



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS FÍSICAS E MATEMÁTICAS
CURSO DE GRADUAÇÃO EM MATEMÁTICA

Beatriz Carolina Souza Rodrigues

Aproximação de funções e o fenômeno de Runge

Florianópolis
2022

Beatriz Carolina Souza Rodrigues

Aproximação de funções e o fenômeno de Runge

Trabalho de Conclusão de Curso do Curso de Graduação em Matemática do Centro de Ciências Físicas e Matemáticas da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Matemática e Computação Científica.
Orientador: Prof. Leonardo Koller Sacht, Dr.

Florianópolis
2022

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Rodrigues, Beatriz Carolina Souza
Aproximação de funções e o fenômeno de Runge / Beatriz
Carolina Souza Rodrigues ; orientador, Leonardo Koller
Sacht, 2022.
53 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro de Ciências
Físicas e Matemáticas, Graduação em Matemática e Computação
Científica, Florianópolis, 2022.

Inclui referências.

1. Matemática e Computação Científica. 2. Interpolação
Polinomial. 3. Fenômeno de Runge. 4. Splines. I. Sacht,
Leonardo Koller. II. Universidade Federal de Santa
Catarina. Graduação em Matemática e Computação Científica.
III. Título.

Beatriz Carolina Souza Rodrigues

Aproximação de funções e o fenômeno de Runge

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de “Bacharel em Matemática e Computação Científica” e aprovado em sua forma final pelo Curso de Graduação em Matemática.

Florianópolis, 12 de dezembro de 2022.

Silvia Martini de Holanda, Dra.
Coordenadora do Curso

Banca Examinadora:

Prof. Leonardo Koller Sacht, Dr.
Orientador

Profa. Luciane Ines Asmann Schuh, Dra.
Avaliadora
Universidade Federal de Santa Catarina

Prof. Wagner Barbosa Muniz, Dr.
Avaliador
Universidade Federal de Santa Catarina

À minha avó, Maria.

AGRADECIMENTOS

Aqui marco o fim de uma jornada e o início de novos caminhos a serem trilhados e sonhados. Me mudar para tão longe de casa com tão pouca idade definitivamente não foi uma tarefa fácil, deixar um lar e um mundo conhecido para ir atrás dos meus sonhos foi um dos maiores desafios vividos até aqui. Logo, se os últimos anos foram possíveis é graças a todos aqueles que se fizeram presentes, cada um a sua forma, sendo rede de apoio e me trazendo até este momento.

Um agradecimento especial à minha avó, que mesmo carinhosamente me chamando de louca, é a pessoa que mais torce e vibra a cada conquista minha. É professora da vida e minha maior referência. Aos meus pais e irmã por acreditarem em mim e nos meus sonhos, embarcando nessa loucura junto comigo. Por todos os conselhos, palavras ditas e não ditas, e principalmente pelo apoio quando acreditei que não ia conseguir. Foi com vocês que aprendi o verdadeiro significado da resiliência, e que nada é tão difícil que não possamos dar um jeitinho. À toda minha família pela compreensão por todas as ausências seja pela distância ou pelos estudos, e na maioria das vezes por ambos. Por todo acolhimento e felicidade quando eu voltava mesmo que fosse para uma visita rápida, todos os almoços feitos com muito amor, todas as comemorações feitas por videochamada... Vocês são incríveis.

À minha segunda família, aquela que eu pude escolher e cultivar. Acho incrível como algumas relações se tornam tão sólidas e preciosas ao longo da vida. Aos meus amigos de São Paulo, eu não me recordo de uma vida antes vocês, e é um imenso prazer dividir a minha vida com todos. Obrigada por todos os nossos encontros anuais que já duram mais de uma década, por todo acolhimento nos dias difíceis, e as alegrias compartilhadas. Agradeço também às amizades cultivadas nessa ilha, alguns conhecidos em festas que chegaram, se aconchegaram e decidiram ficar, colegas de curso e companheiros de PET. Os dias se tornaram muito mais leves, e se existe uma certeza em tudo isso é que só foi possível sobreviver tão longe de casa, pois encontrei uma segunda casa em vocês.

Aos professores que me trouxeram aqui, por dedicarem a vida à uma tarefa tão nobre que é ensinar. Pelos corredores do CFM não aprendi apenas matemática, mas muitas lições de vida e profissionalismo também. Vocês tem minha eterna admiração. Em especial, meu orientador, por todas as trocas nesses anos de pesquisa, pelos conselhos que foram muito além da academia, apoio e paciência. Foi um prazer compartilhar esse tempo com você, não poderia ter feito outra escolha que não essa.

*“Nunca será um verdadeiro matemático
aquele que não for um pouco de poeta.”
(Karl Weierstrass)*

RESUMO

A interpolação polinomial é um caso particular para o problema geral de interpolação, que tem por objetivo aproximar funções através de um conjunto finito de pontos dados. O polinômio interpolador de menor grau é único e pode ser obtido através de alguns métodos, cada qual com as suas particularidades. Como qualquer aproximação está sujeita a erros, existem casos em que os métodos não funcionam como o esperado, ocasionando, por exemplo, o fenômeno de Runge, um problema que envolve oscilações próximas às bordas do intervalo da função que buscamos aproximar. Sendo assim, é necessário utilizar outros métodos que o evitam, como a interpolação por splines ou por partes, que oferecem um maior controle das oscilações, gerando uma aproximação mais suave e, por vezes, mais precisa.

Palavras-chave: Interpolação Polinomial. Fenômeno de Runge. Splines.

ABSTRACT

Polynomial interpolation is a particular case for the general interpolation problem, which aims at approximating functions through a finite set of given points. The polynomial interpolator of smallest degree is unique and can be obtained using a few methods, each one with their particularities. Since any approximation is subject to errors, there are cases in which the methods do not work as expected, causing the Runge phenomenon, a problem of oscillation near the interval's edges of the function that we seek to approximate. Therefore, it is necessary to use other methods that avoid it, such as interpolation by splines or by parts that offer greater control over oscillations, generating a smoother and sometimes more accurate approximation.

Keywords: Polynomial Interpolation. Runge Phenomenon. Splines.

LISTA DE FIGURAS

Figura 1 – Gráfico: $p(x) = \frac{5}{6}x^2 + \frac{1}{6}x + 1$	26
Figura 2 – Gráfico: $p(x) = 1 + 6x - x^3$	26
Figura 3 – Gráfico polinômio interpolador.	27
Figura 4 – Gráfico: $f(x) = e^{-x^2}$ em $[-4, 4]$	28
Figura 5 – Gráfico do polinômio interpolador de grau 6.	29
Figura 6 – Gráfico do polinômio interpolador de grau 10.	29
Figura 7 – Gráfico de interpolação da função $g(x) = \frac{1}{1+25x^2}$, no intervalo $[-1, 1]$, por polinômios de grau 8 e 16 em pontos igualmente espaçados. . .	30
Figura 8 – Gráfico de interpolação da função $g(x) = \frac{1}{1+25x^2}$ por pontos de Chebyshev.	31
Figura 9 – Comparativo dos gráficos dos polinômios de aproximação.	32
Figura 10 – Funções da base de splines lineares.	36
Figura 11 – Funções da base de splines cúbicos.	37
Figura 12 – Interpolação via polinômio de quarto grau e via splines cúbicos. . .	41
Figura 13 – Função de Bessel interpolada via polinômios e via splines cúbicos. .	42

LISTA DE TABELAS

Tabela 1 – Diferenças Divididas.	20
Tabela 2 – Diferenças Finitas.	21
Tabela 3 – Valores de $B_i(x)$ e derivadas nos pontos da partição.	37
Tabela 4 – Função de Bessel avaliada em alguns pontos dados.	42

LISTINGS

B.1	Código gerador da Figura 1.	47
B.2	Código gerador da Figura 2.	47
B.3	Código gerador da Figura 3.	48
B.4	Código gerador para fenômeno de Runge na interpolação da função e^{-x^2}	49
B.5	Código gerador para fenômeno de Runge na função $\frac{1}{1+25x^2}$	50
B.6	Código gerador da interpolação por pontos de Chebyshev para função de Runge.	50
B.7	Código gerador do comparativo de interpolação por pontos de Chebyshev e pontos igualmente espaçados para a função e^{-x^2}	51
B.8	Função geradora da base de splines lineares.	52
B.9	Função geradora da base de splines cúbicos.	53
B.10	Código gerador do comparativo de interpolação via polinômio de quarto grau e via splines cúbicos para função de Runge.	53
B.11	Código gerador do comparativo de interpolação via polinômio de quarto grau e via splines cúbicos para função de Bessel.	54

SUMÁRIO

1	INTRODUÇÃO	13
2	FUNDAMENTOS DE INTERPOLAÇÃO POLINOMIAL	14
2.1	TEORIA DA INTERPOLAÇÃO POLINOMIAL	14
2.2	MÉTODOS PARA INTERPOLAÇÃO	17
2.2.1	Método de Lagrange	17
2.2.2	Método de Diferenças Divididas de Newton	19
2.2.3	Método de Gregory-Newton	21
2.3	ERRO DE TRUNCAMENTO	22
2.4	VANTAGENS E DESVANTAGENS	25
2.5	RESULTADOS	25
3	FENÔMENO DE RUNGE	28
3.1	PONTOS DE CHEBYSHEV	31
4	INTERPOLAÇÃO POR SPLINES	34
4.1	BASES	35
4.1.1	Bases para splines lineares	35
4.1.2	Bases para splines cúbicos	36
4.1.3	Bases genéricas	38
4.2	INTERPOLAÇÃO POR PARTES	38
4.3	EXEMPLOS	41
5	CONCLUSÃO	43
	REFERÊNCIAS	44
	APÊNDICE A – MATRIZ DE VANDERMONDE	45
	APÊNDICE B – CÓDIGOS	47
B.1	CÓDIGOS REFERENTES AO CAPÍTULO 2	47
B.2	CÓDIGOS REFERENTES AO CAPÍTULO 3	49
B.3	CÓDIGOS REFERENTES AO CAPÍTULO 4	52

1 INTRODUÇÃO

A interpolação é uma ferramenta útil para determinar o valor de uma função em um ponto não dado, fornece meios para trabalhar com funções definidas de forma tabular ou dadas por equações. Para o caso particular da interpolação polinomial as aproximações são feitas através de polinômios de grau até n . Esse caso específico é frequentemente usado devido a algumas propriedades da família dos polinômios que os tornam mais razoáveis de se trabalhar.

Dentre suas aplicações destacam-se a estimativa de transferência de calor; encontrar derivadas a partir de dados experimentais; tornar funções complicadas em mais simples, melhorando a eficiência de uma função que é usada muitas vezes; fornecer ferramentas matemáticas usadas no desenvolvimento de métodos para as áreas de teoria da aproximação, integração numérica, solução numérica de equações diferenciais e processamento de imagens.

Existem diversos métodos para realizar uma interpolação, cada qual com as suas particularidades que, a depender do caso, podem tornar a aproximação simples e precisa. Como toda aproximação, essa também está sujeita a erros. No entanto, há uma taxa de tolerância para eles. No processo de interpolação podemos nos deparar com o fenômeno de Runge, um fenômeno oscilatório capaz de aumentar significativamente o erro de interpolação. Sendo assim, é necessário adotar novas táticas a fim de evitá-lo e obter uma aproximação mais precisa.

A fim de realizar uma revisão teórica dos métodos de interpolação e implementá-los, no Capítulo 2 apresentaremos um estudo sobre os fundamentos da interpolação polinomial, alguns métodos existentes para realizar uma interpolação e seus erros de aproximação, além da aplicação destes para alguns exemplos fazendo uso de ferramentas computacionais. Esses métodos podem não funcionar da maneira esperada em alguns casos, surgindo o fenômeno de Runge, considerado um problema na aproximação apresentado no Capítulo 3. Dessa forma, o Capítulo 4 contempla novos métodos que possibilitam evitar o fenômeno de Runge, seja por uma escolha específica de pontos para interpolar ou separando o problema inicial de interpolação em problemas menores.

Para uma melhor compreensão do trabalho é recomendado que o leitor tenha um bom embasamento teórico sobre cálculo, álgebra linear e métodos numéricos. Sendo sugerido a leitura dos livros (STEWART, 2008), (HOFFMAN, 1971) e (FAIRES; BURDEN, 2004).

2 FUNDAMENTOS DE INTERPOLAÇÃO POLINOMIAL

O conceito de interpolação é a seleção de uma função $p(x)$ de uma classe de funções de tal forma que o gráfico de $y = p(x)$ passe por um conjunto finito de pontos dados (ATKINSON, 2008, p. 131).

Entre seus usos, destacam-se o desenvolvimento de meios para trabalhar com funções definidas de forma tabular e fornecer ferramentas matemáticas usadas no desenvolvimento de métodos para as áreas de teoria da aproximação, integração numérica, solução numérica de equações diferenciais e processamento de imagens.

A interpolação polinomial é um caso particular do problema geral de interpolação, no qual a família de funções é constituída por polinômios. Um fator importante é que a derivada e a integral indefinida de um polinômio são fáceis de determinar e também são polinômios. Por essas razões, polinômios são frequentemente usados para aproximar funções contínuas (FAIRES; BURDEN, 2004, p.106). Além disso, dada uma função arbitrária, definida e contínua em um intervalo fechado e limitado, pelo Teorema da Aproximação de Weierstrass (ATKINSON, 2008, Teorema 4.1), sabemos que existe um polinômio que se aproxima uniformemente desta função.

2.1 TEORIA DA INTERPOLAÇÃO POLINOMIAL

Sejam x_0, x_1, \dots, x_n números distintos, e sejam y_0, y_1, \dots, y_n valores associados da função. Para encontrar o polinômio $p(x)$ que interpola os dados fornecidos considere

$$p(x) = a_0 + a_1x + \dots + a_mx^m.$$

Para um polinômio geral de grau m existem $m + 1$ parâmetros independentes a_0, a_1, \dots, a_m .

