



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

DEPARTAMENTO DE MATEMÁTICA

Métodos de Interpolação Polinomial e Aplicações

Jefferson Matheus de Luna Souza

Orientadora: Dra. Maria Ângela Caldas Didier

RECIFE

2026



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

DEPARTAMENTO DE MATEMÁTICA

Jefferson Matheus de Luna Souza

Métodos de Interpolação Polinomial e Aplicações

Monografia de graduação apresentada ao Departamento de Matemática da Universidade Federal Rural de Pernambuco como componente optativo para obtenção de grau de licenciado.

Orientadora: Profa. Dra. Maria Ângela Caldas Didier

RECIFE

2026

Dados Internacionais de Catalogação na Publicação
Sistema Integrado de Bibliotecas da UFRPE
Bibliotecário(a): Auxiliadora Cunha – CRB-4 1134

S719m Souza, Jefferson Matheus de Luna.
Métodos de interpolação polinomial e aplicações /
Jefferson Matheus de Luna Souza. – Recife, 2026.
117 f.; il.

Orientador(a): Maria Ângela Caldas Didier.

Trabalho de Conclusão de Curso (Graduação) –
Universidade Federal Rural de Pernambuco, Licenciatura
em Matemática, Recife, BR-PE, 2026.

Inclui referências e apêndice(s).

1. Interpolação Polinomial. 2. Análise Numérica. 3.
Ajuste de curvas. 4. Métodos de Lagrange e Newton 5.
Splines Cúbicas. I. Didier, Maria Ângela Caldas, orient. II.
Título

CDD 510

Jefferson Matheus de Luna Souza

Métodos de Interpolação Polinomial e Aplicações

Monografia de graduação apresentada ao Departamento de Matemática da Universidade Federal Rural de Pernambuco como componente optativo para obtenção de grau de licenciado.

Trabalho aprovado. Recife, 12 de fevereiro de 2026:

Maria Ângela Caldas Didier
DM/UFRPE
Orientadora

Professor
Edgar Correa de Amorim Filho -
DM/UFRPE

Professor
Rodrigo Genuino Clemente - DM/UFRPE

Recife
2026

Dedico este trabalho à minha querida mãe, Ana Paula, que sempre esteve ao meu lado com amor, carinho e apoio, me incentivando a seguir em frente e a nunca desistir, e que desde cedo me ensinou que o estudo é um caminho capaz de transformar sonhos em conquistas.

Agradecimentos

Agradeço, primeiramente, a mim mesmo. Por todas as vezes em que o café foi meu único combustível, pela paciência de monge ao lidar com códigos que não compilavam e por não ter sucumbido ao impulso de jogar o computador pela janela ao ver um erro na matriz de Vandermonde. Essa conquista é o troféu da minha teimosia e da minha capacidade incrível de transformar litros de café e surtos leves em um diploma de Matemática.

Em seguida, meu agradecimento mais profundo vai para a minha mãe, Ana Paula. Obrigado por estar ao meu lado em cada conquista, cuidando de mim com um amor que nem a álgebra mais complexa consegue mensurar. Seu incentivo para que eu entrasse na universidade foi o empurrão que me trouxe até aqui, e sua dedicação incondicional é a base real sobre a qual todos os meus polinômios se sustentam.

Ao meu querido amigo e companheiro, Ivyson Carlos, minha profunda gratidão por todo o apoio nas fases mais recentes da minha vida. Sua presença constante, paciência e palavras de incentivo nunca me deixaram desistir, ajudando-me a enfrentar as dificuldades com mais leveza e a enxergar novos caminhos.

Aos amigos que a universidade me presenteou, em especial à Evellyn Basílio, agradeço por tornar a caminhada acadêmica mais tranquila e acolhedora. Os cafés da tarde, as longas conversas e os momentos de estudo em minha casa transformaram a rotina em algo mais agradável. Evellyn se tornou uma amiga para a vida, alguém com quem sei que posso contar tanto nos momentos felizes quanto nos mais difíceis.

Aos meus amigos Júnior Santos, Pedro Charles e Inês Eduarda, pelo incentivo constante durante toda esta jornada acadêmica. Sou profundamente grato pela presença e pelo apoio fundamental, especialmente nos desafios da reta final e na elaboração deste trabalho.

À minha orientadora, Maria Ângela, deixo minha profunda gratidão. Presente desde os primeiros passos da graduação, foi quem, ao ingressar no meu primeiro projeto de iniciação científica, trouxe ideias que deram origem a este trabalho. Agradeço por acreditar em mim, pela paciência em cada explicação, pelo cuidado ao orientar e por ser uma inspiração de dedicação e amor pelo ensino e pela pesquisa.

Ao PET Conexões de Saberes, ao Ciranda da Ciência e aos amigos que tive o privilégio de conhecer nesse programa — Jhonatan Ramos, Maria Vitória e Hillary Agatha —, agradeço por tornarem a vida universitária mais prazerosa e significativa. Agradeço também, de forma especial, à tutora Lorena Brizza por todo apoio, principalmente nos momentos mais desafiadores dessa caminhada.

Por fim, agradeço ao Fundo Nacional de Desenvolvimento da Educação (FNDE) pelo apoio financeiro, à Universidade Federal Rural de Pernambuco e ao Departamento de Matemática por me proporcionarem uma formação sólida e enriquecedora. Estudar Matemática ao lado de professores comprometidos e inspiradores foi uma das maiores oportunidades da minha vida, e levo comigo tudo o que aprendi com cada um deles.

