = 100$, o valor obtido é $I = 6.389269$ e o erro é de 0.0002129671.

Os algoritmos de derivação e integração foram utilizados como parte de diversos outros, como por exemplo no algoritmo do produto interno de $W_2^1[0,1]$, que é dado a seguir.

Algoritmo 4. *Produto interno e norma do espaço $W_2^1[0,1]$.*

Primeiramente, é necessário definir as funções f e g e um valor para n . As funções f e g são usadas para definir vetores u e v , com valores $f(t_i)$ e $g(t_i)$, respectivamente, ordenados, onde $t = (t_1, \dots, t_n)$ são os pontos utilizados na integração pela regra dos trapézios.

```
prod = function(u,v){          # Produto interno
du = D%*%u; dv = D%*%v      # Derivada numerica
w = du*dv                    # Produto das derivadas
trap = 0; for (l in 1:n){    # Integracao por trapezios
trap = trap + w[l] + w[l+1] }; trap = trap/(2*n)
p = u[1]*v[1]+ trap; p}
norm <- function(v){        # Norma
p = sqrt(prod(v,v)); p}
```

Este algoritmo foi utilizado como parte de outros, por exemplo, no Algoritmo 8.

Decomposição QR

O Algoritmo 5, implementado para a decomposição QR de matrizes, é dado a seguir e está adaptado para o Exemplo 2.19.

Algoritmo 5. *Decomposição QR via Gram-Schmidt para matrizes.*

```
m=4; n=3; V = matrix(c(1,0,1,0,1,1,0,1,0,2,-1,3),m,n)
Q = matrix(0,m,n); R = matrix(0,n,n)
```

Neste caso, como trabalharemos com vetores numéricos, precisamos calcular a norma euclidiana.

```
norm2 = function(V,i){ p = 0;
for(j in 1:m){p = p + (V[j,i])^2}; p = sqrt(p); p}
```

Agora fazemos a fatoração QR.

```
for(i in 1:n){R[i,i] = norm2(V,i); Q[,i] = V[,i]/R[i,i]
for (j in (i+1):n){
R[i,j] = t(Q[,i])%*%V[,j]      # produto escalar
V[,j] = V[,j]-(R[i,j]*Q[,i]) }}
Q; R; t(Q)%*%Q                  # teste
A-Q%*%R                          # Erro
```

O algoritmo de decomposição QR para funções é dado a seguir e está adaptado para o Exemplo 2.20.

Algoritmo 6. *Decomposição QR via ortogonalização de Gram-Schmidt para funções.*

```
a=0; n = 20; nd = 20; x = seq(0,1,by=1/n); t = seq(0,1,by=1/nd)
n = length(x); m = length(t)
```

Da mesma forma que fizemos para o cálculo de produto interno de duas funções em $W_2^1[0,1]$, vamos utilizar as funções $\Psi_j(t) = K(x_j, t)$ para construir uma matriz fi , em que cada entrada ij é dada por $K(x[j], t[i])$, onde x e t são dois vetores auxiliares.

```
K = function(r,s){1 + min(r,s)} # N. reproducao
fi = matrix(0,m,n)              # Matriz de funcoes tabeladas
for (i in 1:m){for(j in 1:n){fi[i,j] = K([j],t[i]) }}
```

Em seguida, aplicamos o algoritmo de decomposição QR.

```
QRGram = function(fi){V = fi
Q = matrix(0,n,n); R = matrix(0,n,n)
for(i in 1:n){
R[i,i] = norm(V[,i])            # norma
Q[,i] = V[,i]/R[i,i]
for (j in (i+1):n){if (j != (n+1)){
R[i,j] = prod(Q[,i],V[,j])     # prod. interno
V[,j] = V[,j]-(R[i,j]*Q[,i]) }}}}
```

```

B = matrix(0,m,2*n)
B[,1:n] = Q
B[1:n,(n+1):(2*n)] = R; B}
P = QRGram(fi) # Chamando Q e R
Q = P[,1:n]
R = P[1:n,(n+1):(2*n)] #R e triangular superior
fi - Q%*%R; max(abs(fi - Q%*%R)) # Erro

```

Invertendo a matriz R , encontramos uma matriz B com os valores de β_{ij} , para $1 \leq i, j \leq n$.

```

B = pinv(R) # Encontrando os betas
for (i in 1:n){ for (j in 1:n){ if (i>j){B[i,j] = 0}}}

```

As quatro últimas linhas do código são úteis para evitar possíveis erros carregados pelo algoritmo. Desta forma, zeramos todas as entradas de B que ficam abaixo da diagonal principal, pois como sabemos que R é triangular superior, também B o será.