Considerando o caso $n = m$, queremos encontrar a_0, a_1, \dots, a_n tal que

$$a_0 + a_1x_0 + \dots + a_nx_0^n = y_0$$

$$\vdots$$

$$a_0 + a_1x_n + \dots + a_nx_n^n = y_n$$

Assim, existe um sistema linear com $n+1$ equações e $n+1$ incógnitas e resolvê-lo é equivalente a resolver o problema de interpolação polinomial.

Em notação vetorial e matricial, o sistema é

$$Xa = y$$

com

$$X = [x_i^j],$$

onde $i, j = 0, 1, \dots, n$, X é uma matriz de Vandermonde, e

$$a = [a_0, a_1, \dots, a_n]^T,$$

$$y = [y_0, \dots, y_n]^T.$$

A abordagem através da matriz de Vandermonde é ineficiente quando temos um grande número de pontos para interpolar. Essa matriz é mal condicionada, e ao aumentar a sua dimensão nos deparamos com erros de arredondamento na solução do sistema linear que fornece o polinômio interpolador.

Teorema 2.1.1 *Dados $n+1$ pontos distintos x_0, \dots, x_n e $n+1$ ordenadas y_0, \dots, y_n , existe um único polinômio $p(x)$ de grau menor ou igual a n que interpola y_i em x_i , com $i = 0, 1, \dots, n$.*

Demonstração:

- i. De acordo com a demonstração do Teorema A.0.1, podemos mostrar que para a matriz X

$$\det(X) = \prod_{0 \leq j < i \leq n} (x_i - x_j).$$

Assim, temos que $\det(X) \neq 0$, uma vez que os pontos x_i são distintos. Portanto, como garante o Teorema A.0.1, X é não singular e o sistema $Xa = y$ possui uma única solução a . Isso prova a existência e unicidade de um polinômio de interpolação de grau menor ou igual a n .

- ii. Por um teorema padrão da álgebra linear (ATKINSON, 2008, Teorema 7.2), o sistema $Xa = b$ tem única solução se, e somente se, o sistema homogêneo $Xb = 0$ tem apenas a solução trivial $b = 0$. Portanto, assumamos $Xb = 0$ para algum b . Usando b , defina

$$p(x) = b_0 + b_1x + \dots + b_nx^n$$

De $Xb = 0$, tem-se

$$p(x_i) = 0, \text{ para } i = 0, 1, \dots, n$$

O polinômio $p(x)$ possui $n+1$ zeros e grau $p(x) \leq n$. Mas o único polinômio que pode ser escrito como $p(x) = a_0 + a_1x + \dots + a_nx^n$ com $n+1$ raízes é o polinômio identicamente nulo, pois todo polinômio de grau n com coeficientes complexos tem no máximo n raízes complexas (HOFFMAN, 1971, p. 129). Logo $p(x) \equiv 0$. Mas então todos os coeficientes $b_i = 0$, com $i = 0, 1, \dots, n$ completando a prova.

- iii. Considere o problema de interpolação no qual

$$y_i = 1 \text{ e } y_j = 0, \text{ para } i \neq j$$

para algum i , $0 \leq i \leq n$. Queremos um polinômio de grau $\leq n$ com os n zeros x_j , $j \neq i$. Então

$$p(x) = c(x - x_0) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n)$$

para alguma constante c . A condição $p(x_i) = 1$ implica

$$c = \frac{1}{[(x_i - x_0) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)]}$$

Esse polinômio é escrito na forma

$$L_i(x) = \prod_{j \neq i} \frac{(x - x_j)}{(x_i - x_j)}$$

com $i = 0, 1, \dots, n$.

Para resolver o problema geral de interpolação, escreva

$$p(x) = y_0 L_0(x) + y_1 L_1(x) + \dots + y_n L_n(x)$$

Para cada $i = 0, 1, \dots, n$, queremos que $p(x_i) = y_i$, assim temos

$$p(x_i) = y_0 L_0(x_i) + y_1 L_1(x_i) + \dots + y_n L_n(x_i) = y_i$$

Defina,

$$L_k(x) = \frac{(x - x_0)(x - x_1) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_n)}{(x_k - x_0)(x_k - x_1) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_n)}$$

Observe que

$$L_k(x_i) = \begin{cases} 0 & \text{se } k \neq i \\ 1 & \text{se } k = i \end{cases}$$

Assim, temos que o grau de L_k é n uma vez que é o produto de n fatores $(x - x_i)$, então $p(x)$ possui grau $\leq n$.

Para provar a unicidade, suponha $q(x)$ é outro polinômio de grau $\leq n$ que satisfaz o problema geral de interpolação. Defina

$$r(x) = p(x) - q(x)$$

Então o grau de $r(x) \leq n$, e

$$r(x_i) = p(x_i) - q(x_i) = y_i - y_i = 0$$

com $i = 0, 1, \dots, n$.

Como $r(x)$ tem $n + 1$ zeros, segue que $r(x) \equiv 0$. Isso prova que $p(x) \equiv q(x)$, completando a prova. ■

2.2 MÉTODOS PARA INTERPOLAÇÃO

2.2.1 Método de Lagrange

Pelo Teorema 2.1.1 sabemos que existe um único polinômio interpolador que passa por um conjunto de pontos dados. Independente do método adotado para encontrá-lo, esse polinômio deve ser o mesmo. A partir de agora veremos alguns métodos de interpolação polinomial. Pelo item *iii* da demonstração desse teorema é possível construir um polinômio de grau $\leq n$ que interpola y_i em x_i , com $i = 0, 1, \dots, n$. Isso será feito através dos polinômios $L_i(x)$ da demonstração que definem o que chamaremos de polinômio interpolador de Lagrange.

Considere a construção do polinômio que passa pelos pontos arbitrários (x_0, y_0) e (x_1, y_1) . Para determinar esse polinômio podemos obter a função f de tal forma que $f(x_0) = y_0$ e $f(x_1) = y_1$ através de um polinômio do primeiro grau interpolando os valores de f nos pontos dados.

Sendo assim, definimos as funções

$$L_0(x) = \frac{x - x_1}{x_0 - x_1} \text{ e } L_1(x) = \frac{x - x_0}{x_1 - x_0}.$$

O polinômio de interpolação linear de Lagrange que passa por (x_0, y_0) e (x_1, y_1) será dado por

$$P(x) = L_0(x)f(x_0) + L_1(x)f(x_1) = \frac{x - x_1}{x_0 - x_1}f(x_0) + \frac{x - x_0}{x_1 - x_0}f(x_1)$$

Então, considere

$$L_0(x_0) = 1, L_0(x_1) = 0, L_1(x_0) = 0, \text{ e } L_1(x_1) = 1,$$

que implica em

$$P(x_0) = 1 \cdot f(x_0) + 0 \cdot f(x_1) = f(x_0) = y_0$$

e

$$P(x_1) = 0 \cdot f(x_0) + 1 \cdot f(x_1) = f(x_1) = y_1.$$

Dessa forma, P é o único polinômio de grau um que passa por (x_0, y_0) e (x_1, y_1) .

Este resultado para dois pontos pode ser generalizado, resultando no método de Lagrange.

Teorema 2.2.1 *Se x_0, x_1, \dots, x_n são $n + 1$ números distintos e f é uma função cujos valores são dados nesses números, então existe um único polinômio $P(x)$ de grau no máximo n existe com*

$$f(x_k) = P(x_k)$$

para cada $k = 0, 1, \dots, n$.

Este polinômio é dado por

$$P(x) = f(x_0)L_{n,0}(x) + \dots + f(x_n)L_{n,n}(x) = \sum_{k=0}^n f(x_k)L_{n,k}(x)$$

onde, para cada $k = 0, 1, \dots, n$

$$L_{n,k}(x) = \prod_{i=0, i \neq k}^n \frac{(x - x_i)}{(x_k - x_i)}.$$

Demonstração: Veja o item *iii* da demonstração do Teorema 2.1.1. ■

A partir de agora passaremos a substituir a notação $L_{n,k}(x)$ por $L_k(x)$.

Exemplo 2.2.1 Determine, via método de Lagrange, o polinômio interpolador que passa pelos pontos $(-1, 4)$, $(0, 1)$, $(1, -4)$. Depois calcule $P_2(0, 5)$.

Considere os pontos dados, note que o polinômio interpolador será de grau dois e, portanto,

$$P_2(x) = y_0L_0(x) + y_1L_1(x) + y_2L_2(x) = 4L_0(x) + 1L_1(x) - 4L_2(x).$$

Além disso, segue que

$$\begin{aligned} \bullet L_0(x) &= \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} = \frac{(x - 0)(x - 1)}{(-1 - 0)(-1 - 1)} = \frac{x^2 - x}{2} \\ \bullet L_1(x) &= \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} = \frac{(x + 1)(x - 1)}{(0 + 1)(0 - 1)} = \frac{x^2 - 1}{-1} \\ \bullet L_2(x) &= \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} = \frac{(x + 1)(x - 0)}{(1 + 1)(1 - 0)} = \frac{x^2 + x}{2} \end{aligned}$$

$$\text{Logo, } P_2(x) = 4 \left[\frac{x^2 - x}{2} \right] + \left[\frac{x^2 - 1}{-1} \right] - 4 \left[\frac{x^2 + x}{2} \right] = -x^2 - 4x + 1.$$

Finalmente, calculando $P_2(0, 5)$:

$$P_2(0, 5) = -x^2 - 4x + 1 = -0,5^2 - 4 \cdot 0,5 + 1 = -1,25.$$

Verificando se o polinômio realmente passa pelos pontos dados, temos:

$$P_2(-1) = -(-1)^2 - 4 \cdot (-1) + 1 = 4,$$

$$P_2(0) = -0^2 - 4 \cdot 0 + 1 = 1,$$

$$P_2(1) = -1^2 - 4 \cdot 1 + 1 = -4.$$

2.2.2 Método de Diferenças Divididas de Newton

A forma de Lagrange é normalmente usada para gerar sucessivamente aproximações polinomiais de grau mais alto em um ponto específico, sendo inconveniente passar de um polinômio de interpolação para outro de grau maior. Mas existem outras formas que são muito mais convenientes, por exemplo, o método de diferenças divididas de Newton, usado para gerar sucessivamente os próprios polinômios a partir de pontos não uniformemente espaçados.

Suponha que $P_n(x)$ seja o n -ésimo polinômio de Lagrange que coincide com uma função f nos pontos x_0, x_1, \dots, x_n . Embora este polinômio seja único, existem formas diferentes de representá-lo algebricamente. As diferenças divididas de f em relação a x_0, x_1, \dots, x_n são usados para expressar $P_n(x)$ na forma

$$P_n(x) = d_0 + d_1(x - x_0) + d_2(x - x_0)(x - x_1) + \dots + d_n(x - x_0) \cdots (x - x_{n-1})$$

para constantes apropriadas d_0, d_1, \dots, d_n .

Para determinar o valor de d_0 , calculamos $P_n(x)$ em x_0 , isto é,

$$d_0 = P_n(x_0) = f(x_0)$$

Analogamente, calculando $P_n(x_1)$, temos

$$f(x_0) + d_1(x_1 - x_0) = P_n(x_1) = f(x_1)$$

Assim, para calcular o valor de d_1 , segue que

$$d_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

Considere então a construção do operador de diferenças divididas.

A diferença dividida de ordem zero da função f em relação à x_i , denotada por $f[x_i]$ é definida pelo valor de f em x_i , logo

$$f[x_i] = f(x_i)$$

A primeira diferença dividida da função f em relação à x_i e x_{i+1} , denotada por $f[x_i, x_{i+1}]$, é definida por

$$f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}$$

A segunda diferença dividida da função f em relação à x_i, x_{i+1} e x_{i+2} denotada por $f[x_i, x_{i+1}, x_{i+2}]$, é definida por

$$f[x_i, x_{i+1}, x_{i+2}] = \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i}$$

Analogamente, depois das $(k-1)$ -ésimas diferenças divididas $f[x_i, x_{i+1}, \dots, x_{i+k-1}]$ e $f[x_{i+1}, x_{i+2}, \dots, x_{i+k}]$ serem calculadas, a k -ésima diferença dividida da função f em relação à $x_i, x_{i+1}, x_{i+2}, \dots, x_{i+k}$ é definida por

$$f[x_i, x_{i+1}, \dots, x_{i+k-1}, x_{i+k}] = \frac{f[x_{i+1}, x_{i+2}, \dots, x_{i+k}] - f[x_i, x_{i+1}, \dots, x_{i+k-1}]}{x_{i+k} - x_i}.$$

Assim, de modo geral a n -ésima diferença dividida será definida por

$$f[x_1, x_2, \dots, x_n] = \frac{f[x_1, \dots, x_n] - f[x_1, x_2, \dots, x_{n-1}]}{x_n - x_0}.$$

Por fim, o polinômio interpolador $P_n(x)$ é dado por

$$P_n(x) = f[x_0] + \sum_{k=1}^n f[x_0, x_1, \dots, x_k](x - x_0) \dots (x - x_{k-1}),$$

com o valor de $f[x_0, x_1, \dots, x_k]$ independente da ordem dos números x_0, x_1, \dots, x_k .

Exemplo 2.2.2 *Determine, via método de diferenças divididas de Newton, o polinômio interpolador que passa pelos pontos $(-1, 4)$, $(0, 1)$, $(1, -4)$. Depois calcule $P_2(0, 5)$.*

Pelo método de diferenças divididas de Newton, temos que

$$P_2(x) = f(x_0) + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2].$$

Calculando as diferenças divididas, obtemos:

Tabela 1 – Diferenças Divididas.

x	Ordem 0	Ordem 1	Ordem 2
-1	$f[x_0] = 4$	$f[x_0, x_1] = -3$	$f[x_0, x_1, x_2] = -1$
0	$f[x_1] = 1$	$f[x_2, x_1] = -5$	
-1	$f[x_2] = -4$		

Fonte: **Autora**.