Resumo

Este trabalho apresenta um estudo sobre a interpolação polinomial no contexto da Análise Numérica e da Teoria da Aproximação, abordando desde os métodos clássicos até técnicas mais robustas baseadas em interpolação por partes. O objetivo principal é investigar a validade, a eficiência e as limitações dessas técnicas na aproximação de funções e na representação de conjuntos de dados, tanto sob o ponto de vista teórico quanto computacional. São estudados os métodos de Lagrange e de Newton, destacando-se suas propriedades, vantagens operacionais e restrições. Além disso, discute-se o erro de interpolação de forma geral, enfatizando seus aspectos teóricos, estimativas e limitantes superiores, sem a atribuição específica a cada método. A aplicabilidade das técnicas é ilustrada por meio de problemas práticos, como a calibração de termômetros e a modelagem da média móvel de casos de COVID-19, evidenciando o papel da interpolação na análise de dados reais. As implementações computacionais dos principais algoritmos foram realizadas utilizando a linguagem de programação *Python*, possibilitando a geração de tabelas, gráficos e a validação numérica dos resultados obtidos. Por fim, discutem-se limitações inerentes ao uso de polinômios de alto grau, com destaque para o Fenômeno de Runge, e apresenta-se a interpolação por *splines* cúbicas como uma alternativa mais estável e precisa, aplicada à reconstrução de imagens digitais. Os resultados reforçam a relevância da interpolação polinomial como ferramenta fundamental na modelagem matemática, ressaltando a importância da escolha adequada do método de acordo com a natureza dos dados e os objetivos da aproximação.

Palavras-chaves: Interpolação Polinomial; Análise Numérica; Ajuste de curvas; Métodos de Lagrange e Newton; *Splines* Cúbicas.

Abstract

This work presents a study on polynomial interpolation within the context of Numerical Analysis and Approximation Theory, ranging from classical methods to more robust techniques based on piecewise interpolation. The main objective is to investigate the validity, efficiency, and limitations of these techniques in function approximation and dataset representation, from both a theoretical and computational viewpoint. The Lagrange and Newton methods are studied, highlighting their properties, operational advantages, and restrictions. Furthermore, the interpolation error is discussed in a general manner, emphasizing its theoretical aspects, estimates, and upper bounds, without specific attribution to each method. The applicability of these techniques is illustrated through practical problems, such as thermometer calibration and modeling the moving average of COVID-19 cases, demonstrating the role of interpolation in real data analysis. The computational implementations of the main algorithms were performed using the Python programming language, enabling the generation of tables, graphs, and the numerical validation of the obtained results. Finally, inherent limitations in the use of high-degree polynomials are discussed, with emphasis on the Runge Phenomenon, and cubic splines interpolation is presented as a more stable and accurate alternative, applied to digital image reconstruction. The results reinforce the relevance of polynomial interpolation as a fundamental tool in mathematical modeling, underscoring the importance of choosing the appropriate method according to the nature of the data and the approximation objectives.

Keywords: Polynomial Interpolation; Numerical Analysis; Curve Fitting; Lagrange and Newton Methods; Cubic Splines.

Lista de ilustrações

Figura 1 – Interpolação do conjunto de pontos $\{(x_i, f(x_i))\}_{i=0}^8$ pela função $g(x)$. . .	33
Figura 2 – Exemplos de polinômios interpoladores: (a) linear (grau 1); (b) quadrático (grau 2) e (c) cúbico (grau 3).	34
Figura 3 – Gráfico do polinômio interpolador $p_2(x) = -x^2 + 3x + 1$	36
Figura 4 – Comportamento dos polinômios base de Lagrange em nós equidistantes.	43
Figura 5 – Polinômios de base de Lagrange ($L_k(x)$) para os nós $[0, 10, 20, 30, 40]$	46
Figura 6 – Curva de calibração do termômetro por Interpolação de Lagrange.	47
Figura 7 – Polinômio interpolador de Newton para a média móvel dos casos de COVID-19 em 2021.	57
Figura 8 – Polinômio interpolador de Newton para a média móvel dos casos de COVID-19 em 2023.	58
Figura 9 – Polinômio interpolador de Newton para a média móvel dos casos de COVID-19 em 2025.	59
Figura 10 – Gráfico comparativo dos polinômios interpoladores de Newton para casos de COVID-19, nos anos de 2021, 2023 e 2025.	59
Figura 11 – Interpolação das funções $f_1(x)$ e $f_2(x)$ pelo polinômio linear $p_1(x)$	62
Figura 12 – Gráficos da função $f(x)$ (em azul) e do seu polinômio interpolador de grau 10 (em vermelho) — Fenômeno de Runge.	77
Figura 13 – Gráficos da função $f(x)$ (em azul) e do seu polinômio interpolador de grau 5 (em verde).	78
Figura 14 – Interpolação linear por partes.	79
Figura 15 – Função $g(x)$ diferenciável com derivada descontínua em $x = 0$	81
Figura 16 – <i>Spline</i> cúbico imitando o comportamento de uma régua delgada.	84
Figura 17 – Pontos utilizados na interpolação superpostos ao contorno da folha.	92
Figura 18 – Contorno da parte superior da folha por <i>spline</i> cúbico.	93