Observação 0.1. *Na implementação do Algoritmo 6, utilizamos o comando `pinv()` para calcular a inversa generalizada de matrizes. Este comando faz parte do pacote `pracma` instalado no software *R*.*

Algoritmos para Aproximação de Solução de Equações Funcionais

Nesta seção exibimos os algoritmos elaborados para a aproximação de solução de equações funcionais.

Primeiramente, apresentamos uma breve fundamentação teórica do método de Nystrom. Este método nos retorna a aproximação da solução aplicada a um certo número de pontos. Em seguida, temos o algoritmo do método de colocação.

Por fim, expomos os algoritmos implementados para o método de espaços de Hilbert de reprodução conforme as diferentes propostas estudadas.

Método de Nystrom

Considere a equação integral não linear

$$u(x) = f(x) + \int_a^b K(x, t, u(t))dt, \quad a \leq x, t \leq b. \quad (0.1)$$

Sejam $x_1 < x_2 < \dots < x_{n+1}$ pontos igualmente espaçados por uma distância $h = \frac{b-a}{n+1}$, com $a = x_1$ e $b = x_{n+1}$. O objetivo do método é aproximar a solução $u(x)$ da Equação (0.1) em cada um dos pontos $x_i = a + (i-1)h$, para $i = 1, \dots, n+1$.

Podemos reescrever a Equação (0.1) nos pontos dados como

$$u(x_i) = f(x_i) + \sum_{j=1}^n \int_{x_j}^{x_{j+1}} K(x_i, t, u(t))dt.$$

Utilizando a regra dos trapézios para integração numérica, temos

$$u(x_i) \simeq f(x_i) + \frac{h}{2} \left(K(x_i, x_1, u(x_1)) + 2 \sum_{j=2}^n K(x_i, x_j, u(x_j)) + K(x_i, x_{n+1}, u(x_{n+1})) \right).$$

Tomando $f(x_i) = f_i$ e $K(x_i, x_j, u(x_j)) = K_{i,j,u_j}$ temos que

$$u_i \simeq f_i + \frac{h}{2} \left(K_{i,1,u_1} + 2 \sum_{j=2}^n K_{i,j,u_j} + K_{i,n+1,u_{n+1}} \right), \quad (0.2)$$

o que produz um sistema não linear de equações algébricas da forma

$$u = Tu,$$

em que $u = [u_1 \dots u_{n+1}]^t$, então podemos utilizar o método de ponto fixo para tentar resolver o sistema não linear. Temos então o seguinte esquema de recorrência

$$u^{s+1} = Tu^s \iff u_i^{s+1} = f_i + \frac{h}{2} \left(K_{i,1,u_1^s} + 2 \sum_{j=2}^n K_{i,j,u_j^s} + K_{i,n+1,u_{n+1}^s} \right), \quad (0.3)$$

para u_0 dado e $s = 0, 1, 2, \dots$. Se aplicarmos o método de Simpson ao sistema obtemos outra

discretização

$$u_i \simeq f_i + \frac{h}{3} \left[K_{i,1,u_1} + 2 \sum_{j=1}^{\frac{n}{2}} (K_{i,2j-1,u_{2j-1}} + K_{i,2j,u_{2j}}) + K_{i,n+1,u_{n+1}} \right], \quad (0.4)$$

desta forma, temos outro sistema não linear de $n+1$ equações e $n+1$ incógnitas (u_1, \dots, u_{n+1}) , também escrito na forma