Escrevendo o polinômio interpolador

$$P_2(x) = 4 + (-3)(x + 1) + (-1)(x + 1)(x - 0) = -x^2 - 4x + 1.$$

Calculando $P_2(0, 5)$:

$$P_2(0, 5) = -x^2 - 4x + 1 = -0,5^2 - 4 \cdot 0,5 + 1 = -1,25.$$

E verificando se o polinômio passa pelos pontos dados:

$$P_2(-1) = -(-1)^2 - 4 \cdot (-1) + 1 = 4,$$

$$P_2(0) = -0^2 - 4 \cdot 0 + 1 = 1,$$

$$P_2(1) = -1^2 - 4 \cdot 1 + 1 = -4.$$

2.2.3 Método de Gregory-Newton

O método de Newton possui um caso particular para pontos igualmente espaçados. Enquanto no método anterior é utilizado o operador de diferenças divididas, neste novo método teremos o operador de diferenças finitas, muito similar ao anterior.

As diferenças finitas podem ser usadas para determinar o grau máximo do polinômio de interpolação que pode ser usado com segurança, com base na precisão das entradas da tabela, a imprecisão nos dados afeta diretamente o resultado final. Destaca-se também no uso para detectar ruído nos dados, como situações em que o ruído é grande em relação aos erros de arredondamento.

A fórmula da diferença finita é construída fazendo uso do operador Δ da diferença dividida de Newton. De forma geral, os polinômios de Gregory-Newton são dados por

$$P_n(x) = f(x_0) + \sum_{i=1}^n \left(\frac{\Delta^i f(x_0)}{i! h^i} \right) (x - x_0)(x - x_1) \cdots (x - x_{i-1})$$

onde $n, i = 1, 2, 3, \dots$ e $h = x_{i+1} - x_i$ uma constante, ou seja, o espaçamento entre os pontos dados.

Exemplo 2.2.3 Determine, via polinômios de Gregory-Newton, o valor em $x = 0,5$ do polinômio interpolador que passa pelos pontos $(-1, 4)$, $(0, 1)$, $(1, -4)$.

Considere os pontos dados e note que

$$h = 0 - (-1) = 1.$$

Então, temos

$$P_2(x) = f(x_0) + \frac{\Delta^1 f(x_0)}{1! 1^1} \cdot (x + 1) + \frac{\Delta^2 f(x_0)}{2! 1^2} \cdot (x + 1)(x - 0)$$

Agora, considere o tabelamento

Tabela 2 – Diferenças Finitas.

x	Ordem 0	Ordem 1	Ordem 2
-1	4	$\Delta^1 f(x_0) = -3$	$\Delta^2 f(x_0) = -2$
0	1	-5	
1	-4		

Fonte: **Autora**.

Calculando o polinômio interpolador:

$$P_2(x) = 4 + \frac{(-3)}{1! 1^1} (x + 1) + \frac{(-2)}{2! 1^2} (x + 1)(x - 0) = -x^2 - 4x + 1.$$

Assim, $P_2(0,5)$ é dado por

$$P_2(0,5) = -0,5^2 - 4 \cdot 0,5 + 1 = -1,25.$$

Por último, verificando se o polinômio passa pelos pontos dados:

$$P_2(-1) = -(-1)^2 - 4 \cdot (-1) + 1 = 4,$$

$$P_2(0) = -0^2 - 4 \cdot 0 + 1 = 1,$$

$$P_2(1) = -1^2 - 4 \cdot 1 + 1 = -4.$$

2.3 ERRO DE TRUNCAMENTO

Os métodos de interpolação polinomial são utilizados para resolução de diversos problemas, mas tratam-se de aproximações passíveis de erros. Sendo assim, com a fórmula do erro é possível obter a diferença aproximada para analisar se a aproximação é válida.

A fórmula do erro é um resultado teórico importante, não é uma fórmula única, portanto pode ser apresentada em muitas variações. A seguir, apresentamos uma delas.

Teorema 2.3.1 *Suponha x_0, x_1, \dots, x_n números distintos no intervalo $[a, b]$ e $f \in C^{n+1}[a, b]$. Então, para cada x em $[a, b]$, um $\xi(x)$ (geralmente desconhecido) entre x_0, x_1, \dots, x_n , e portanto em (a, b) , existe com*

$$f(x) = P(x) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!} (x-x_0)(x-x_1) \cdots (x-x_n). \quad (1)$$

Demonstração: Considere x_0, x_1, \dots, x_n números distintos pertencentes ao intervalo $[a, b]$ e $f \in C^{n+1}[a, b]$.

CASO I: Se $x = x_k$, para algum $k = 0, 1, \dots, n$ então $f(x_k) = P(x_k)$, para qualquer $\xi(x_k)$ arbitrário em (a, b) .

CASO II: Se $x \neq x_k$, para todo $k = 0, 1, \dots, n$ defina a função g para $t \in [a, b]$ como

$$g(t) = f(t) - P(t) - [f(x) - P(x)] \frac{(t-x_0)(t-x_1) \cdots (t-x_n)}{(x-x_0)(x-x_1) \cdots (x-x_n)} = f(t) - P(t) - [f(x) - P(x)] \prod_{i=0}^n \frac{(t-x_i)}{(x-x_i)}.$$

Como $f \in C^{n+1}[a, b]$ e $P \in C^\infty[a, b]$, segue que $g \in C^{n+1}[a, b]$.

Para $t = x_k$, temos

$$g(x_k) = f(x_k) - P(x_k) - [f(x) - P(x)] \prod_{i=0}^n \frac{(x_k - x_i)}{(x - x_i)} = 0 - [f(x) - P(x)] \cdot 0 = 0.$$

Além disso,

$$g(x) = f(x) - P(x) - [f(x) - P(x)] \prod_{i=0}^n \frac{(x - x_i)}{(x - x_i)} = f(x) - P(x) - [f(x) - P(x)] = 0.$$

Assim, $g \in C^{n+1}[a, b]$ e g é zero nos $n + 2$ números distintos x, x_0, x_1, \dots, x_n .

Pelo Teorema Generalizado de Rolle, enunciado abaixo como Teorema 2.3.3, existe um número ξ em (a, b) tal que $g^{(n+1)}(\xi) = 0$. Sendo assim,

$$0 = g^{(n+1)}(\xi) = f^{(n+1)}(\xi) - P^{(n+1)}(\xi) - [f(x) - P(x)] \frac{d^{n+1}}{dt^{n+1}} \left[\prod_{i=0}^n \frac{(t - x_i)}{x - x_i} \right]_{t=\xi}. \quad (2)$$

Mas lembre que $P(x)$ é um polinômio de grau no máximo n , então a $(n+1)$ -ésima derivada, $P^{(n+1)}(x)$ é identicamente zero.

Além disso, $\prod_{i=0}^n \frac{(t - x_i)}{(x - x_i)}$ é um polinômio de grau $n + 1$, então

$$\prod_{i=0}^n \frac{(t - x_i)}{(x - x_i)} = \left[\frac{1}{\prod_{i=0}^n (x - x_i)} \right] t^{(n+1)} + (\text{termo de menor grau em } t)$$

e

$$\frac{d^{n+1}}{dt^{n+1}} \prod_{i=0}^n \frac{(t - x_i)}{(x - x_i)} = \frac{(n+1)!}{\prod_{i=0}^n (x - x_i)}.$$

Dessa forma, a equação 2 torna-se

$$0 = f^{(n+1)}(\xi) - 0 - [f(x) - P(x)] \frac{(n+1)!}{\prod_{i=0}^n (x - x_i)}.$$

Resolvendo para $f(x)$, obtemos

$$f(x) = P(x) + \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^n (x - x_i).$$

■

Note que a fórmula do erro apresentada pelo Teorema 2.3.1 é difícil de aplicar, pois não temos a expressão geral da função f , nem das suas derivadas. O que esperamos é que $f^{(n+1)}$ seja limitada.

Por fim, considere o Teorema Generalizado de Rolle, citado na demonstração anterior.

Teorema 2.3.2 (Teorema de Rolle) *Seja f uma função que satisfaça as seguintes hipóteses:*

1. f é contínua no intervalo fechado $[a, b]$.
2. f é derivável no intervalo aberto (a, b) .

$$3. f(a) = f(b).$$

Então existe um número c em (a, b) tal que $f'(c) = 0$.

Demonstração: Existem três casos:

CASO I: $f(x) = k$, k uma constante.

Então $f'(x) = 0$, assim, o número pode ser tomado como *qualquer* número em (a, b) .

CASO II: $f(x) > f(a)$ para algum x em (a, b) .

Pelo Teorema dos Valores Extremos (que pode ser aplicado pela hipótese 1), f tem um valor máximo em um algum lugar de $[a, b]$. Como $f(a) = f(b)$, ele deverá ter esse valor máximo em um número c num intervalo aberto (a, b) . Então f tem um máximo *local* em c e, pela hipótese 2, f é derivável em c . Portanto $f'(c) = 0$ pelo Teorema de Fermat.

CASO III: $f(x) < f(a)$ para algum x em (a, b) .

Pelo Teorema dos Valores Extremos, f tem um valor mínimo em $[a, b]$ e, uma vez que $f(a) = f(b)$, ela assume esse valor mínimo em um número c em (a, b) . Novamente $f'(c) = 0$ pelo Teorema de Fermat (STEWART, 2008, p. 257).

■

Teorema 2.3.3 (Teorema Generalizado de Rolle) *Suponha que $f \in C[a, b]$ seja n vezes diferenciável em (a, b) . Se $f(x) = 0$ nos $n + 1$ números distintos $a \leq x_0 < x_1 < \dots < x_n \leq b$, então um número c em (x_0, x_n) e, portanto, em (a, b) , existe com $f^{(n)}(c) = 0$.*

Demonstração: Seja $f \in C[a, b]$, n vezes diferenciável em (a, b) .

Suponha que $f(x) = 0$ nos $n + 1$ pontos distintos $x_0 < x_1 < \dots < x_n$ em $[a, b]$.

Queremos provar que existe um número c em (a, b) tal que $f^{(n)}(c) = 0$. Faremos isso por indução.

CASO BASE: $n=1$ Segue diretamente do Teorema 2.3.2.

HIPÓTESE DE INDUÇÃO: Suponha que o resultado vale para $k = n$ com $k \in \mathbb{N}$ e $c \in (a, b)$.

INDUÇÃO: Agora, note que este resultado também vale para $n = k + 1$.

Por hipótese, $f(x) = 0$ em $k + 1$ pontos distintos $x_0 < x_1 < \dots < x_k < x_{k+1}$.

Aplicando o Teorema 2.3.2 nos intervalos $[x_0, x_1], \dots, [x_{k-1}, x_k], [x_k, x_{k+1}]$ existem $c_1, \dots, c_k, c_{k+1} \in (x_0, x_1), \dots, (x_{k-1}, x_k), (x_k, x_{k+1})$ respectivamente tal que $f'(c_1) = 0, \dots, f'(c_k) = 0, f'(c_{k+1}) = 0$.

Novamente pelo Teorema 2.3.2 $f'(c_1) = \dots = f'(c_k) = f'(c_{k+1}) = 0$, pela hipótese de indução $f^{(k)}(c) = 0$, então $f^{(k+1)}(c) = 0$.

■

2.4 VANTAGENS E DESVANTAGENS

No método de Lagrange não é preciso que os dados sejam colocados em ordem crescente ou decrescente, esse método é tão eficiente quanto o método de Newton quando feita somente uma interpolação, sendo mais prático visto que não é necessário armazenar os dados em um tabelamento de diferenças divididas. Em contrapartida, quando for necessário realizar várias interpolações teremos uma quantidade excessiva de cálculos, e ao adicionarmos um novo termo todos os valores de $L(x)$ precisam ser recalculados. É um método instável quando a interpolação é de ordem alta e apresenta um alto custo computacional.

Já no método de diferenças divididas de Newton ao aumentar os nós de interpolação é preciso apenas adicionar um termo no polinômio obtido anteriormente, tendo um custo computacional consideravelmente menor que o método de Lagrange. Entretanto, nesta forma também só é possível deduzir o erro de truncamento se a função interpolada for conhecida analiticamente.

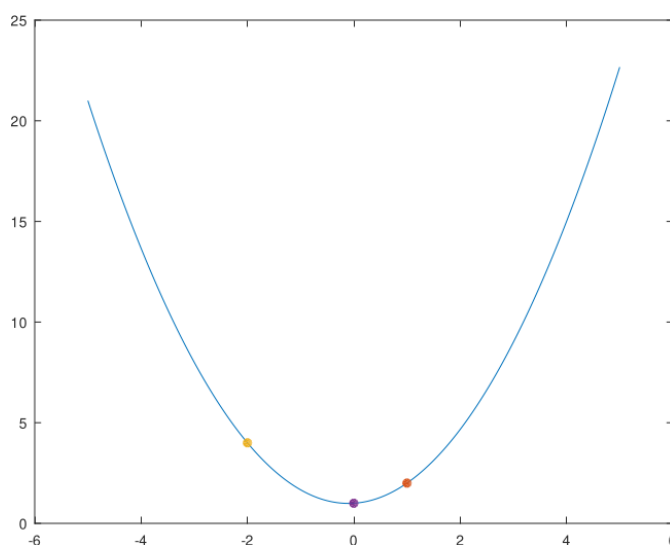
O método de Gregory-Newton é o que apresenta menor custo computacional, porém conta com a grande desvantagem que os pontos no eixo das abscissas, utilizados para encontrar o polinômio interpolador, precisam estar obrigatoriamente equidistantes.

2.5 RESULTADOS

Os exemplos a seguir foram realizados através do software Matlab, aplicando todo o conteúdo visto anteriormente. Os códigos utilizados para solução desses podem ser encontrados no Apêndice B.

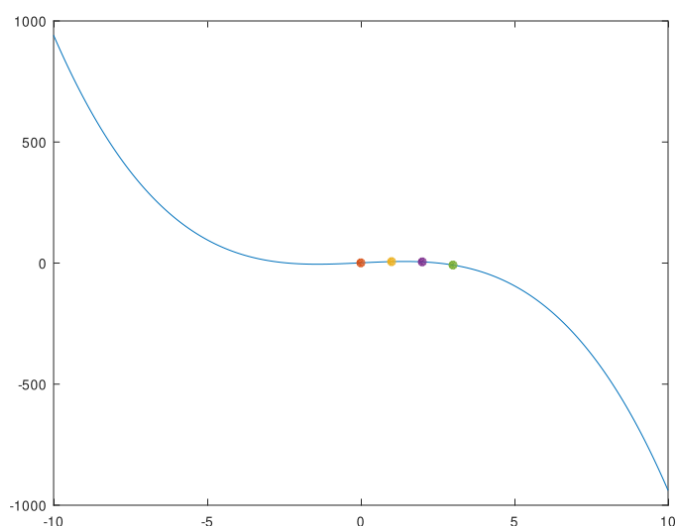
Exemplo 2.5.1 *Calcule o polinômio que interpola os pontos $(1,2)$, $(-2,4)$ e $(0,1)$. Plote os pontos e o polinômio.*

Através do Matlab encontramos o polinômio interpolador $p(x) = \frac{5}{6}x^2 + \frac{1}{6}x + 1$. Seu gráfico é apresentado na Figura 1.