Lista de tabelas

Tabela 1 – Conjunto de pontos utilizados na interpolação de f .	34
Tabela 2 – Dados de interpolação para obtenção de $p_3(x)$.	38
Tabela 3 – Dados de interpolação para obter $p_2(x)$.	43
Tabela 4 – Comparação entre a temperatura medida (x) e a temperatura padrão ($f(x)$) ambas em °C.	45
Tabela 5 – Estrutura do cálculo das diferenças divididas.	51
Tabela 6 – Dados de interpolação para obter $p_2(x)$ pelo método de Newton.	53
Tabela 7 – Tabela de diferenças divididas para $f(x)$.	54
Tabela 8 – Média Móvel Semanal do número de casos de COVID-19 por mês e ano (2ª Semana).	55
Tabela 9 – Tabela de diferenças divididas da média móvel de casos de COVID-19 — Ano de 2021.	56
Tabela 10 – Tabela de diferenças divididas da média móvel de casos de COVID-19 — Ano de 2023.	57
Tabela 11 – Tabela de diferenças divididas da média móvel de casos de COVID-19 — Ano de 2025.	58
Tabela 12 – Valores da função $\ln(x)$.	66
Tabela 13 – Valores da função $f(x) = e^x + x - 1$ em nós igualmente espaçados.	71
Tabela 14 – Valores da função $f(x)$ utilizados na interpolação polinomial e na estimativa do erro.	73
Tabela 15 – Tabela de diferenças divididas para $p_2(x)$.	73
Tabela 16 – Valores tabelados de $f(x)$ no intervalo $[-1, 1]$.	76
Tabela 17 – Tabela de diferenças divididas para $f(x)$ no intervalo $[-1, 1]$.	77
Tabela 18 – Coordenadas dos pontos selecionados no <i>GeoGebra</i> para a reconstrução do contorno da folha por <i>splines</i> cúbicas.	92
Tabela 19 – Coeficientes do <i>spline</i> cúbico interpoladora $S_i(x)$.	93

Lista de abreviaturas e siglas

IC	Iniciação Científica
PET	Programa de Educação Tutorial
TCC	Trabalho de Conclusão de Curso
$O(n^2)$	Tempo de execução do algoritmo proporcional ao quadrado do tamanho da entrada n
$O(n^3)$	Tempo de execução do algoritmo proporcional ao cubo do tamanho da entrada n
MMS_t	Média Móvel Simples no tempo t
MMS	Média Móvel Semanal
SIR	Modelo Epidemiológico Compartmental (Susceptível-Infetado-Recuperado)
VIM	Vocabulário Internacional de Metrologia

Lista de símbolos

n	Número natural
\mathbb{R}	Conjunto dos números reais
\mathcal{C}^∞	Conjunto das funções infinitamente diferenciáveis
\mathcal{C}^2	Conjunto das funções que possuem derivadas contínuas até a segunda ordem
$p_6^{2021}(x)$	Polinômio interpolador de grau 6 para a média móvel dos casos de COVID-19 referentes às segundas semanas dos meses de fevereiro a agosto de 2021
$p_6^{2023}(x)$	Polinômio interpolador de grau 6 para a média móvel dos casos de COVID-19 referentes às segundas semanas dos meses de fevereiro a agosto de 2023
$p_6^{2025}(x)$	Polinômio interpolador de grau 6 para a média móvel dos casos de COVID-19 referentes às segundas semanas dos meses de fevereiro a agosto de 2025
$\phi(t)$	Função auxiliar para o erro de interpolação
$E_n(x)$	Erro de interpolação
$f^{(n+1)}(x)$	Derivada de ordem $(n + 1)$ da função f
M_{n+1}	Limitante superior do módulo da derivada de ordem $(n + 1)$ da função f no intervalo I
$T_n(x)$	Polinômio de Taylor de ordem n da função f centrado no ponto x_0

Sumário

Introdução	23
1 FUNDAMENTAÇÃO TEÓRICA	26
1.1 Polinômios	26
1.2 Matriz de Vandermonde	27
1.3 Interpolação polinomial e o polinômio interpolador	32
1.4 Teorema de existência e unicidade	36
2 MÉTODOS DE INTERPOLAÇÃO POLINOMIAL	40
2.1 Método de Lagrange	40
2.1.1 Polinômio Interpolador de Lagrange	40
2.1.2 Um modelo de calibração de termômetro	44
2.2 Método de Newton	48
2.2.1 Diferenças divididas	48
2.2.2 Polinômio Interpolador de Newton	51
2.2.3 Um modelo para a média móvel dos casos de COVID-19	54
3 ANÁLISE DO ERRO DE INTERPOLAÇÃO	61
3.1 Erro de interpolação	61
3.2 Limitante superior para o erro de interpolação	68
3.3 Estimativa do erro por meio de diferenças divididas	72
4 INTERPOLAÇÃO POR SPLINES	75
4.1 O fenômeno de Runge	76
4.2 Interpolação por <i>spline</i> linear	79
4.3 Interpolação por <i>spline</i> cúbico	82
4.4 Reconstrução de imagens digitais	91
5 CONCLUSÃO	95
REFERÊNCIAS	97
Apêndice A — Resultados Complementares	99
A.1 Resultados sobre Aproximações de Funções Reais	99
A.2 Resultados de Álgebra Linear	99
A.3 Simetria das Diferenças Divididas	103
A.4 Teorema de Gerschgorin	105

A.5 Teorema de Lévy-Desplanques	106
Apêndice B — Algoritmos Implementados	108
B.1 Polinômio interpolador por sistema linear (Exemplo 1.3.2)	108
B.2 Calibração de termômetro pelo método de Lagrange	110
B.3 Média móvel dos casos de COVID-19 pelo método de Newton.	112
B.4 Reconstrução digital do contorno de uma folha por <i>spline</i> cúbico.	115

Introdução

A Interpolação Polinomial foi selecionada como tema central deste trabalho, dando continuidade a um estudo aprofundado iniciado durante a Iniciação Científica (IC), no âmbito do Programa de Educação Tutorial (PET), a partir de uma proposta inicial da orientadora. A transição e expansão para o Trabalho de Conclusão de Curso (TCC) foram motivadas pelo interesse no ajuste de curvas e pelo vasto potencial da modelagem matemática na exploração de fenômenos. Este trabalho visa consolidar o conhecimento adquirido e promover o crescimento acadêmico, explorando a Análise Numérica e a Teoria da Aproximação, campos que frequentemente não são abordados com a devida profundidade na grade curricular da Licenciatura.