$$u = Tu, \quad i = 1, 2, \dots, n+1,$$

em que $u = [u_1 \dots u_{n+1}]^t$. Então utilizando o método de ponto fixo podemos tentar resolver o sistema. Temos assim o esquema de recorrência dado a seguir

$$u^{s+1} = Tu^s \quad (0.5)$$

que é equivalente à

$$u_i^{s+1} = f_i + \frac{h}{3} \left[K_{i,1,u_1^s} + 2 \sum_{j=1}^{\frac{n}{2}} (K_{i,2j-1,u_{2j-1}^s} + K_{i,2j,u_{2j}^s}) + K_{i,n+1,u_{n+1}^s} \right].$$

Nas Equações (0.3) e (0.5), s é o número de iterações e podemos tomar o valor inicial $u_i^0 = f(x_i)$, $i = 1, 2, \dots, n+1$. Por recorrência, então podemos calcular o vetor de soluções u para todos os pontos x_i , com $i = 1, 2, \dots, n+1$.

Segue o algoritmo do método de Nystrom, adaptado para a Equação (5.1).

Algoritmo 7. *Método de Nystrom.*

```
f<-function(s) {exp(s+1)}
n = 10; k = 50; x = seq(0,1,by=1/n)
g = 0*f(x) # T(g_n) = g_(n+1)
for ( l in 1:k) {for (i in 1:(n+1)) {trap = 0
for ( j in 1:n) {trap = trap+(exp(x[i]-2*x[j]))*(g[j])^3 +
exp(x[i]-2*x[j+1]))*(g[j+1])^3}
g[i] = f(x[i])-trap/(2*n) }} # g(x[i])
erro = g-exp(x); max(abs(erro))
plot(x,erro, xlab="x",ylab="erro")# grafico
```



```
plot(x, g); points(x, exp(x), col="red")
```

Método de Colocação

Segue o algoritmo implementado para o método de colocação, descrito na Seção 4.1, utilizado no Exemplo 5.1.

Algoritmo 8. *Método de colocação.*

Primeiramente construímos o vetor de pontos para os cálculos, fornecemos a quantidade de iterações e a aproximação inicial.

```
n = 20; m = 20 # m = iteracoes
x = seq(0, 1, by=1/n)
c = 0*x # aprox. inicial
```

Em seguida, fornecemos a função g , o núcleo de reprodução K e construímos a matriz com as funções $\Psi'_i s$.

```
g <- function(s){s^3 + s*(cos(1)-1)/6} # funcoes
K <- function(i, s){1 + min(x[i], s)}
fi = matrix(0, n+1, n+1) # matriz de valores fi
for(i in 1:(n+1)){for(j in 1:(n+1)){fi[i, j] = K(j, x[i])}}
```

Agora, utilizando o comando `pinv()` para calcular matriz inversa construímos o sistema e iteramos.

```
z = pinv(fi); G = 0*x
for(i in 1:(n+1)){G[i] = g(x[i])} # Vetor F
for(l in 1:m){ aux <- function(s){q = 0
for (i in 1:(n+1)){q = q + c[i]*K(i, s)}; q}
Nuc = function(i, s){x[i]*s^2*sin(aux(s))/2}; trap = 0*x
for (i in 1:(n+1)){
for ( j in 1:n){trap[i]=trap[i]+Nuc(i, x[j])+ Nuc(i, x[j+1])}
trap[i] = trap[i]/(2*n)}; t = 0*x
for(i in 1:(n+1)){t[i] = F[i] + trap[i]}
c = z%*%t} # tc = (fi)^(-1) * t
```

Por fim, construímos a solução, calculamos o erro e plotamos o gráfico.

```
sol = function(s) { q=0
for (i in 1:(n+1)) {q = q + c[i]*K(i,s)}; q}
xx = seq(0,1,by=1/n); Sol = 0*x
for (i in 1:(length(xx))) {Sol[i] = sol(xx[i])}
max(abs(Sol-x^3))          # Erro maximo
par(mfrow = c(1,2)); plot(xx,Sol,'l',xlab="x",ylab="f(x)")
points(x,x^3,col="red")
legend("topleft", legend=c("Sol. aproximada", "Sol. exata"),
lty=c(1,NA),col=c(1,2),lwd=1,bty="n",pch=c(NA,1))
plot(xx,(xx)^3 - Sol,'l',xlab="x",ylab="erro")
```