Figura 1 – Gráfico: $p(x) = \frac{5}{6}x^2 + \frac{1}{6}x + 1$.Fonte: **Autora**.

Exemplo 2.5.2 Encontre o polinômio interpolador dos pontos $(0,1)$, $(1,6)$, $(2,5)$ e $(3,-8)$. Plote os pontos e o polinômio.

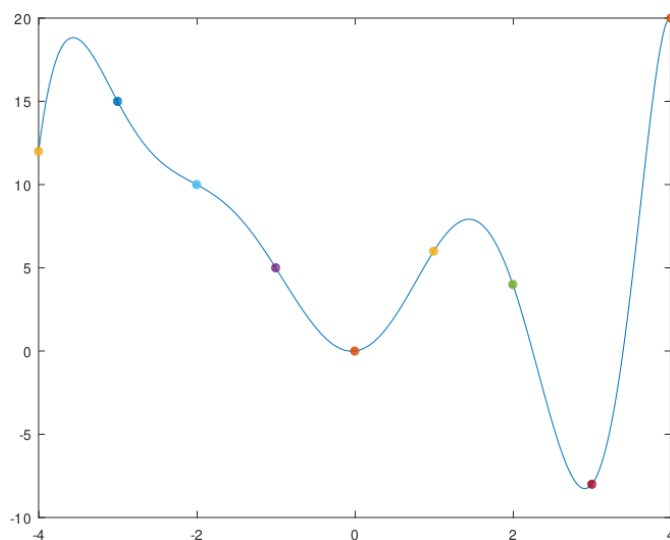
O código em Matlab é similar ao exemplo anterior. Encontramos então o polinômio interpolador dado por $p(x) = 1 + 6x - x^3$, seu gráfico é mostrado na Figura 2.

Figura 2 – Gráfico: $p(x) = 1 + 6x - x^3$.Fonte: **Autora**.

Exemplo 2.5.3 Encontre o polinômio interpolador dos pontos $(0,0)$, $(1,6)$, $(-1,5)$, $(2,4)$, $(-2,10)$, $(3,-8)$, $(-3,15)$, $(4,20)$ e $(-4,12)$. Plote os pontos e o polinômio.

Assim como nos exemplos anteriores o polinômio interpolador foi calculado através Matlab, polinômio esse que possui como gráfico a Figura 3.

Figura 3 – Gráfico polinômio interpolador.



Fonte: **Autora.**

Este último exemplo trata-se de um polinômio de oitavo grau, assim ferramentas computacionais, como o Matlab tornam-se extremamente necessárias para realizar a interpolação com precisão. Embora os métodos apresentem algebrismos simples, o alto grau os tornam difíceis para serem realizados mecanicamente.

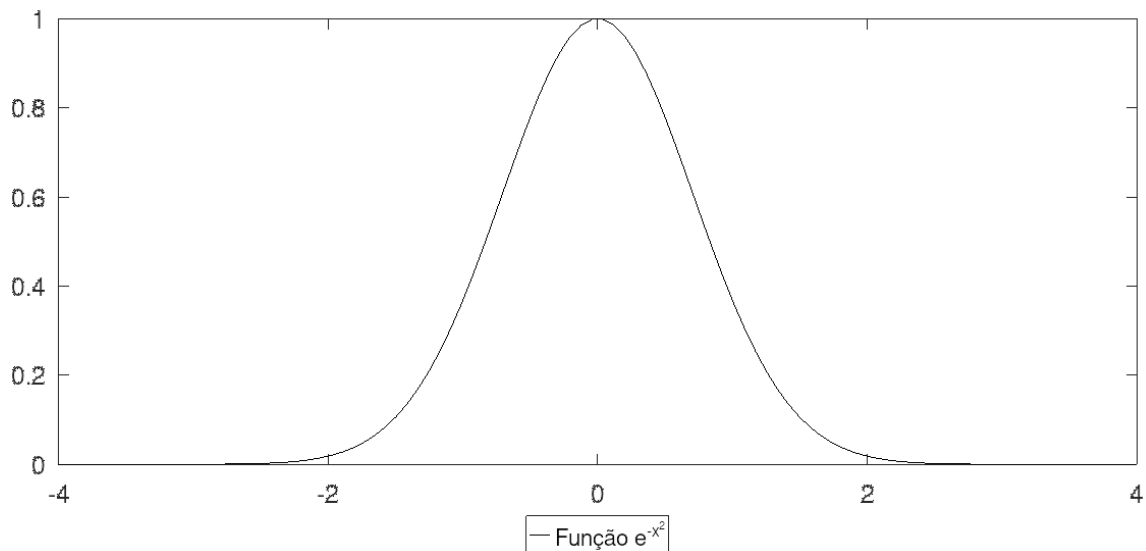
3 FENÔMENO DE RUNGE

Vimos que existem algumas maneiras diferentes de aproximar uma função através de pontos dados, porém podem existir alguns casos específicos onde esses métodos podem não funcionar da maneira esperada, e então se torna necessário uma nova abordagem para o problema.

Considerando o caso de pontos igualmente espaçados, intuitivamente, temos que quanto mais pontos dados para realizar a interpolação, mais próximo da função será o polinômio interpolador, porém este fato não procede.

Considere o gráfico da função $f(x) = e^{-x^2}$, no intervalo $[-4, 4]$, mostrado na Figura 4.

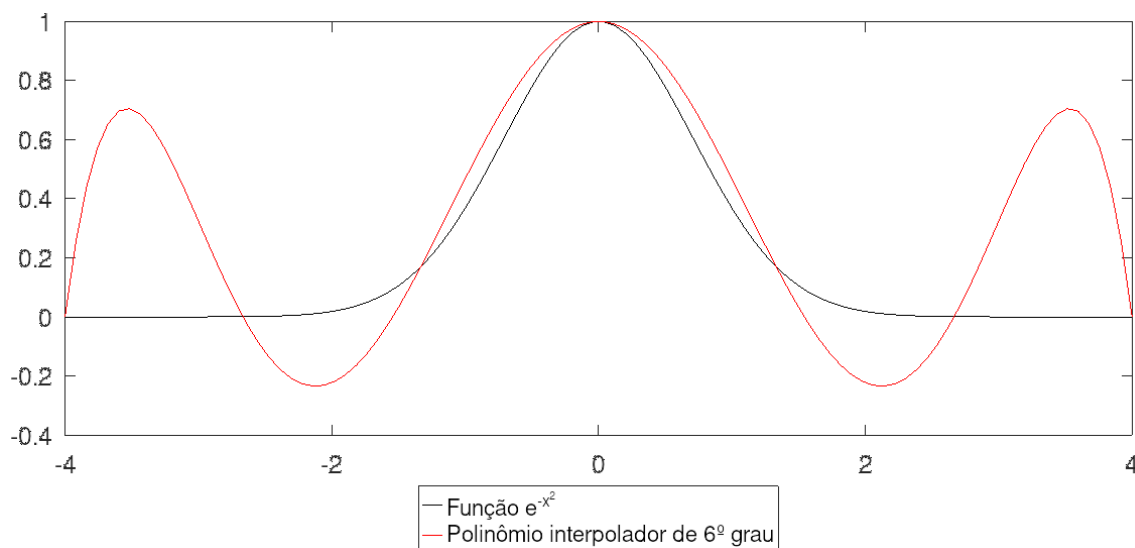
Figura 4 – Gráfico: $f(x) = e^{-x^2}$ em $[-4, 4]$.



Fonte: **Autora.**

Fazendo o gráfico do polinômio interpolador de grau 6, que interpola a função f em 7 pontos igualmente espaçados, obtemos a Figura 5.

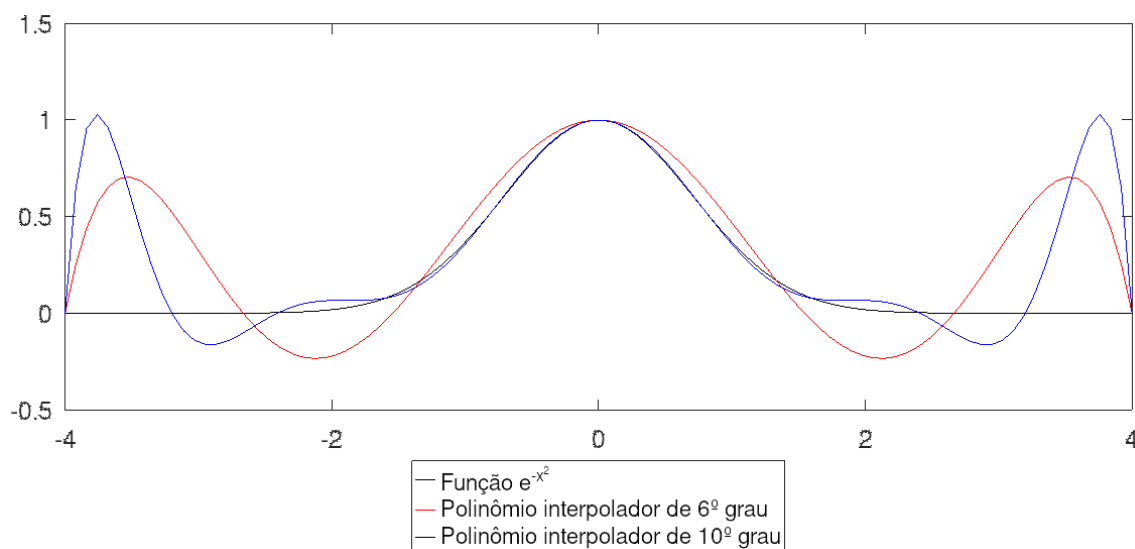
Figura 5 – Gráfico do polinômio interpolador de grau 6.



Fonte: **Autora.**

Note que essa aproximação apresenta oscilações significativas nas bordas do intervalo, podemos tentar encontrar uma aproximação melhor, intuitivamente quanto mais pontos melhor será a interpolação. Considere o novo polinômio interpolador de grau 10, que interpola f em 11 pontos igualmente espaçados dado pela Figura 6.

Figura 6 – Gráfico do polinômio interpolador de grau 10.



Fonte: **Autora.**

É possível notar que o erro máximo de interpolação aumentou, contrariando a hipótese que tínhamos inicialmente. Este é um fenômeno conhecido e leva o nome de Fenômeno de Runge.

Descoberto por Carl Runge, esse fenômeno é um problema de oscilação nas bordas de um intervalo, onde o erro de interpolação aumenta quando o grau do polinômio interpolador cresce. É amplamente conhecido através da função $f(x) = \frac{1}{1 + 25x^2}$ definida no intervalo $[-1, 1]$.

Teorema 3.0.1 (Fenômeno de Runge) *Se x_k são escolhidos para serem pontos da forma*

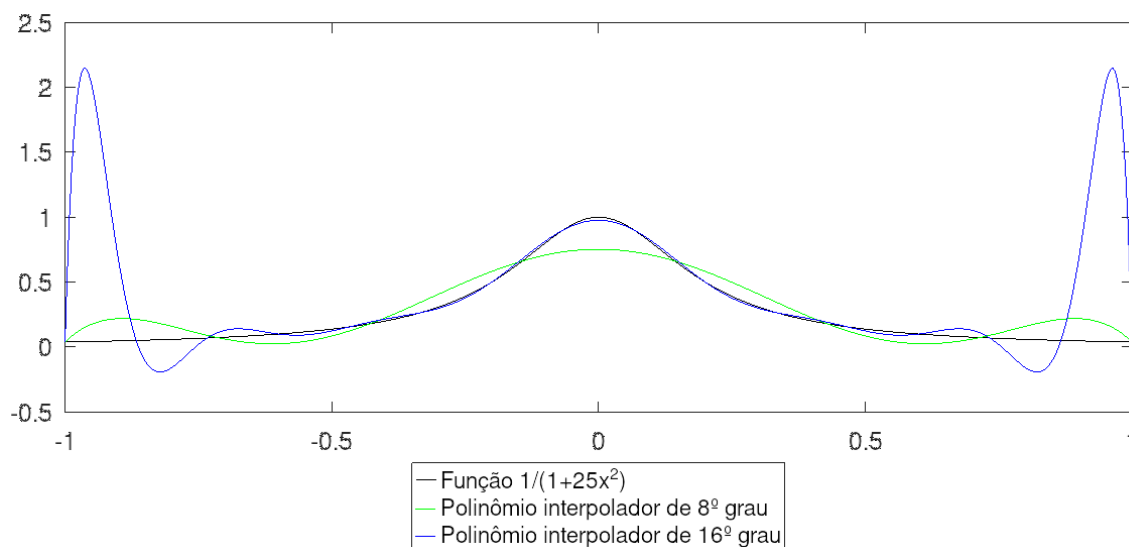
$$x_k = -1 + \frac{2k + 1}{n + 1},$$

com $k = 0, \dots, n$, para $n \geq 0$, então o polinômio de interpolação $p_n(x)$ não converge uniformemente em $[-1, 1]$ quando $n \rightarrow \infty$ para a função $g(x) = \frac{1}{1 + 25x^2}$.

A demonstração desse teorema pode ser encontrada na íntegra em (HANDSCOMB, 2014, p. 21-26). Para um melhor entendimento é necessário algum domínio sobre variáveis complexas, como funções analíticas, resíduos e polos, tais informações podem ser encontradas em (BROWN; CHURCHILL, 2015, capítulos 1-6). Pode ser feita começando pela fórmula do erro abordada na Seção 2.3, que de fato converge quando $n \rightarrow \infty$. Porém, para o caso $g(x) = \frac{1}{1 + 25x^2}$, $p(x)$ não converge uniformemente em $[-1, 1]$, isso acontece de forma análoga para a função $f(x) = \frac{1}{1 + x^2}$ no intervalo $(-5, 5)$.