A Análise Numérica, um dos pilares da Matemática Aplicada, tem como um de seus alicerces a Teoria da Aproximação. Seu papel é substituir funções complexas por outras mais simples, facilitando o tratamento e a resolução computacional. Nesse campo, destacam-se dois problemas fundamentais: o primeiro busca construir uma função que se aproxime da original de modo a minimizar o erro em um intervalo; o segundo exige que a aproximação coincida exatamente com a função em determinados pontos. É nesse cenário que a Interpolação Polinomial assume um papel crucial. Esta técnica consiste em determinar um único polinômio que passe exatamente por um conjunto finito de pontos de dados conhecidos. Essa abordagem oferece uma forma prática e eficiente de representar funções cuja expressão analítica é complexa, difícil de obter ou mesmo desconhecida. A partir dela, torna-se possível modelar fenômenos complexos, analisar comportamentos e estimar valores intermediários em diversas áreas da ciência, como engenharia, física, economia e computação.

A escolha do Método Interpolador, contudo, é crítica, pois distintas abordagens apresentam vantagens e desvantagens em termos de estabilidade, custo computacional e flexibilidade.

O método mais direto é a Resolução de Sistemas Lineares, que determina os coeficientes do polinômio interpolador ao solucionar o sistema de equações gerado pelos pontos de dados. Embora prove a unicidade do polinômio, sua implementação é prejudicada pelo alto custo computacional, tipicamente $O(n^3)$ e, principalmente, pelo mau condicio-

namento da Matriz de Vandermonde para polinômios de grau elevado, o que induz a erros significativos.

Em contraste, o método de Lagrange simplifica a formulação ao construir polinômios de base que fornecem uma fórmula analítica elegante, evitando o sistema linear. Sua desvantagem, no entanto, é a inflexibilidade: qualquer adição de um novo ponto de dados exige o recálculo completo de todos os termos, tornando-o ineficiente para processos incrementais.

O método de Newton (Diferenças Divididas) supera esta limitação ao permitir a construção incremental do polinômio, onde a adição de novos pontos exige apenas o cálculo de um novo termo, otimizando o tempo de processamento. A complexidade de cálculo dos coeficientes é reduzida para $O(n^2)$.

Apesar das diferenças em eficiência, os métodos baseados em um único polinômio de alto grau (Sistema Linear, Lagrange e Newton) compartilham uma desvantagem crucial quando o número de pontos aumenta: a instabilidade nas extremidades do intervalo. Esta instabilidade é conhecida como o Fenômeno de Runge, que manifesta grandes oscilações e impede a obtenção de aproximações precisas em todo o domínio.

Para solucionar o problema da instabilidade inerente ao alto grau polinomial, a Interpolação por *Splines* constitui o método mais adotado na prática. Em vez de utilizar um único polinômio sobre todo o intervalo, as *splines* dividem o domínio em segmentos menores, aplicando polinômios de grau baixo (geralmente cúbico) em cada subintervalo. A principal vantagem das *splines* é a estabilidade, que elimina o Fenômeno de Runge e garante a suavidade da curva (mantendo a continuidade das derivadas nos nós). Embora exija a resolução de um sistema linear para determinar os coeficientes de suavidade, este sistema é tridiagonal e facilmente solucionável em $O(n)$, tornando a *spline* o método mais robusto e preferido para modelagem e aplicações em engenharia e computação gráfica.

Este trabalho pretende apresentar as técnicas de interpolação polinomial, abordando seus fundamentos teóricos, métodos clássicos e aplicações computacionais. A análise será desenvolvida por meio de diferentes abordagens, selecionadas tanto pela relevância conceitual quanto pela utilidade prática, e distribuídas ao longo de quatro capítulos principais.

No Capítulo 1, abordamos os conceitos fundamentais da interpolação, estabelecendo o arcabouço teórico necessário por meio de resultados essenciais, como a matriz de Vandermonde e o Teorema de Existência e Unicidade do polinômio interpolador.

No Capítulo 2, estudamos os métodos clássicos: inicia-se pelo Polinômio Interpolador de Lagrange, aplicado ao problema de calibração de termômetros, e segue-se com o Método de Newton por Diferenças Divididas, escolhido por sua eficiência computacional na incorporação de novos pontos e aplicado à modelagem da média móvel de casos de

COVID-19 com dados do Ministério da Saúde.

No Capítulo 3, realizamos a análise do Erro de Interpolação, discutindo a precisão, os limitantes superiores e a estimativa via diferenças divididas.

No Capítulo 4, tratamos a Interpolação por *Splines*, com foco na Interpolação por *Spline* Linear e *Spline* Cúbico. Para ilustrar seu potencial, apresenta-se uma aplicação prática utilizando *spline* cúbico na reconstrução de imagens digitais.

No Capítulo Final, apresentamos as conclusões gerais e as perspectivas futuras da pesquisa.

Observamos que os materiais complementares estão reunidos em dois apêndices: o Apêndice A contém o suporte teórico adicional, enquanto o Apêndice B disponibiliza os códigos em *python* utilizados nas principais implementações e aplicações numéricas do trabalho.

1 Fundamentação Teórica

Neste capítulo, estabelecemos a fundamentação teórica necessária para o desenvolvimento do trabalho, com foco nas ferramentas essenciais de Análise Numérica. Serão apresentados os conceitos de polinômios, a Matriz de Vandermonde e o Polinômio Interpolador, bem como o Teorema da Existência e Unicidade que o define.