Método de Espaços de Hilbert de Reprodução para Equações Funcionais Lineares

Segue o algoritmo do método de espaços de Hilbert de reprodução utilizando decomposição QR via Gram-Schmidt modificado para equações integrais lineares. O algoritmo está adaptado para o Exemplo 5.6. Nesse caso, temos $f_n = \sum_{i=1}^n \langle f, \Psi_i \rangle \Psi_i$ e estimamos os coeficientes de Fourier $\langle f, \Psi_i \rangle$.

Algoritmo 9. *Método de espaços de Hilbert de reprodução aplicável a equações integrais lineares, utilizando decomposição QR via Gram-Schmidt para encontrar uma base ortonormal.*

Definimos primeiramente o número de pontos n para o problema e nd para os cálculos de integrais.

```
a = 0; b = 1; n = 20; nd = 50
x = seq(a,b,by=1/n); t = seq(a,b,by=1/nd)
h = 1/nd; n = length(x); m = length(t)
```

Definimos também as funções referentes ao problema, como o núcleo de reprodução K , o núcleo da integral Nuc , g e o operador linear L .

```
K <- function(s,u) {1+min(s,u)}; Nuc = function(s,u) {s*u}
g <- function(s) {exp(s)+s}; L <- function(g,s) {
w = function(r) {Nuc(s,r)*(g(r))}
```

```
ps=0; for (l in 1:nd){ ps = ps+w(t[l])+w(t[l+1]) }
ps = ps/(2*nd)g(s); q}
```

Agora, definimos as funções $L^*K_{x_i}$, que chamamos de f_i , lembrando que $L^*K_{x_i}(y) = \overline{LK_y(x_i)}$, construímos a matriz A do problema, encontramos os $\beta'_i s$ por meio da decomposição QR, conforme a Observação 4.7, construímos as funções $\Psi_i(x) = \sum_{j=1}^i \beta_{ji} L^*K_{x_j}(x)$, que chamamos de F_i e calculamos os coeficientes de Fourier aproximados (q_i).

```
fi <- function(i,s){g <- function(u){K(s,u)}; L(g,x[i])}
A = matrix(0,m,n)
for (i in 1:m){for (j in 1:n){A[i,j] = fi(j,t[i])}}
B = QRGram(A) # Teste
Q = B[,1:n]; R = B[1:n,(n+1):(2*n)] # R e triangular superior
B = pinv(R)
for (i in 1:n){for (j in 1:n){ if (i>j){B[i,j]=0}} }
Fi <- function(i,s){ # funcoes ortogonais em W12
p = 0; for (j in 1:i){p = p+B[j,i]*fi(j,s)}; p}
fex <- function(s){exp(s)} # Teste do processo
fe = g(t); qi = 0*x
for (i in 1:n){for (j in 1:i){qi[i] = qi[i]+B[j,i]*g(x[j])}}
```

Por fim, calculamos a solução aproximada g_a , conforme a Equação (4.24), o erro e plotamos os gráficos.

```
ga <- function(s){ # aproximacao de fex
p=0; for ( j in 1:n ){
p = p+qi[j]*Fi(j,s)}; p}
max(abs(nSf-fex(t))) # Erro maximo
Sf<-function(s){ga(s)}
nSf=0*t; for ( i in 1:length(t)){nSf[i]=Sf(t[i])}
curve(fex,0,1); points(t,nSf,col="red")
```

Segue agora o algoritmo implementado para o método de espaços de Hilbert de reprodução para equações lineares baseado no Corolário 4.22. Este está adaptado para o Exemplo 5.8.