Observe na Figura 7 a seguir o comportamento do fenômeno de Runge, para a função $g(x) = \frac{1}{1 + 25x^2}$ dada no Teorema 3.0.1.

Figura 7 – Gráfico de interpolação da função $g(x) = \frac{1}{1 + 25x^2}$, no intervalo $[-1, 1]$, por polinômios de grau 8 e 16 em pontos igualmente espaçados.

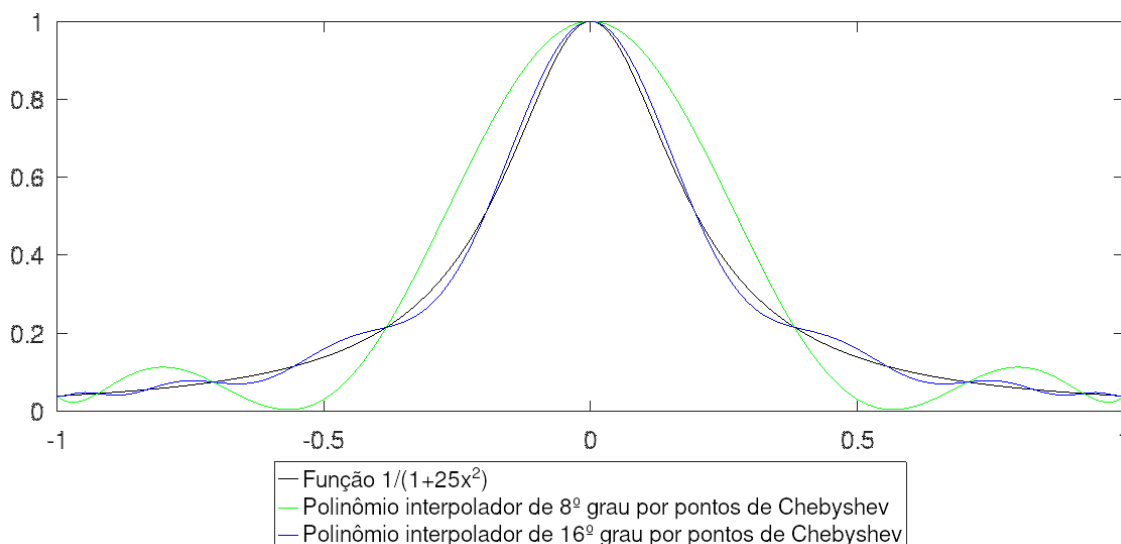


Fonte: **Autora.**

E se fosse possível que ao escolhermos os pontos, ao invés de utilizá-los igualmente espaçados, adotássemos um critério que evitasse o fenômeno de Runge? A

Figura 8 mostra como é possível fazer isso diminuindo significativamente o erro de interpolação. Esses pontos são conhecidos como pontos de Chebyshev.

Figura 8 – Gráfico de interpolação da função $g(x) = \frac{1}{1+25x^2}$ por pontos de Chebyshev.



Fonte: **Autora.**

3.1 PONTOS DE CHEBYSHEV

No capítulo anterior definimos, na equação (1), o erro de aproximação como

$$f(x) = P(x) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!} (x-x_0)(x-x_1) \cdots (x-x_n),$$

que é equivalente à

$$f(x) - P(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!} (x-x_0)(x-x_1) \cdots (x-x_n).$$

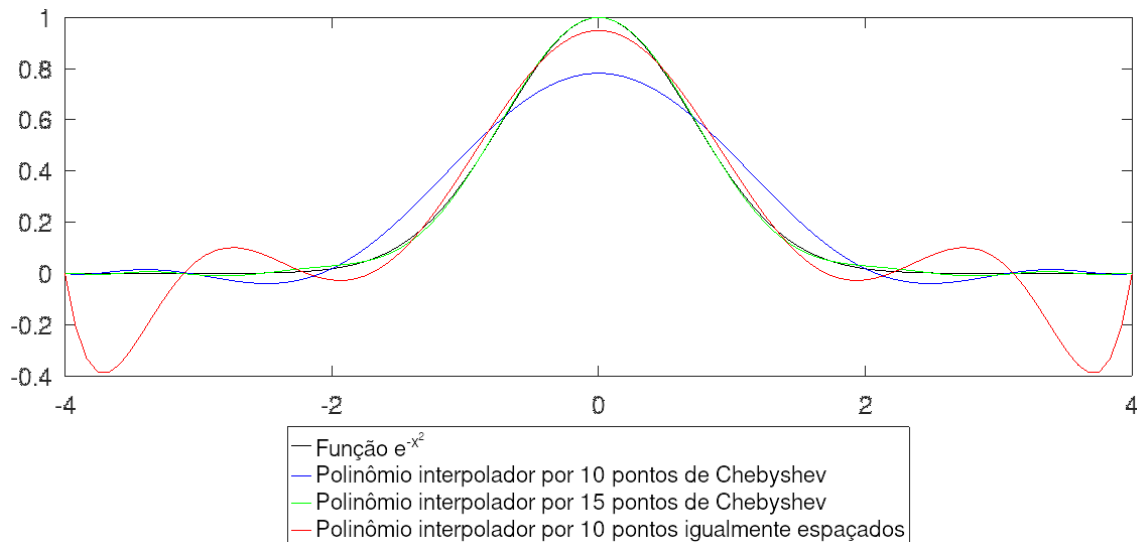
Note que ao aumentar o grau do polinômio interpolador, o denominador dessa função também aumenta, o que diminuiria o erro. Em contrapartida os termos $(x-x_0)(x-x_1) \cdots (x-x_n)$ também aumentam, então ao considerarmos um x perto de um dos extremos dos intervalos, os termos $(x-x_j)$ podem equivaler a um valor alto. Assim, ao invés de reduzirmos o erro máximo podemos acabar aumentando.

Essa situação pode ser revertida se ao invés de pontos igualmente espaçados escolhidos ao acaso, os escolhermos a partir de um critério. Esses novos pontos são conhecidos como pontos de Chebyshev e garantem convergência uniforme da função, embora não funcionem para toda função contínua.

Dessa forma, ainda considerando a função $f(x) = e^{-x^2}$ podemos obter o novo gráfico de aproximação mostrado a seguir pela Figura 9, dessa vez considerando os pontos de Chebyshev. Temos então em preto a própria função $f(x)$. Em azul, a

aproximação com 11 pontos de Chebyshev. Em vermelho, a aproximação em 10 pontos igualmente espaçados. Por fim, em verde, a aproximação em 15 pontos de Chebyshev.

Figura 9 – Comparativo dos gráficos dos polinômios de aproximação.



Fonte: **Autora.**

Note que o gráfico da aproximação em 15 pontos de Chebyshev é praticamente igual ao da função $f(x)$. Isso acontece porque enquanto o erro de aproximação cresce para pontos igualmente espaçados, ele decresce para pontos de Chebyshev, fornecendo uma aproximação muito melhor.

Os pontos de Chebyshev são raízes dos polinômios de Chebyshev, tais polinômios quando são de primeira ordem possuem grande importância na teoria da interpolação polinomial, pois fornecem como resultado da interpolação um minimizador para o fenômeno de Runge. Mais detalhes sobre esses polinômios podem ser encontradas em (FAIRES; BURDEN, 2004).

Os pontos ou nós de Chebyshev, em um intervalo $[a, b]$, são dados por

$$x_j = \frac{a+b}{2} + \frac{b-a}{2} \cos\left(\frac{(n-j)\pi}{n}\right), \quad j = 0, 1, \dots, n.$$

Quando o intervalo for $[-1, 1]$ teremos

$$x_j = \cos\left(\frac{(n-j)\pi}{n}\right), \quad j = 0, 1, \dots, n.$$

De fato, se temos uma escolha livre de pontos de interpolação, não é necessariamente uma boa escolha que eles sejam igualmente espaçados. Dado o intervalo $[-1, 1]$, divida-o de forma que para cada n , tenha-se

$$x_k = -1 + \frac{2k+1}{n+1}, \quad k = 0, \dots, n.$$

Esses pontos possuem um espaçamento de $\frac{2}{(n+1)}$ entre eles. Vimos no Teorema 3.0.1 que os pontos dessa forma não são a melhor escolha para todas as funções contínuas quando n se torna suficientemente grande.

4 INTERPOLAÇÃO POR SPLINES

Além dos pontos de Chebyshev, há mais uma maneira de evitarmos o fenômeno de Runge. Podemos restringir a solução para polinômios de grau mais baixo. Como não é sempre possível utilizar os nós de interpolação de Chebyshev, podemos dividir o problema inicial de interpolação em problemas menores.

Essa forma de aproximação por polinômios é muito conveniente, enquanto polinômios de grau mais alto tendem a apresentar grandes oscilações, a interpolação por splines também é uma forma de ligar pontos, mas é limitada a polinômios por partes de grau mais baixo apresentando menos variações e oscilações mínimas. Dessa forma, pode-se pensar que esse tipo de interpolação é uma boa solução para aquelas funções que quando aproximadas por outros métodos apresentam variações abruptas, e conseqüentemente, um erro de aproximação grande.

Os polinômios são funções interessantes e convenientes de utilizar para aproximação, uma vez que nos garantem propriedades importantes como a analiticidade, ou seja, é possível calcular suas derivadas e integrais.

O spline linear apresenta primeira derivada descontínua nos nós de interpolação. O quadrático, derivadas contínuas até ordem 1, além de bicos e alteração de curvatura nos nós. Por último, o cúbico - o mais popular - tem a primeira e segunda derivadas contínuas concedendo melhor suavidade nos nós do que os splines linear e quadrático.

De acordo com (CUNHA, 2000), a utilização de polinômios por partes permite escapar da analiticidade no intervalo inteiro, podendo conter discontinuidades das derivadas de ordem mais elevadas em alguns pontos. Esta característica de "pseudo-analiticidade" confere às funções polinomiais por partes, chamadas splines, boas propriedades de aproximação, convergência e estabilidade com respeito aos erros de arredondamento.

O termo splines surgiu de uma técnica de desenho onde o desenhista coloca uma folha de papel com pontos pré-estabelecidos sobre uma tábua de madeira e dispõe pinos nos pontos, então é desenhada uma curva suave passando por esses pinos utilizando uma régua perfeitamente flexível, chamada de spline. Para realizar o desenho da curva a régua é fixada em alguns pontos e encurvada numa posição de mínima energia, essa configuração é um spline cúbico. Assim, foi adotado o nome spline para os polinômios desse tipo.

Para um intervalo $[a, b]$ temos funções splines associadas a alguma partição predeterminada. Chamaremos essa partição de Π , definida pelos pontos x_0, x_1, \dots, x_m , tais que

$$\Pi : a = x_0 < x_1 < x_2 < \dots < x_{m-1} < x_m = b. \quad (3)$$

Os splines são polinômios que possuem todas as derivadas em cada um desses subintervalos (x_{i-1}, x_i) , para $i = 1, \dots, m$. Assim, neste processo estamos dividindo um

problema de interpolação maior em vários menores obtendo “pedaços” de polinômios, os quais colamos convenientemente de forma a garantir a existência das derivadas de ordem mais alta em todo o intervalo $[a, b]$.

Definição 4.0.1 *Uma função $s(x)$ é chamada spline de grau n , associada a uma partição de $[a, b]$ como definida em (3), se*

1. $s(x)$ é um polinômio de grau n em cada subintervalo (x_{i-1}, x_i) ;
2. $s(x)$ tem $(n-1)$ derivadas contínuas em cada x_i , isto é, $s(x)$ é uma função com $(n-1)$ derivadas em (a, b) .

4.1 BASES

Seja $\{\phi_i(x)\}_{i=1}^k$ uma base para o espaço vetorial de splines, então qualquer elemento deste espaço pode ser escrito pela combinação linear

$$s(x) = C_1\phi_1(x) + C_2\phi_2(x) + \dots + C_k\phi_k(x),$$

onde C_1, C_2, \dots, C_k são os coeficientes e k é a dimensão do espaço vetorial.

No caso de splines, as bases mais interessantes são as locais e geradoras dos splines, pois quanto mais bases locais, melhores resultados teremos. Essas bases fazem com que os sistemas de equações associados à discretização sejam esparsos e com melhores propriedades de estabilidade na sua solução numérica.

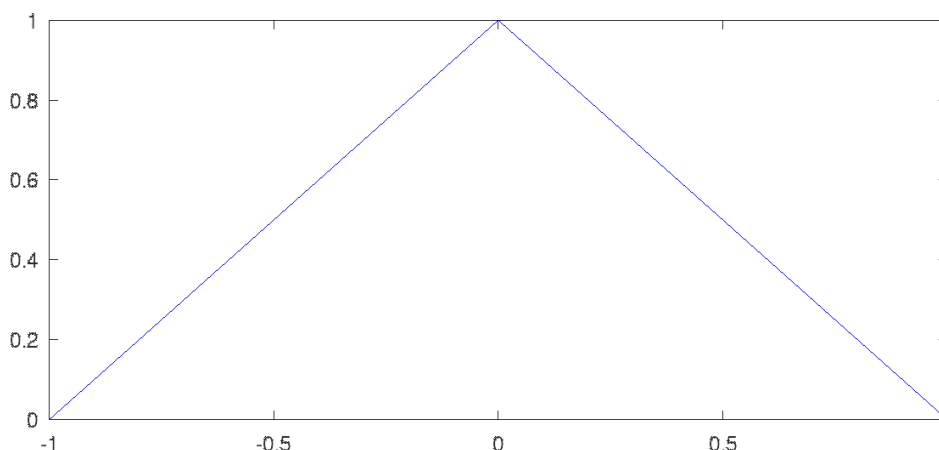
4.1.1 Bases para splines lineares

Seja $l_i(x)$ uma função definida para cada x_i nó de uma partição como na equação (3), dada por:

$$l_i(x) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}} & \text{se } x_{i-1} \leq x \leq x_i \\ \frac{x_{i+1} - x}{x_{i+1} - x_i} & \text{se } x_i \leq x \leq x_{i+1} \\ 0 & \text{se } x < x_{i-1} \text{ ou } x > x_{i+1}. \end{cases} \quad (4)$$

O conjunto das $l_i(x)$, $i = 0, \dots, m$, forma uma base para os splines lineares. Ao restringirmos as funções $l_i(x)$ a um intervalo $[a, b]$ arbitrário obtemos o gráfico dado pela Figura 10, chamadas de funções chapéu.