1.1 Polinômios

Segundo Gonçalves [1], o estudo do conceito de polinômio, fundamental na Matemática, tem suas raízes nas civilizações antigas. Os Babilônios, por volta de 1800 a.C., já resolviam o equivalente às equações quadráticas (grau 2), focando na busca pelas raízes das equações algébricas. Contudo, a formalização do conceito levou milênios. Um avanço importante ocorreu no Renascimento Italiano do século XVI, com a solução das equações de grau 3 (cúbicas) e grau 4 (quárticas). Figuras como Cardano e Tartaglia disputaram a descoberta da fórmula para a cúbica, enquanto Ferrari solucionou a quártica. Essas soluções demonstraram, de forma surpreendente, a necessidade de introduzir os números complexos, mesmo que inicialmente de maneira incipiente, e estabeleceram que as raízes poderiam ser expressas usando radicais até o grau 4. Mais tarde, a notação algébrica moderna, impulsionada por Viète e Descartes, permitiu a transição do problema retórico para o objeto matemático formal que hoje conhecemos.

Definição 1.1.1 (Polinômio real de grau n). Um polinômio de grau n sobre \mathbb{R} em uma indeterminada x é definido pela expressão:

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0,$$

onde $a_i \in \mathbb{R}$, para $i = 0, \dots, n$, com $a_n \neq 0$.

O conceito do polinômio atingiu o seu ápice formal no final do século XVIII, quando Carl Friedrich Gauss, em 1799, apresentou a primeira prova rigorosa do Teorema Fundamental da Álgebra (TFA) em sua tese de doutoramento na Universidade de Helmstadt. Este teorema estabelece a propriedade de completude dos polinômios sobre o campo dos números complexos: todo polinômio de grau $n \geq 1$ possui exatamente n raízes complexas

(contando suas multiplicidades). O trabalho de Gauss transformou o Teorema Fundamental da Álgebra em um pilar da Matemática Moderna, definindo o domínio \mathbb{C} como o ambiente “natural” para a solução completa das equações polinomiais.

Na Análise Real (Cálculo), os polinômios são valorizados por sua suavidade e previsibilidade: são contínuos e infinitamente diferenciáveis (\mathcal{C}^∞) em todo o seu domínio (\mathbb{R}). Isso simplifica a aplicação de operações, como o cálculo de limites, derivadas e integrais. Além disso, o seu comportamento assintótico, isto é, o seu crescimento quando $x \rightarrow \pm\infty$, é sempre dominado pelo seu termo de maior grau ($a_n x^n$).

Globalmente, o Teorema de Aproximação de Weierstrass afirma que qualquer função contínua em um intervalo fechado pode ser uniformemente aproximada por um polinômio com um grau de precisão arbitrário. Este resultado estabelece que o conjunto dos polinômios é denso no espaço das funções contínuas. Localmente, os polinômios de Taylor fornecem a melhor aproximação possível para uma função diferenciável em torno de um ponto específico, sendo essenciais para a análise do erro e a modelagem funcional [2, p. 233].

Consequentemente, as funções polinomiais se estabelecem como peças essenciais e fundamentais na Teoria da Aproximação e na Análise Numérica, sendo frequentemente empregadas para modelar e aproximar funções mais complexas em diversas áreas da ciência e da engenharia.

1.2 Matriz de Vandermonde

Nesta seção, introduziremos a Matriz de Vandermonde, que é a ferramenta algébrica principal para a demonstração do Teorema de Existência e Unicidade do Polinômio Interpolador, um resultado que será importante saber ao iniciarmos nosso estudo de Interpolação.

Definição 1.2.1. Seja $A = \{\alpha_i\}_{i=0}^n$ um conjunto ordenado de $n + 1$ escalares (reais ou complexos), denominados elementos característicos. A matriz de Vandermonde de ordem $(n + 1) \times (n + 1)$, denotada por $V(A)$, é a matriz quadrada cujos elementos v_{ij} (na i -ésima linha e j -ésima coluna) são definidos por:

$$v_{ij} = \alpha_{i-1}^{j-1}, \quad \text{para } 1 \leq i \leq n + 1 \quad \text{e} \quad 1 \leq j \leq n + 1.$$

Na notação matricial padrão, onde as colunas correspondem aos elementos caracterís-

tivos elevados a diferentes potências, a matriz é representada como:

$$V(A) = \begin{pmatrix} 1 & \alpha_0 & \alpha_0^2 & \cdots & \alpha_0^n \\ 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^n \\ 1 & \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_n & \alpha_n^2 & \cdots & \alpha_n^n \end{pmatrix}.$$

Observa-se que os elementos de cada linha i da matriz $V(A)$ formam uma progressão geométrica com primeiro termo igual a 1 e razão α_{i-1} . Alternativamente, algumas referências adotam a matriz transposta, V^T , onde as colunas (em vez das linhas da forma original) estão em progressão geométrica:

$$V^T = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ \alpha_0 & \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \alpha_0^2 & \alpha_1^2 & \alpha_2^2 & \cdots & \alpha_n^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_0^n & \alpha_1^n & \alpha_2^n & \cdots & \alpha_n^n \end{pmatrix}.$$

Observe que o valor do determinante é invariante sob a transposição, uma vez que $\det(V(A)) = \det(V(A)^T)$.

Para que a interpolação polinomial seja bem definida, é essencial que a matriz associada ao sistema linear seja não singular. Isso é garantido pela Proposição do Determinante de Vandermonde:

Proposição 1.2.1. Seja V^T uma matriz de Vandermonde de ordem $(n+1) \times (n+1)$. Se $\alpha_0, \alpha_1, \dots, \alpha_n$ são distintos, então:

$$\det(V) = \det(V^T) = \prod_{0 \leq i < j \leq n} (\alpha_j - \alpha_i).$$

Esta expressão é conhecida como o produto de diferenças de Vandermonde. Ela representa o produto de todas as diferenças $\alpha_j - \alpha_i$ onde o índice do segundo fator, j , é estritamente maior que o do primeiro, i .