Algoritmo 10. *Método de espaços de Hilbert de reprodução para equações lineares.*

Primeiramente, definimos os dados e as funções referentes ao problema. Neste caso, definimos diretamente $L^*K_{x_i}(y)$ como sendo o operador L aplicado à função núcleo K_y no ponto x_i e chamamos de ψ a função $L^*K_{x_i}$. Além disso, chamamos de $L\psi$ a função $L(L^*R_{x_i})$.

```
x = seq(0,1,by=1/10); n= length(x); h=10^(-5)
k=20; ts = seq(0,1,by=1/k)
K <- function(s,t){1+min(s,t)} ; Nuc <- function(s,t){s*t}
LK <- function(s,t){                                     # L(K_s) (t)
w <- function(r){Nuc(t,r)*( K(s,r) )}; ps=0
for (m in 1:k){ps=ps+w(ts[m])+w(ts[m+1])}
q =K(s,t) + ps/(2*k); q}
psi <- function(s,i){
q = LK(x[i],s); q}
Lpsi <- function(t,i){                                   # L aplicado em psi
ww <- function(r){ Nuc(t,r)*(psi(r,i))}; ps=0
for (m in 1:k){ps=ps+ww(ts[m])+ww(ts[m])}
q = psi(t,i) + ps/(2*k); q}
```

Agora, calculamos a matriz Ψ , chamando-a de Ψ . Além disso, calculamos o vetor G (G) e calculamos as matrizes inversas necessárias para a resolução do sistema.

```
Psi = matrix(0,n,n)                                     # Matriz psi_ij
for (j in 1:n) { for (i in 1:n) {
uv <- function(s){(Lpsi(s+h, j) - Lpsi(s-h, j))*(psi(s+h, i)
- psi(s-h, i))*10^10/4}; ps=0                          # Dericacao numerica
for (m in 1:k){ps=ps+uv(ts[m])+uv(ts[m])}
Psi[i,j] = ps/(2*k) + Lpsi(0,j)*psi(0,i) }}
G = 0*x; for (i in 1:n) {
gpsi = function(s){( exp(s)+1 )*(psi(s+h,i)- psi(s-h,i))*h/2}
ps=0; for (m in 1:k){ps= ps + gpsi(ts[m])+ gpsi(ts[m])}
G[i] = ps/(2*k) + psi(0,i)}; z = pinv(Psi); U=z%*%U
```

Por fim, calculamos a solução aproximada, o erro e plotamos os gráficos.

```

Sol<-function(s){ p=0; for (i in 1:n){p=p+U[i]*psi_i(s,i)}; p}
xx = seq(0,1,by=1/50); m=length(xx); sol=0*xx
for (j in 1:m){sol[j]=Sol(xx[j])}; Exact<-function(s){exp(s)}
curve(Exact,0,1,xlab="x",ylab="f(x)"); points(xx,sol,col="red")
legend("topleft", legend=c("Sol. exata", "Sol. aproximada"),
lty=c(1,NA), col=c(1,2), lwd=1, bty="n", pch=c(NA,1))
plot(xx,sol-Exact(xx),'l',xlab="x",ylab="Erro")

```

Apresentamos agora o algoritmo baseado no Corolário 4.24. A ideia aqui é a mesma do Algoritmo 10, temos que encontrar os valores $\alpha'_i s$ resolvendo o sistema $\Psi\alpha = \mathbf{G}$. Todavia, as entradas de Ψ e de \mathbf{G} são calculadas sem utilizar o produto interno do espaço. Utilizamos apenas o núcleo de reprodução e não precisamos conhecer o produto interno nem o espaço que está sendo tratado. Isto torna o algoritmo bem mais elegante. Este algoritmo está adaptado para o Exemplo 5.7.