Figura 10 – Funções da base de splines lineares.



Fonte: **Autora.**

Através dessa base os splines lineares (de grau 1) associados à partição dadas por x_i , com $i = 0, \dots, m$, são definidos por:

$$s(x) = s_0 l_0(x) + s_1 l_1(x) + \dots + s_m l_m(x)$$

onde $s_i = s(x_i)$ são os valores que $s(x)$ assume em x_i .

Caso $x_{i+1} - x_i = h$, ou seja, a partição do intervalo é uniforme, a equação (4) torna-se a equação (5).

$$L(x) = \begin{cases} 1 + x & \text{se } x \in [-1, 0] \\ 1 - x & \text{se } x \in [0, 1] \\ 0 & \text{se } x \notin [-1, 1]. \end{cases} \quad (5)$$

4.1.2 Bases para splines cúbicos

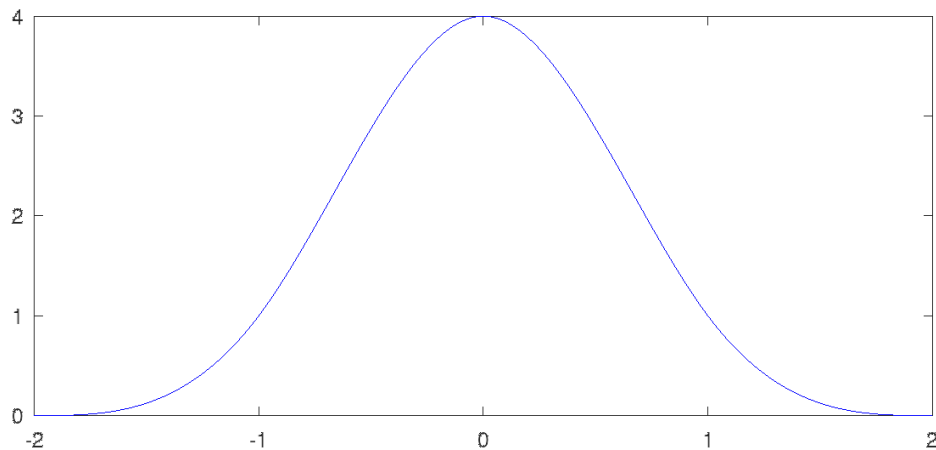
Pela definição 4.0.1, um spline cúbico consiste em “colar” pedaços de polinômios de grau 3, obtendo ao fim uma função $s(x)$ com duas derivadas contínuas em um intervalo arbitrário $[a, b]$.

Assuma que as partições do intervalo são uniformes, com subintervalos de tamanho h , isto é, $x_{i+1} - x_i = h$.

A função $B_i(x)$ é a base geradora dos splines cúbicos, que é dada pela equação (6) e possui como gráfico sinus com vértices em x_i como mostra a Figura 11.

$$B_i(x) = \frac{1}{h^3} \begin{cases} (x - x_{i-2})^3 & x_{i-2} \leq x \leq x_{i-1} \\ h^3 + 3h^2(x - x_{i-1}) + 3h(x - x_{i-1})^2 - 3(x - x_{i-1})^3 & x_{i-1} \leq x \leq x_i \\ h^3 + 3h^2(x_{i+1} - x) + 3h(x_{i+1} - x)^2 - 3(x_{i+1} - x)^3 & x_i \leq x \leq x_{i+1} \\ (x_{i+2} - x)^3 & x_{i+1} \leq x \leq x_{i+2} \\ 0 & x \notin [x_{i-2}, x_{i+2}]. \end{cases} \quad (6)$$

Figura 11 – Funções da base de splines cúbicos.



Fonte: **Autora.**

A partir da equação (6) obtemos a continuidade das duas primeiras derivadas de B_i . A tabela 3 mostrada a seguir apresenta os valores de $B_i(x)$, assim como suas derivadas em $x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2}$.

Tabela 3 – Valores de $B_i(x)$ e derivadas nos pontos da partição.

	x_{i-2}	x_{i-1}	x_i	x_{i+1}	x_{i+2}
B_i	0	1	4	1	0
B'_i	0	$3/h$	0	$-3/h$	0
B''_i	0	$6/h^2$	$-12/h^2$	$6/h^2$	0

Note que ao dividirmos o intervalo total obtemos subintervalos arbitrários (x_{i-1}, x_i) , teremos em cada um deles um polinômio $B_i(x)$ de grau 3, e se fizermos uma combinação linear dessas funções manteremos essa propriedade. Dessa forma, a função spline de grau 3 é definida pela equação (7) a seguir.

$$s_3(x) = C_{-1}B_{-1}(x) + C_0B_0(x) + C_1B_1(x) + \dots + C_mB_m(x) + C_{m+1}B_{m+1}(x). \quad (7)$$

De acordo com (CUNHA, 2000), como na base dos splines lineares as funções se anulam fora de intervalos de comprimento $2h$, as funções da base dos splines cúbicos são menos locais, posto que elas se anulam fora de intervalos de comprimento $4h$. Isto se reflete na esparsidade das matrizes dos sistemas que surgem em aplicações. Entretanto, ganhamos mais suavidade nas funções geradas, ao passar das bases dos splines lineares para as bases cúbicas.

Considerando

$$B(x) = \begin{cases} (x+2)^3 & -2 \leq x \leq -1 \\ 1 + 3(x+1) + 3(x+1)^2 - 3(x+1)^3 & -1 \leq x \leq 0 \\ 1 + 3(1-x) + 3(1-x)^2 - 3(1-x)^3 & 0 \leq x \leq 1 \\ (2-x)^3 & 1 \leq x \leq 2 \\ 0 & x \notin [-2, 2]. \end{cases}$$

É possível verificar que

$$B_i(x) = B\left(\frac{x-x_0}{h} - i\right).$$

4.1.3 Bases genéricas

4.2 INTERPOLAÇÃO POR PARTES

O uso das funções splines na interpolação polinomial por partes se dá através do uso dos nós da partição dos splines como pontos de interpolação.

Considere a interpolação linear por partes da função $f(x)$ nos pontos x_i , para $i = 0, \dots, n$, queremos encontrar a_i tais que

$$s_1(x) = \sum_{i=0}^n a_i l_i(x) \quad \text{e} \quad s_1(x_j) = f(x_j) = f_j, \quad j = 0, \dots, n$$

O somatório envolve spline linear, que como vimos anteriormente, podem ser gerados por bases e escritos em função delas, enquanto a segunda expressão diz respeito às condições de interpolação. Lembrando que $l_i(x_j) = 0$, se $i \neq j$ e $l_i(x_i) = 1$. Temos, enfim, ao aplicar a expressão que define $s_1(x)$ nos pontos de interpolação x_i ,

$$s_1(x_i) = a_i = f(x_i) = f_i.$$

O spline linear interpolador de f_0, f_1, \dots, f_n é dado por

$$s_1(x) = \sum_{i=0}^n f_i l_i(x).$$

Para os splines cúbicos queremos encontrar a_i de forma que:

$$s_3(x) = \sum_{i=-1}^{n+1} a_i B_i(x) \quad \text{e} \quad s_3(x_j) = f(x_j) = f_j, \quad j = 0, \dots, n$$

com B_i sendo a base dos splines cúbicos. Da seção 4.1.2 sabemos que, para x_j arbitrário, $B_{j-1}(x_j)$, $B_j(x_j)$ e $B_{j+1}(x_{j+1})$ são os únicos valores da base diferentes de zero. Temos então:

$$s_3(x_j) = a_{j-1} B_{j-1}(x_j) + a_j B_j(x_j) + a_{j+1} B_{j+1}(x_j). \quad (8)$$

Pela Tabela 3, temos que $B_{j-1}(x_j) = 1$, $B_j(x_j) = 4$, $B_{j+1}(x_j) = 1$, e tendo em vista que $s_3(x_j) = f_j$, obtemos um sistema que deve ser satisfeito pelos a_i :

$$a_{j-1} + 4a_j + a_{j+1} = f(x_j) = f_j, \quad j = 0, \dots, n$$

Assim, temos um sistema de $n+1$ equações e $n+3$ incógnitas, $a_{-1}, a_0, \dots, a_n, a_{n+1}$. Para garantir que o sistema possua única solução impomos algumas condições adicionais nos extremos dos splines, para as quais existem várias possíveis escolhas. Considere

$$s'_3(x_0) = f'(x_0) \quad \text{e} \quad s'_3(x_n) = f'(x_n), \quad (9)$$

ou então

$$s''_3(x_0) = 0 \quad \text{e} \quad s''_3(x_n) = 0. \quad (10)$$

Na primeira opção temos uma condição que é conhecida como condição de engaste. Neste caso, juntamos ao sistema o cálculo $s'_3(x_0)$ e $s'_3(x_n)$ usando as equações dadas em (9) e a tabela 3, obtendo

$$-3a_{-1} + 3a_1 = hf'(x_0) \quad \text{e} \quad -3a_{n-1} + 3a_{n+1} = hf'(x_n).$$

Na segunda opção, temos a condição de fronteira natural, onde as equações adicionais do sistema são dadas pelas equações descritas em (10), e utilizamos a equação (8) para o cálculo da segunda derivada e a tabela 3, obtendo

$$a_{-1} - 2a_0 + a_1 = 0 \quad \text{e} \quad a_{n-1} - 2a_n + a_{n+1} = 0.$$

Em uma terceira opção basta supormos suavidade na cauda da função. Neste caso, podemos obter condições adicionais independentes das condições de contorno exigindo, em $x = x_1$ e $x = x_{n-1}$, a continuidade da terceira derivada.

Por fim, considerando o caso da expressão da base dos splines ser de difícil manipulação e dos pontos de interpolação não serem igualmente espaçados, podemos aplicar a definição de splines cúbicos para interpolar f_i , $i = 0, \dots, n$, impondo a continuidade das derivadas. Neste caso, o Teorema 4.2.1 a seguir é bastante útil para encontrar o polinômio interpolador.

Teorema 4.2.1 *Chamemos*

$$x_i - x_{i-1} = h_i, \quad \frac{f_i - f_{i-1}}{h_i} = d_i \quad \text{e} \quad \frac{x - x_{i-1}}{h_i} = t,$$

para $x_{i-1} \leq x \leq x_i$. O spline cúbico, com nós em x_i , $i = 0, \dots, n$, tal que $s_3(x_i) = f_i$, $i = 0, \dots, n$, tem como expressão no intervalo $[x_{i-1}, x_i]$, $i = 1, \dots, n$

$$q_i(x(t)) = tf_i + (1-t)f_{i-1} + h_it(1-t)[(k_{i-1} - d_i)(1-t) - (k_i - d_i)t], \quad (11)$$

onde k_0, k_1, \dots, k_n satisfazem ao sistema tridiagonal

$$h_{i+k}k_{i-1} + 2(h_i + h_{i+1})k_i + h_i k_{i+1} = 3(h_i d_{i+1} + h_{i+1} d_i), \quad (12)$$

para $i = 1, \dots, n-1$.

Demonstração: Inicialmente vamos verificar as condições de interpolação e a continuidade de $s_3(x)$.

Da equação (11) temos que $q_i(x)$ é um polinômio de grau 3. Visto isso, também temos que $x = x_{i-1}$ equivale a $t = 0$, da mesma forma que $x = x_i$ equivale a $t = 1$, substituindo isso na equação (11), obtemos

$$q_i(x_{i-1}) = f_{i-1} \quad \text{e} \quad q_i(x_i) = f_i,$$

assim, $q_i(x_i) = q_{i+1}(x_i) = f_i$. Dessa forma, verificamos a continuidade de $s_3(x)$.

Para verificar a continuidade da primeira derivada, derivamos (11), usando $d_i = \frac{f_i - f_{i-1}}{h_i}$ e pela regra da cadeia:

$$q'_i(x(t)) = d_i + (1-2t)[(k_{i-1} - d_i)(1-t) - (k_i - d_i)] + t(1-t)[2d_i - k_{i-1} - k_i].$$

Sendo $t = 0$ e $t = 1$, obtemos:

$$q'_i(x_{i-1}) = k_{i-1} \quad \text{e} \quad q'_i(x_i) = k_i.$$

Logo, $q'_{i+1}(x_i) = q'_i(x_i)$ comprovando a continuidade da primeira derivada.

Por último, queremos verificar a continuidade da segunda derivada que é equivalente ao sistema tridiagonal (12). Portanto, aplicando a regra da cadeia em $q'_i(x(t))$, segue que

$$q''_i(x(t)) = \frac{2}{h_i} [-(k_{i-1} - d_i)(1-t) + (k_i - d_i)t + (1-2t)(2d_i - k_{i-1} - k_i)];$$

que nos dá $q''_i(x_i) = q''_{i+1}(x_i)$ que corresponde ao sistema tridiagonal (12).

Logo, a última condição de spline cúbico está provada, a “colagem” dos pedaços de polinômios $q_i(x)$ preserva a continuidade da segunda derivada. ■

Apenas essas condições não são suficientes para determinar a função interpoladora, pois o sistema tridiagonal (12) possui $n-1$ equações e $n+1$ incógnitas. Assim, são necessárias mais duas condições, sendo elas as equações (9) e (10). Em (9), temos como condições adicionais

$$k_0 = f'(x_0) \quad \text{e} \quad k_n = f'(x_n),$$

enquanto para (10), temos

$$2k_0 + k_1 = 3d_1 \quad \text{e} \quad k_{n-1} + 2k_n = 3d_n.$$

4.3 EXEMPLOS

Os exemplos a seguir foram adaptados de (CHAPRA, 2013) e ilustram o uso dos splines para interpolação. Foram resolvidos através do software Matlab, e seus códigos solução estão disponíveis no Apêndice B.