Demonstração. Demonstraremos esta fórmula pelo método de Indução Finita sobre a ordem $m = n + 1$ da matriz. Para o caso base $m = 2$ (ou $n = 1$), a verificação é direta via Regra de Sarrus:

$$\det(V^T) = \begin{vmatrix} 1 & 1 \\ \alpha_0 & \alpha_1 \end{vmatrix} = \alpha_1 - \alpha_0 = \prod_{0 \leq i < j \leq 1} (\alpha_j - \alpha_i).$$

Para o caso $m = 3$ (ou $n=2$), isto é, uma matriz de ordem 3×3 :

$$\det(V^T) = \begin{vmatrix} 1 & 1 & 1 \\ \alpha_0 & \alpha_1 & \alpha_2 \\ \alpha_0^2 & \alpha_1^2 & \alpha_2^2 \end{vmatrix},$$

o determinante pode ser calculado pela Regra de Sarrus ou por expansão em cofatores.

Utilizando a Regra de Sarrus, temos que:

$$\det(V^T) = \alpha_1\alpha_2^2 + \alpha_2\alpha_0^2 + \alpha_0\alpha_1^2 - \alpha_1\alpha_0^2 - \alpha_2\alpha_1^2 - \alpha_0\alpha_2^2.$$

Reorganizando os termos em função de α_2 e fatorando, obtemos:

$$\begin{aligned} \det(V^T) &= \alpha_2^2(\alpha_1 - \alpha_0) - \alpha_2(\alpha_1^2 - \alpha_0^2) + \alpha_1\alpha_0(\alpha_1 - \alpha_0) \\ &= \alpha_2^2(\alpha_1 - \alpha_0) - \alpha_2(\alpha_1 - \alpha_0)(\alpha_1 + \alpha_0) + \alpha_1\alpha_0(\alpha_1 - \alpha_0). \end{aligned}$$

Seguindo que:

$$\det(V^T) = (\alpha_1 - \alpha_0)[\alpha_2^2 - \alpha_2(\alpha_1 + \alpha_0) + \alpha_1\alpha_0].$$

Como o termo entre colchetes é um trinômio em α_2 , podemos fatorá-lo como:

$$\alpha_2^2 - \alpha_1\alpha_2 - \alpha_0\alpha_2 + \alpha_1\alpha_0 = \alpha_2(\alpha_2 - \alpha_1) - \alpha_0(\alpha_2 - \alpha_1) = (\alpha_2 - \alpha_1)(\alpha_2 - \alpha_0)$$

Dessa forma, ficamos com:

$$\det(V^T) = (\alpha_1 - \alpha_0)(\alpha_2 - \alpha_1)(\alpha_2 - \alpha_0).$$

Logo, os casos base $m = 2$ e $m = 3$ confirmam a fórmula do produto.

Agora, iremos provar para uma matriz de ordem $(n + 1) \times (n + 1)$, supondo que a fórmula do determinante de Vandermonde seja válida para as matrizes de ordem $n \times n$.

Assumimos que, para um conjunto de n escalares $\{\alpha_0, \alpha_1, \dots, \alpha_{n-1}\}$, o determinante da Matriz de Vandermonde de ordem $n \times n$ é dado por:

$$\det(V_{n \times n}^T) = \prod_{0 \leq i < j \leq n-1} (\alpha_j - \alpha_i).$$

Considere D_n o determinante da matriz de Vandermonde de ordem $(n+1) \times (n+1)$, com elementos característicos $\alpha_0, \alpha_1, \dots, \alpha_n$ tais que:

$$D_n = \begin{vmatrix} 1 & 1 & \cdots & 1 & 1 \\ \alpha_0 & \alpha_1 & \cdots & \alpha_{n-1} & \alpha_n \\ \alpha_0^2 & \alpha_1^2 & \cdots & \alpha_{n-1}^2 & \alpha_n^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_0^n & \alpha_1^n & \cdots & \alpha_{n-1}^n & \alpha_n^n \end{vmatrix}.$$

Nosso objetivo é demonstrar que:

$$D_n = \left(\prod_{i=0}^{n-1} (\alpha_n - \alpha_i) \right) \cdot D_{n-1},$$

onde D_{n-1} é o determinante de Vandermonde de ordem $n \times n$ formado pelos elementos $\{\alpha_0, \alpha_1, \dots, \alpha_{n-1}\}$.

Utilizaremos a propriedade de que a subtração de um múltiplo de uma linha de outra não altera o valor do determinante.

O objetivo é transformar o problema de ordem $n+1$ em um problema de ordem $n \times n$, o qual já pode ser resolvido pela hipótese de indução. Para isso, aplicaremos as seguintes operações elementares de linha, começando da última linha (onde k é o expoente):

$$L_{k+1} \leftarrow L_{k+1} - \alpha_n L_k \quad \text{para} \quad k = n, n-1, \dots, 1.$$

Dessa forma, o novo elemento $v'_{i,j}$ na linha $i = k+1$ é determinado pela diferença:

$$v'_{k+1,j} = \alpha_{j-1}^k - \alpha_n \alpha_{j-1}^{k-1} = \alpha_{j-1}^{k-1} (\alpha_{j-1} - \alpha_n).$$