Algoritmo 11. *Método de espaços de Hilbert de reprodução para equações lineares sem utilizar produto interno.*

```

x = seq(0,1,by=1/10); n= length(x); nd=10; t = seq(0,1,by=1/nd)
K = function(s,t){exp(-4*(s-t)^2)} # N. de reproducao
gg = function(s){3*exp(s)+exp(sin(s)-s)}
L = function(s,g) { # equacao linear L(f)=gg
w = function(u){2*exp(s-u)*g(u)}
ps=0 # regra do trapezio
for (m in 1:nd){ ps=ps+w(t[m])+w(t[m+1]) }
p=ps/(2*nd)+g(s)+exp(-s)*g(sin(s)); p}
psi = function(s,i) { # para calcular L(K^x_i)(s)
ga=function(t){K(x[i],t)}; q = L(s,ga); q}
Psi = matrix(0,n,n) # Matriz Psi
for (j in 1:n) {
g1=function(s){psi(s,j)}
g2=function(s){L(s,g1)}
for (i in 1:n) {
Psi[i,j]=L(x[i],g2)} # L^2(L^*(K^{x_j})(x_i))
G = 0*x

```

```

for (i in 1:n) {G[i]=L(x[i],gg)}      # para calcular o vetor G
require(pracma); z = pinv(Psi)      # inversa general. de Psi
alpha=z%*%G
Sol<-function(s) { p=0
for (i in 1:n){ p=p+alpha[i]*psi(s,i)}; p}

```

O cálculo do erro e o plot dos gráficos são feitos de maneira análoga ao Algoritmo 10.

Método de Espaços de Hilbert de Reprodução para Equações Funcionais Não Lineares

Segue o algoritmo implementado para o método de espaços de Hilbert de reprodução para equações não lineares, utilizado no Exemplo 5.3.

Algoritmo 12. *Algoritmo para equações não lineares usando ortogonalização.*

Para este algoritmo, primeiramente são necessários algoritmos básicos de derivada numérica, produto interno e decomposição QR, semelhantes aos Algoritmos 1, 4 e 6, respectivamente. Definimos então os dados do problema e funções como núcleo de reprodução (K), operador L (L) e $L^*R_{x_i}$ (fi).

```

a=0; b=1; n=10; x=seq(a,b,by=1/n); n=length(x)
nd=20; t=seq(a,b,by=1/nd); h=1/nd; m=length(t)
nuc <- function(s,u) {u*((s-1)*u^2+s*(2-3*s+s^2+6))/6}
K <- function(s,u) {                                     # N. de reproducao
if (s>=u) {p= nuc(s,u) }
else {p= nuc(u,s)}; p}
f <- function(g,s) {-g(s)^2+3*s^2-2*s}
L <- function(g,s) {
if ( s== 0) {p=(-3*g(0)+4*g(h)-g(2*h))/(2*h) }
if ( s==1) {p=(g(1-2*h)-4*g(1-h)+3*g(1))/(2*h) }
else{p=(g(s+h)-g(s-h))/(2*h) }
p=p -(s^3-s^2)*g(s); p}
fi <- function(i,u) { gg <- function(s) {K(u,s)}; p=L(gg,x[i]); p}

```

Agora, construímos a matriz com as funções $L^*R_{x_i}$ e por meio da decomposição QR calculamos as funções ortonormais Ψ_i (Fi).

```

A=matrix(0,m,n) # matriz do problema
for (i in 1:m){for (j in 1:n){A[i,j]=fi(j,t[i])}}
B=QRGram(A)
Q=B[,1:n]; R=B[1:n,(n+1):(2*n)]
B=pinv(R)
for (i in 1:n){for (j in 1:n){ if (i>j){ B[i,j]=0}}}
max(abs(A%*%pinv(R)-Q)) # Teste de eficiencia QR
Fi <- function(i,s){p=0 # funcoes ortonormais
for (j in 1:i){p=p+B[j,i]*fi(j,s)}; p}

```

Podemos utilizar a solução exata para calcular os coeficientes de Fourier ($fou[i]$). Então, calculamos a aproximação da solução e o erro. Os gráficos são feitos como no Algoritmo 10.

```

fex <- function(s){ s^3-s^2 } # Solucao exata
fe=fex(t); fou=0*x
for (i in 1:n){ fou[i]=prod(fe,Q[,i])}
fou # Coeficientes de Fourier
gfou=function(u){q=0 # aproximacao da solucao
for ( j in 1:n ){ q=q+fou[j]*Fi(j,u)}; q}
tg=seq(0,1,by=10^(-2))
Sfou=0*tg # p/ calculo erro e grafico
for ( i in 1:length(tg) ){Sfou[i]=gfou(tg[i])}
max(abs((Sfou-fex(tg))))