Exemplo 4.3.1 A função de Runge é escrita como

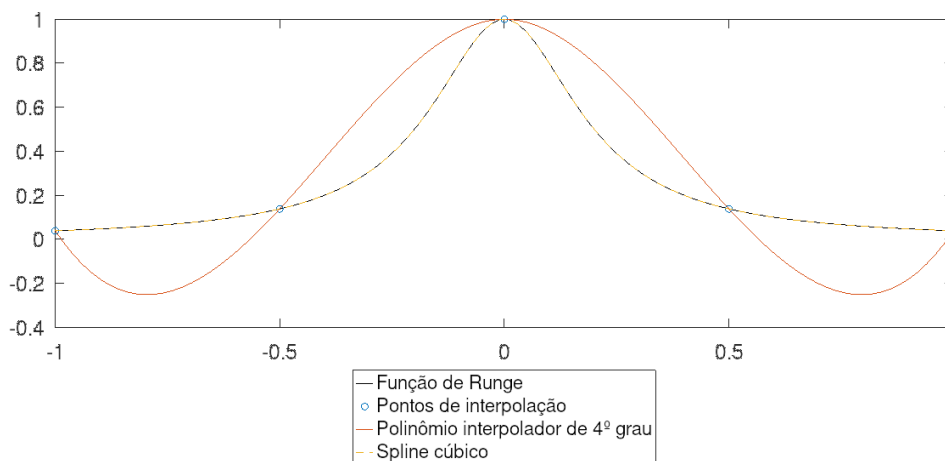
$$f(x) = \frac{1}{1 + 25x^2}$$

Gere cinco valores igualmente espaçados dessa função ao longo do intervalo $[-1, 1]$. Ajuste esses dados com **(a)** um polinômio de quarto grau e **(b)** splines cúbicos.

Para realizarmos a interpolação da função $f(x)$ para um polinômio de quarto grau foi utilizada a mesma ideia empregada nos exemplos do capítulo 2. São gerados cinco pontos igualmente espaçados ao longo do intervalo $[-1, 1]$ e via resolução de sistema linear encontramos o polinômio interpolador. Para o caso da interpolação via splines cúbicos, o Matlab possui uma função nativa que leva o próprio nome da interpolação, assim o ajuste de dados é feito automaticamente com o uso do comando “splines”.

Dessa forma, é gerado um gráfico comparativo dado pela Figura 12.

Figura 12 – Interpolação via polinômio de quarto grau e via splines cúbicos.



Fonte: **Autora.**

Note como o ajuste via polinômio de quarto grau oscila mais devido ao fenômeno de Runge, enquanto no ajuste por spline cúbico temos maior controle e interpolação coincide visualmente com a própria função de Runge.

Exemplo 4.3.2 As funções de Bessel aparecem com frequência em análises avançadas de engenharia tais como o estudo de campos elétricos. Essas funções geralmente

não são passíveis de avaliação simples e, portanto, são frequentemente compiladas em tabelas matemáticas padrões. Por exemplo,

Tabela 4 – Função de Bessel avaliada em alguns pontos dados.

x	1,8	2	2,2	2,4	2,6
$J_1(x)$	0,5815	0,5767	0,556	0,5202	0,4708

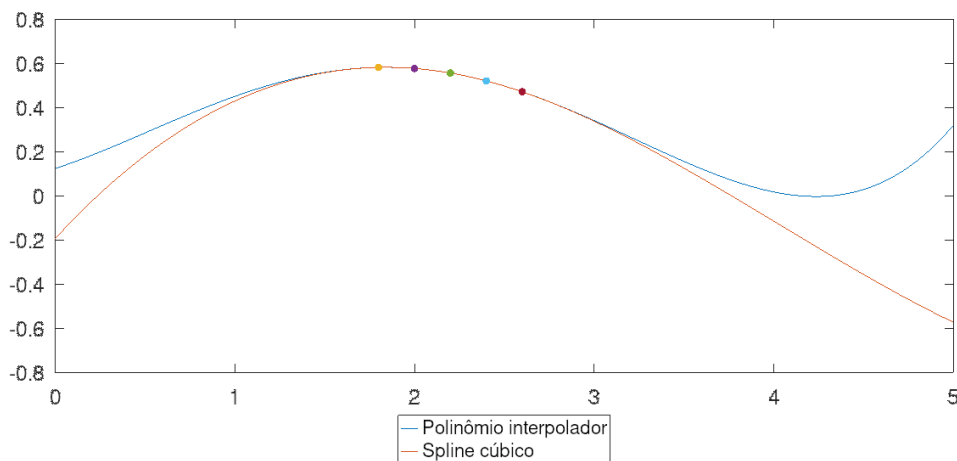
Efetue uma interpolação com os pontos dados **(a)** utilizando um polinômio interpolador e **(b)** utilizando splines cúbicos.

A função de Bessel é um série de potência de infinitos termos de difícil avaliação, por esse motivo geralmente é apresentada através de um tabelamento. Dado um conjunto de valores como a tabela 4, podemos aproximar a função via interpolação e avaliá-la para algum x não dado. Mais detalhes sobre esse tipo de função estão disponíveis em (WIMP, 1962) e (HITCHCOCK, 1957).

Para resolver esse problema empregamos a mesma ideia do exemplo 4.3.1, porém agora temos cinco pontos dados no enunciado, e não é preciso gerar pontos aleatórios igualmente espaçados dentro de um intervalo dado.

Assim, obtemos então a Figura 13 através do Matlab, ilustrando um comparativo de interpolações via polinômio e via splines cúbicos.

Figura 13 – Função de Bessel interpolada via polinômios e via splines cúbicos.



Fonte: **Autora.**

Note que assim como no exemplo anterior a interpolação via polinômio apresenta mais oscilações do que através de splines cúbicos.

5 CONCLUSÃO

A interpolação surge da necessidade de aproximar o valor de uma função a partir de pontos dados, possibilitando descobrir o seu valor em um ponto não dado. Para o caso particular da interpolação polinomial, frequentemente usado para aproximação de funções contínuas, sabemos que existe um único polinômio que interpola essa função. Para isso existem diversos métodos, dentre eles o de Lagrange, diferenças divididas de Newton, e seu caso particular para pontos igualmente espaçados, conhecido como método de Gregory-Newton.

Como qualquer aproximação, esses métodos são passíveis de erros. Assim, pelo erro de truncamento é possível analisar se a aproximação realizada é válida ou não. Conseqüentemente, em alguns casos específicos esses métodos podem não funcionar da maneira esperada, podendo apresentar significativos erros de aproximação e, quanto maior o grau do polinômio interpolador, maior será o erro, principalmente nas bordas do intervalo onde a função está definida. Esse problema é conhecido como fenômeno de Runge. A fim de evitá-lo podemos fazer uma escolha específica de pontos de interpolação, conhecida como nós de Chebyshev, ou dividir o problema inicial de interpolação em problemas menores, ou seja, realizando uma interpolação por partes ou por splines.

Sendo assim, temos uma revisão teórica completa dos métodos de interpolação propostos, assim como a implementação desses através de ferramentas computacionais para solução de alguns problemas. Para continuidade desse trabalho é sugerido um estudo sobre outras formas de aproximação que não seja via interpolação polinomial, e aplicações dos métodos estudados para otimização e processamento de imagens.

REFERÊNCIAS

ATKINSON, Kendall E. **An introduction to numerical analysis**. [S.l.]: John Wiley & sons, 2008.

BROWN, James Ward; CHURCHILL, Ruel V. **Variáveis complexas e aplicações**. [S.l.]: McGraw Hill Brasil, 2015.

CHAPRA, Steven C. **Métodos Numéricos Aplicados com MATLAB® para Engenheiros e Cientistas**. 3ª edição. [S.l.]: AMGH Editora, 2013.

CUNHA, M Cristina C. **Métodos numéricos**. [S.l.]: Editora da UNICAMP, 2000.

FAIRES, J Douglas; BURDEN, Richard L. **Métodos numéricos**. [S.l.]: Thomson, 2004.

HANDSCOMB, David Christopher. **Methods of numerical approximation: lectures delivered at a Summer School held at Oxford University, September 1965**. [S.l.]: Elsevier, 2014.

HITCHCOCK, AJM. Polynomial approximations to Bessel functions of order zero and one and to related functions. **Mathematics of Computation**, v. 11, n. 58, p. 86–88, 1957.

HOFFMAN, Kenneth. **Linear algebra**. [S.l.]: Englewood Cliffs, NJ, Prentice-Hall, 1971.

STEWART, James. **Cálculo: volume 1**. [S.l.]: Thomson, 2008.

WIMP, Jet. Polynomial expansions of Bessel functions and some associated functions. **Mathematics of Computation**, JSTOR, v. 16, n. 80, p. 446–458, 1962.

APÊNDICE A – MATRIZ DE VANDERMONDE

Este apêndice tem o intuito de complementar a demonstração do Teorema 2.1.1 de modo a não interferir no corpo do trabalho.

Teorema A.0.1 *Toda matriz de Vandermonde com $x_i \neq x_j$, para $i \neq j$ é não singular.*

Demonstração: Primeiramente suponha $W = \begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{n-1} & x_2^{n-1} & \dots & x_n^{n-1} \end{pmatrix}$ uma matriz de Vandermonde arbitrária.

Queremos provar que toda matriz de Vandermonde da forma $x_i \neq x_j$, para $i \neq j$ é não singular, que é o mesmo que provar que

$$\det(W) = \prod_{n \geq k > j \geq 1} (x_k - x_j), \forall n \geq 2.$$

Faremos isso via indução na dimensão da matriz.

CASO BASE: $n = 2$

Temos $W = \begin{pmatrix} 1 & 1 \\ x_1 & x_2 \end{pmatrix} \Rightarrow \det(W) = x_2 - x_1$.

Mas $x_1 \neq x_2$, então $x_2 - x_1 \neq 0$. Assim, está provado para $n = 2$.

HIPÓTESE DE INDUÇÃO: Seja $W_{n+1} = \begin{pmatrix} 1 & 1 & \dots & 1 & 1 \\ x_1 & x_2 & \dots & x_n & x_{n+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_1^{n-1} & x_2^{n-1} & \dots & x_n^{n-1} & x_{n+1}^{n-1} \\ x_1^n & x_2^n & \dots & x_n^n & x_{n+1}^n \end{pmatrix}$ matriz arbitrária, e suponha, por hipótese de indução, que

$$\det(W_n) = \begin{vmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{n-1} & x_2^{n-1} & \dots & x_n^{n-1} \end{vmatrix} = \prod_{n \geq k > j \geq 1} (x_k - x_j).$$

Primeiramente, aplicando expansão por cofatores para calcular o determinante, obtemos $\det(W_{n+1}) = x_{n+1}^n \det(W_n) + \dots$. Dessa forma, é possível notar que $\det(W_{n+1})$ é dado por um polinômio que possui grau n em x_{n+1} .

Agora, é importante ver que quando $x_{n+1} = x_i$, para $i = 1, 2, \dots, n$, o polinômio se anula. Isso acontece pois a matriz W_{n+1} possui colunas iguais, o que por definição anula o determinante.

Por fim, decompondo $\det(W_{n+1})$ de forma que o coeficiente líder seja $\det(W_n)$, obtemos:

$$\det(W_{n+1}) = \det(W_n)(x_{n+1} - x_n)(x_{n+1} - x_{n-1}) \cdots (x_{n+1} - x_1) = \prod_{n+1 \geq k > j \geq 1} (x_k - x_j).$$

Dessa forma, garantido pelo princípio de indução matemática, obtemos que $\det(W_n) = \prod_{n \geq k > j \geq 1} (x_k - x_j)$, $\forall n \geq 2$.

Como $\det(W_n) \neq 0$, pois da hipótese temos que $x_i \neq x_j$, para $i \neq j$. Conclui-se que qualquer matriz de Vandermonde da forma $x_i \neq x_j$, para $i \neq j$, é não singular como queríamos.

■

APÊNDICE B – CÓDIGOS

Esse apêndice é destinado aos códigos desenvolvidos em Matlab e utilizados para gerar todas as imagens do trabalho.

B.1 CÓDIGOS REFERENTES AO CAPÍTULO 2

A seguir estão os códigos utilizados no Capítulo 2.

A fim de gerar a Figura 1 do Exemplo 2.5.1, foi criado o código B.1 que através da resolução de um sistema linear encontra o polinômio interpolador, e posteriormente plota o polinômio e os pontos que foram interpolados.

```

1 disp('A. Calcule o polinomio que interpola os pontos (1,2), (-2,4) e
   (0,1)')
2 A = [ 1 1 1; 1 -2 -2; 1 0 0]
3 b = [2; 4; 1]
4 x = A\b
5
6 disp('B. Plote os pontos e o polinomio')
7 y = linspace(-5,5,1000);
8 p = 1+(1/6)*y+(5/6)*y.^2;
9 plot(y,p)
10 hold on
11 plot(1,2, '.', 'MarkerSize', 20)
12 plot(-2,4, '.', 'MarkerSize', 20)
13 plot(0,1, '.', 'MarkerSize', 20)
14 hold off

```

Listing B.1 – Código gerador da Figura 1.

Para o Exemplo 2.5.2, o código gerador foi o B.2, que segue a mesma estrutura do exemplo anterior, porém com mais pontos, gerando a Figura 2.

```

1 disp('A. Encontre o polinomio interpolador dos pontos (0,1), (1,6),
   (2,5) e (3,-8).')
2 A = [1 0 0 0; 1 1 1 1; 1 2 4 8; 1 -8 64 -512]
3 b = [1; 6; 5; -8] x = A\b
4
5 disp('B. Plote os pontos e o polinomio.')
6 x = linspace(-10,10,1000);
7 p = 1+6*x-x.^3;
8 plot(x,p)
9 hold on
10 plot(0,1, '.', 'MarkerSize', 20)
11 plot(1,6, '.', 'MarkerSize', 20)
12 plot(2,5, '.', 'MarkerSize', 20)
13 plot(3,-8, '.', 'MarkerSize', 20)

```



```
14 hold off
```

Listing B.2 – Código gerador da Figura 2.