A aplicação sequencial dessas operações em todas as linhas transforma o determinante D_n em um novo determinante, $\det(D')$:

$$\begin{vmatrix} 1 & 1 & \cdots & 1 & 1 \\ \alpha_0 & \alpha_1 & \cdots & \alpha_{n-1} & \alpha_n \\ \alpha_0^2 & \alpha_1^2 & \cdots & \alpha_{n-1}^2 & \alpha_n^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_0^n & \alpha_1^n & \cdots & \alpha_{n-1}^n & \alpha_n^n \end{vmatrix} \xrightarrow{L_{k+1} \leftarrow L_{k+1} - \alpha_n L_k} \det(D')$$

O resultado final do determinante D' é:

$$D' = \begin{vmatrix} 1 & 1 & \cdots & 1 & 1 \\ \alpha_0 - \alpha_n & \alpha_1 - \alpha_n & \cdots & \alpha_{n-1} - \alpha_n & 0 \\ \alpha_0(\alpha_0 - \alpha_n) & \alpha_1(\alpha_1 - \alpha_n) & \cdots & \alpha_{n-1}(\alpha_{n-1} - \alpha_n) & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_0^{n-1}(\alpha_0 - \alpha_n) & \alpha_1^{n-1}(\alpha_1 - \alpha_n) & \cdots & \alpha_{n-1}^{n-1}(\alpha_{n-1} - \alpha_n) & 0 \end{vmatrix}.$$

Uma vez que as operações de linha $L_{k+1} \leftarrow L_{k+1} - \alpha_n L_k$ não alteram o valor do determinante [3, p. 69], temos $D_n = \det(D')$.

Expandindo o determinante $\det(D')$ ao longo da última coluna (C_{n+1}), apenas o primeiro cofator será não nulo ($v'_{1,n+1} = 1$). O sinal do cofator é dado por $(-1)^{1+(n+1)} = (-1)^{n+2} = (-1)^n$. Assim:

$$D_n = (-1)^n \cdot 1 \cdot \det(M_{1,n+1}), \quad (1.1)$$

onde $M_{1,n+1}$ é a matriz obtida removendo a primeira linha e a última coluna de D' , isto é:

$$M_{1,n+1} = \begin{vmatrix} \alpha_0 - \alpha_n & \alpha_1 - \alpha_n & \cdots & \alpha_{n-1} - \alpha_n \\ \alpha_0(\alpha_0 - \alpha_n) & \alpha_1(\alpha_1 - \alpha_n) & \cdots & \alpha_{n-1}(\alpha_{n-1} - \alpha_n) \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_0^{n-1}(\alpha_0 - \alpha_n) & \alpha_1^{n-1}(\alpha_1 - \alpha_n) & \cdots & \alpha_{n-1}^{n-1}(\alpha_{n-1} - \alpha_n) \end{vmatrix}_{n \times n}.$$

Este produto final engloba o produto de todas as diferenças $(\alpha_j - \alpha_i)$ onde $j > i$ para todos os índices de 0 até n e, portanto, podemos reescrevê-lo como:

$$\det(M_{1,n+1}) = \prod_{j=0}^{n-1} (\alpha_j - \alpha_n) \cdot \det(V'_{n \times n}). \quad (1.2)$$

A matriz $V'_{n \times n}$ representa a matriz de Vandermonde obtida após o processo de fatoração, sendo dada por:

$$V'_{n \times n} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha_0 & \alpha_1 & \cdots & \alpha_{n-1} \\ \alpha_0^2 & \alpha_1^2 & \cdots & \alpha_{n-1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_0^{n-1} & \alpha_1^{n-1} & \cdots & \alpha_{n-1}^{n-1} \end{pmatrix}.$$

Pela hipótese de indução, o determinante de $V'_{n \times n}$ é dado por:

$$\det(V'_{n \times n}) = \prod_{0 \leq i < j \leq n-1} (\alpha_j - \alpha_i). \quad (1.3)$$

Substituindo (1.3) em (1.2), obtemos:

$$\det(M_{1,n+1}) = \left[\prod_{j=0}^{n-1} (\alpha_j - \alpha_n) \right] \cdot \left[\prod_{0 \leq i < j \leq n-1} (\alpha_j - \alpha_i) \right]. \quad (1.4)$$

Por fim, substituindo o resultado em (1.1), concluímos que:

$$D_n = (-1)^n \cdot \left[\prod_{j=0}^{n-1} (\alpha_j - \alpha_n) \right] \cdot \left[\prod_{0 \leq i < j \leq n-1} (\alpha_j - \alpha_i) \right]. \quad (1.5)$$

Para reescrever o resultado no padrão do produto de Vandermonde $(\alpha_j - \alpha_i)$, com $j > i$, reorganizamos o primeiro produto:

$$\prod_{j=0}^{n-1} (\alpha_j - \alpha_n) = \prod_{i=0}^{n-1} -(\alpha_n - \alpha_i) = (-1)^n \prod_{i=0}^{n-1} (\alpha_n - \alpha_i). \quad (1.6)$$

Substituindo o resultado da equação (1.6) em (1.5):

$$D_n = (-1)^n \cdot \left[(-1)^n \prod_{i=0}^{n-1} (\alpha_n - \alpha_i) \right] \cdot \left[\prod_{0 \leq i < j \leq n-1} (\alpha_j - \alpha_i) \right].$$

Como $(-1)^n \cdot (-1)^n = (-1)^{2n} = 1$:

$$D_n = \left[\prod_{i=0}^{n-1} (\alpha_n - \alpha_i) \right] \cdot \left[\prod_{0 \leq i < j \leq n-1} (\alpha_j - \alpha_i) \right].$$