```

Podemos ainda estimar os coeficientes de Fourier ($fQ[i]$), dados pela Equação (4.15), e então calcular a solução aproximada.

```

gQ=0*x; fQ=gQ # Coef. Fourier aprox.
g <- function(u){ 0 }
for (i in 1:n){
gQ[i]= f(g,x[i])
for (j in 1:i){
fQ[i]=fQ[i]+B[j,i]*gQ[j]}
g <- function(u){q=0 # aproximacao
for ( j in 1:i ){q=q+fQ[j]*Fi(j,u)}; q} }

```

```

g <- function(u){ q=0
for ( j in 1:n ){q=q+fQ[j]*Fi(j,u)}; q}
tg=seq(0,1,by=10^(-2)) # grafico
Sf=0*tg
for ( i in 1:length(tg)){ Sf[i]=g(tg[i])}
max(abs((Sf-fex(tg))))

```

Podemos ainda, avaliar a precisão das estimativas dos coeficientes de Fourier.

```

plot(1:n,fou); points(1:n,fQ,col="red")# Comp. coef. Fourier
plot(1:n,fou-fQ); max(abs(fou-fQ))

```

Apresentamos agora o algoritmo baseado no Teorema 4.21, adaptado para o Exemplo 5.5.

Algoritmo 13. *Algoritmo para equações não lineares.*

```

x = seq(0,1,by=1/10); n= length(x) # pontos x_i
nd=10; t = seq(0,1,by=1/nd) # pontos regra trapezio
K <- function(s,t){1+min(s,t)} # nucleo de reproducao
g<-function(s){0} # aprox. inicial
# pra L(f)=G(.,f)
L <- function(s,g){g(s)} # operador linear L(g)
NL <- function(s,g){ # parte nao linear
NL(.,g)=G(.,g)
}
w <- function(u){-exp(s-2*u)*(g(u)^3)}
ps=0
for (m in 1:nd){ ps=ps+w(t[m])+w(t[m+1]) }
p=ps/(2*nd)+exp(s+1); p}
psi <- function(s,i){ # calcular L(R_x_i)(s)
ga<-function(t){K(x[i],t)}; q = L(s,ga); q}
Psi = matrix(0,n,n) # matriz Psi
for (j in 1:n) { g1<-function(s){psi(s,j)}
g2<- function(s){L(s,g1)}
for (i in 1:n) {Psi[i,j]=L(x[i],g2)}# L^2(L^*(K^{x_j})(x_i))

```



```

gls<- function(a){p=0*x           # calcular o vetor G
for ( i in 1:n){p[i]=NL(x[i],g)}; p}
fls<-function(a){p=Psi%*%a-gls(a)} # funcao para minimizar
# |Psi*a-G|^2=fls(a)^2
gradls<-function(a){ p=0*Psi      # funcao auxiliar
for ( i in 1:n){ for (j in 1:n){
w  <- function(u){-3*exp(x[i]-2*u)*(g(u)^2)*psi(u,j)}
ps=0; for (m in 1:nd){ ps=ps+w(t[m])+w(t[m+1]) }
ps=ps/(2*nd); p[i,j]=ps }}; p}
gradmin<-function(a){             # gradiente para
                                   |Psi*a-G|^2
p=2*(Psi-gradls(a))%*%fls(a); p}
a=0*x; pass=1                     # met. parecido gradiente
while(max(abs(fl s(a)))>10^(-5)){  # eqnt max|fls(a)|>10^(-3)
b=a;a=a-pass*gradmin(a)
if (sum(fl s(a)*fl s(a))>sum(fl s(b)*fl s(b))){a=b; pass=pass/10}
else{ pass=pass*10}              # Condicao de Armijo
g<-function(s){ p=0              # solucao aprox. para
                                   L(f)=G(.,f)
for ( i in 1:n){p=p+a[i]*psi(s,i)}; p}}

```