Por fim, temos o Exemplo 2.5.3 que, assim como os exemplos anteriores, possui a mesma estrutura de código, porém com 9 pontos, resultando em um polinômio de grau 8 gerando a Figura 3.

```
1 disp('A. Encontre o polinomio interpolador dos pontos (0,0), (1,6),
    (-1,5), (2,4), (-2,10), (3,-8), (-3,15), (4,20) e (-4,12).')
```

```
2
```

```
3 A = [1 0 0 0 0 0 0 0 0 0;
4 1 1 1 1 1 1 1 1 1;
5 1 -1 1 -1 1 -1 1 -1 1 -1;
6 1 2 4 8 16 32 64 128 256 512;
7 1 -2 4 -8 16 -32 64 -128 256 -512;
8 1 3 9 27 81 243 729 2187 6561 19683;
9 1 -3 9 -27 81 -243 729 -2187 6561 -19683;
10 1 4 16 64 256 1024 4096 16384 65536 262144;
11 1 -4 16 -64 256 -1024 4096 -16384 65536 -262144]
12 b = [0; 6; 5; 4; 10; -8; 15; 20; 12]
13 format short
14 x = A\b
15
16
```

```
17 disp('B. Plote os pontos e o polinomio.')
```

```
18 x = linspace(-4,4,1000);
19 p = 9.9996e-15 + 5.2054e-01*x + 7.4603e+00*x.^2 + 2.5982e-01*x.^3 -
    2.1625e+00*x.^4 - 3.1405e-01*x.^5 + 2.0833e-01*x.^6 + 3.4694e-02*x.^7
    - 6.1508e-03*x.^8 - 9.9775e-04*x.^9
20 plot(x,p)
21 hold on
22 plot(0,0, '.', 'MarkerSize',20)
23 plot(1,6, '.', 'MarkerSize',20)
24 plot(-1,5, '.', 'MarkerSize',20)
25 plot(2,4, '.', 'MarkerSize',20)
26 plot(-2,10, '.', 'MarkerSize',20)
27 plot(3,-8, '.', 'MarkerSize',20)
28 plot(-3,15, '.', 'MarkerSize',20)
29 plot(4,20, '.', 'MarkerSize',20)
30 plot(-4,12, '.', 'MarkerSize',20)
31 hold off
```

Listing B.3 – Código gerador da Figura 3.

B.2 CÓDIGOS REFERENTES AO CAPÍTULO 3

Para gerar as Figuras 4, 5 e 6 do Capítulo 3, foi criado o código B.4 que primeiramente gera a imagem da referente a função e^{-x^2} e depois suas interpolações com polinômios de grau 6 e 10.

```

1 f = @(x) exp(-x.^2);           # Funcao
2 xx = linspace(-4,4,100);      # Malha densa para exibicao dos graficos
3 plot(xx, f(xx), 'k');
4 legend({"Funcao e^{{-x}}^2"}, "location", "southoutside", "fontsize", 15)
5
6 pause
7
8 n=6;
9 x = linspace(-4,4,n+1);      # Malha regular
10 y = f(x);                    # Avaliacao de f sobre a malha
11 p = polyfit(x,y,n);          # Determinacao do polinomio interpolador
12 yy1 = polyval(p,xx);         # Avaliacao do polinomio interpolador
13
14 plot(xx, f(xx), 'k');
15 hold on
16 plot(xx,yy1,'r');
17 legend({"Funcao e^{{-x}}^2", "Polinomio interpolador de grau 6"}, "
    location", "southoutside", "fontsize", 15);
18
19 pause
20
21 n=10;
22 x = linspace(-4,4,n+1);      # Malha regular
23 y = f(x);                    # Avaliacao de f sobre a malha
24 p = polyfit(x,y,n);          # Determinacao do polinomio interpolador
25 yy2 = polyval(p,xx);         # Avaliacao do polinomio interpolador
26
27 plot(xx, f(xx), 'k');
28 hold on
29 plot(xx,yy1,'r');
30 hold on
31 plot(xx,yy2,'b');
32
33 legend({"Funcao e^{{-x}}^2", "Polinomio interpolador de grau 6", "
    Polinomio interpolador de grau 10"}, "location", "southoutside", "
    fontsize", 15);

```

Listing B.4 – Código gerador para fenômeno de Runge na interpolação da função e^{-x^2} .

No Teorema 3.0.1 temos a Figura 7, que apresenta um gráfico da função $g(x) = \frac{1}{1+25x^2}$ e a interpolação dessa por polinômio de grau 4, 6 e 10. Para isso foi utilizado o

código B.5 a seguir:

```

1 f = @(x) 1./(1+25*x.^2); # Funcao
2 xx = linspace(-1,1,500); # Malha densa para exibicao dos graficos
3 plot(xx, f(xx), 'k');
4 hold on
5
6 n=7;
7 x = linspace(-1,1,n+1); # Malha regular
8 y = f(x); # Avaliacao de f sobre a malha
9 p = polyfit(x,y,n); # Determinacao do polinomio interpolador
10 yy3 = polyval(p,xx); # Avaliacao do polinomio interpolador
11 plot(xx, yy3, 'g');
12
13 n=15;
14 x = linspace(-1,1,n+1); # Malha regular
15 y = f(x); # Avaliacao de f sobre a malha
16 p = polyfit(x,y,n); # Determinacao do polinomio interpolador
17 yy5 = polyval(p,xx); # Avaliacao do polinomio interpolador
18 plot(xx, yy5, 'b');
19
20 legend({"Funcao 1/(1+25x^2)", "Polinomio interpolador de 8 grau", "
    Polinomio interpolador de 16 grau"}, "location", "southoutside", "
    fontsize", 15);

```

Listing B.5 – Código gerador para fenômeno de Runge na função $\frac{1}{1+25x^2}$.

Para implementar a interpolação através dos pontos de Chebyshev da Figura 8, utilizamos o código B.6 que encontra os pontos de interpolação e realiza a interpolação para polinômios de grau 8 e 16.

```

1 g = @(x) 1./(1+25*x.^2);
2 c = -1;
3 d = 1;
4 xx = linspace(c,d,300);
5 plot(xx, g(xx), 'k');
6
7 hold on
8
9 a = -1;
10 b = 1;
11
12 f = @(j,n) (((a+b)./2 + (b-a)./2) * cos(((n-j)./n)*pi));
13
14
15 grau = 8;
16 for var = 0:grau;
17 x1(var+1) = f(var,grau);
18 y1(var+1) = g(x1(var+1));

```

```

19 end
20 p = polyfit(x1,y1,grau);           # Determinacao do polinomio
    interpolador
21 yy1 = polyval(p,xx);             # Avaliacao do polinomio
    interpolador
22 plot(xx,yy1,'g');
23
24
25 grau = 16;
26 for var = 0:grau;
27     x2(var+1) = f(var,grau);
28     y2(var+1) = g(x2(var+1));
29 end
30 p = polyfit(x2,y2,grau);         # Determinacao do polinomio
    interpolador
31 yy2 = polyval(p,xx);             # Avaliacao do polinomio interpolador
32 plot(xx,yy2,'b');
33
34 legend({"Funcao  $1/(1+25x^2)$ ", "Polinomio interpolador de grau 8 por
    pontos de Chebyshev", "Polinomio interpolador de grau 16 por pontos
    de Chebyshev"}, "location", "southoutside", "fontsize", 15);

```

Listing B.6 – Código gerador da interpolação por pontos de Chebyshev para função de Runge.

A Figura 9 que realiza um comparativo de interpolações via pontos de Chebyshev e pontos igualmente espaçados foi gerada através do código B.7 a seguir.

```

1 g = @(x) exp(-x.^2);
2 a = -4;
3 b = 4;
4 xx = linspace(a,b,100);
5 plot(xx, g(xx), 'k');
6
7 hold on
8
9 f = @(j,n) (((a+b)./2 + (b-a)./2) * cos(((n-j)./n)*pi));
10
11 grau = 9;
12 for var = 0:grau;
13     x1(var+1) = f(var,grau);
14     y1(var+1) = g(x1(var+1));
15 end
16 p = polyfit(x1,y1,grau);         # Determinacao do polinomio
    interpolador
17 yy1 = polyval(p,xx);             # Avaliacao do polinomio interpolador
18 plot(xx,yy1,'b');
19
20 hold on

```

```

21
22 grau = 14;
23 for var = 0:grau;
24     x2(var+1) = f(var,grau);
25     y2(var+1) = g(x2(var+1));
26 end
27 p = polyfit(x2,y2,grau);           # Determinacao do polinomio
    interpolador
28 yy2 = polyval(p,xx);             # Avaliacao do polinomio
    interpolador
29 plot(xx,yy2,'g');
30
31 hold on
32
33 grau=9;
34 x3 = linspace(a,b,grau+1);       # Malha regular
35 y3 = g(x3);                       # Avaliacao de f sobre a malha
36 p = polyfit(x3,y3,grau);         # Determinacao do polinomio
    interpolador
37 yy3 = polyval(p,xx);
38 plot(xx,yy3,'r');
39
40
41 legend({"Funcao e^{-x^2}", "Polinomio interpolador por 10 pontos de
    Chebyshev", "Polinomio interpolador por 15 pontos de Chebyshev", "
    Polinomio interpolador por 10 pontos igualmente espaçados"}, "location
    ", "southoutside", "fontsize", 15);

```

Listing B.7 – Código gerador do comparativo de interpolação por pontos de Chebyshev e pontos igualmente espaçados para a função e^{-x^2} .

B.3 CÓDIGOS REFERENTES AO CAPÍTULO 4

A base de splines lineares possui como gráfico uma função chapéu, na Figura 10 o intervalo dado é $[-1, 1]$, mas o código B.8 a seguir foi construído como uma função onde o usuário escolhe o intervalo para gerar o gráfico.

```

1 function blinear(a,b)
2
3     for i = a:1:(b-1)
4         if mod(i,2) == 1
5             x = linspace(i,i+1,1000);
6             l = (x - i);
7             plot(x,l,'b')
8             hold on
9         else
10            x = linspace(i,i+1,1000);

```

```

11     l = (i - x) + 1;
12     plot(x,l,'b')
13     hold on
14     endif
15
16     endfor
17
18 end

```

Listing B.8 – Função geradora da base de splines lineares.

Para a Figura 11 foi utilizado o código B.9 que gera o gráfico da base de splines cúbicos para o intervalo $[-1, 1]$.

```

1 x = linspace(-2,-1,1000);
2 l = (x+2).^3;
3 plot(x,l,'b')
4 hold on
5
6 x = linspace(-1,0,1000);
7 l = (1 + 3*(x + 1) + 3*(x + 1).^2 - 3*(x + 1).^3);
8 plot(x,l,'b')
9 hold on
10
11 x = linspace(0,1,1000);
12 l = (1 + 3*(1 - x) + 3*(1 - x).^2 - 3*(1 - x).^3);
13 plot(x,l,'b')
14 hold on
15
16 x = linspace(1,2,1000);
17 l = (2-x).^3;
18 plot(x,l,'b')
19 hold on

```

Listing B.9 – Função geradora da base de splines cúbicos.

No Exemplo 4.3.1 foi realizado um comparativo de interpolação via pontos igualmente espaçados e via splines cúbicos. Para pontos igualmente espaçados foi utilizado a mesma estrutura dos exemplos anteriores, mas via splines cúbicos foi utilizado uma função nativa do Matlab que realiza a interpolação da forma desejada. Dessa forma, o código B.10 gera a Figura 12.

```

1 % Funcao Runge
2 f = @(x) 1./(1+25*x.^2);
3 x = linspace(-1,1);
4 plot(x, f(x), 'k')
5 hold on
6
7 % Gerando cinco pontos igualmente espaçados

```

```

8 xx = linspace(-1,1,5);
9
10 % Interpolacao por spline
11 z = spline(xx,f(xx),x);
12
13 plot(xx,f(xx),'o',x,z,x,f(x),'--')
14 legend({"Funcao de Runge", "Pontos de interpolacao", "Polinomio
interpolador de grau 4", "Spline cubico"}, "location", "southoutside",
"fontsize", 15);

```

Listing B.10 – Código gerador do comparativo de interpolação via polinômio de quarto grau e via splines cúbicos para função de Runge.

O Exemplo 4.3.2 se resolve através da Figura 13, gerada pelo código B.11, onde calculamos o polinômio interpolador utilizando a mesma estrutura dos exemplos do capítulo 2, realizamos a interpolação polinomial e via splines cúbicos, e plotamos os pontos dados inicialmente para verificar se as interpolações de fato passam por eles.

```

1 % Calculando o polinomio interpolador
2 A = [1 1.8 3.24 5.832 10.4976;
3 1 2 4 8 16;
4 1 2.2 4.84 10.648 23.4256;
5 1 2.4 5.76 13.824 33.1776;
6 1 2.6 6.76 17.576 45.6976];
7
8 b = [0.5815; 0.5767; 0.556; 0.5202; 0.4708];
9
10 x = A\b
11
12 y = linspace(0,5);
13
14 p = 0.123700 + 0.262958*y + 0.181771*y.^2 - 0.136458*y.^3 + 0.018229*y
.^4;
15
16 plot(y,p)
17 hold on
18
19
20 % Interpolando por spline
21
22 xx = [1.8; 2; 2.2; 2.4 ;2.6];
23 yy = [0.5815; 0.5767; 0.556; 0.5202; 0.4708];
24
25 f = spline(xx,yy,y);
26
27 plot(y,f)
28 plot(1.8,0.5815,'.','MarkerSize',20)
29 plot(2,0.5767,'.','MarkerSize',20)

```

```
30 plot(2.2,0.556, '.', 'MarkerSize',20)
31 plot(2.4,0.5202, '.', 'MarkerSize',20)
32 plot(2.6,0.4708, '.', 'MarkerSize',20)
33 legend({"Polinomio interpolador", "Spline cubico"}, "location", "
        southoutside", "fontsize", 15);
```

Listing B.11 – Código gerador do comparativo de interpolação via polinômio de quarto grau e via splines cúbicos para função de Bessel.