Este produto final engloba exatamente todas as diferenças $(\alpha_j - \alpha_i)$ onde $j > i$ para todos os índices de 0 até n e, portanto, podemos reescrevê-lo como:

$$D_n = \prod_{0 \leq i < j \leq n} (\alpha_j - \alpha_i).$$

□

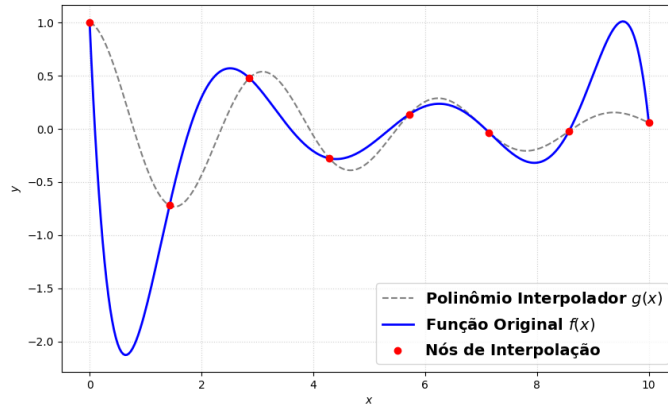
1.3 Interpolação polinomial e o polinômio interpolador

Segundo Atkinson [4, p. 131], interpolar consiste em determinar uma função g , pertencente a uma classe previamente especificada, cujo gráfico passe exatamente por um conjunto finito de pontos conhecidos. Em outras palavras, a interpolação permite representar ou aproximar uma função desconhecida (ou até mesmo conhecida, mas complexa) a partir de informações discretas conhecidas, assegurando que o modelo reproduza fielmente os valores observados. O processo consiste em substituir uma função f por outra função g , escolhida dentro de uma classe previamente definida, que coincida com f nos pontos conhecidos. Assim, g torna-se uma aproximação válida de f ao longo do intervalo considerado.

Conforme destacado por Ruggiero e Lopes [5, p. 212], a interpolação é fundamental quando se dispõe apenas de valores discretos de uma função — ou seja, um conjunto finito de dados isolados, provenientes de medições ou tabelas. Nesses casos, a técnica permite estimar com precisão o comportamento da função em pontos não tabelados, preenchendo as lacunas informacionais entre as amostras conhecidas. Além disso, em geral, a interpolação é empregada quando a expressão analítica de f apresenta uma forma complicada, tornando operações como a diferenciação e integração de difícil execução ou impossíveis de serem realizadas de maneira direta.

A seguir, ilustramos esse processo para o conjunto de pontos $\{(x_i, y_i)\}_{i=0}^7$. Na Figura 1, a função original $f(x)$ é representada juntamente com uma função interpoladora $g(x)$, a qual fornece uma aproximação analiticamente mais simples e operacionalmente mais conveniente.

Figura 1 – Interpolação do conjunto de pontos $\{(x_i, f(x_i))\}_{i=0}^8$ pela função $g(x)$.



Fonte: Autores.

Definição 1.3.1 (Função interpoladora). Sejam $n+1$ pontos distintos $\{x_i\}_{i=0}^n$, ordenados de modo que $x_i < x_{i+1}$ para todo i , denominados nós de interpolação, e considere o conjunto de pontos

$$\{(x_i, y_i) \in I \times \mathbb{R}\}_{i=0}^n,$$

onde $I = [x_0, x_n]$. Seja $\mathcal{F}_I = \{\pi : I \rightarrow \mathbb{R}\}$ uma família de funções definidas no intervalo I . Diz-se que uma função $g \in \mathcal{F}_I$ é uma função interpoladora dos pontos dados se

$$g(x_i) = y_i, \text{ para } i = 0, 1, 2, \dots, n.$$

Nesse caso, afirma-se que a função g interpola o conjunto de pontos $\{(x_i, y_i)\}_{i=0}^n$.

Na prática da interpolação, dado um conjunto de $n+1$ pontos de dados distintos, a classe de funções interpoladoras \mathcal{F}_I é, de forma convencional, escolhida como o espaço dos polinômios de grau no máximo n , denotado por \mathcal{P}_n . Esta escolha tem fundamentos sólidos. Ela se justifica pelas propriedades analíticas e computacionais superiores dos polinômios, que garantem a existência e unicidade da solução interpoladora e permitem representar funções com simplicidade. Além disso, os polinômios facilitam a execução eficiente de operações fundamentais do cálculo, como a diferenciação e a integração, conforme observamos na Seção 1.1.

Definição 1.3.2 (Polinômio Interpolador). Sejam $n+1$ nós distintos $\{x_i\}_{i=0}^n$, ordenados de modo que $x_i < x_{i+1}$ para todo i e f é uma função definida nestes pontos. Considere os valores correspondentes $y_i = f(x_i)$. O polinômio interpolador de grau no máximo n é

o polinômio $p_n(x)$ que satisfaz as condições de interpolação:

$$p_n(x_i) = y_i, \quad i = 0, 1, \dots, n.$$

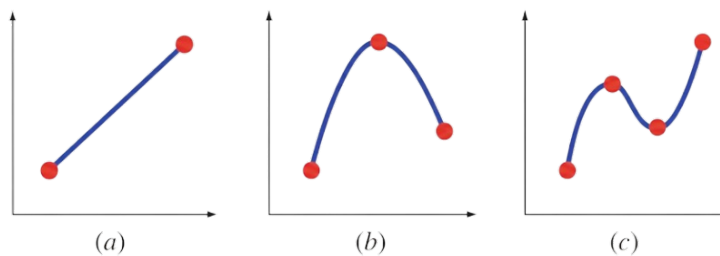
Esse polinômio pode ser expresso na forma canônica como:

$$p_n(x) = \sum_{k=0}^n a_k x^k,$$

em que os coeficientes $a_k \in \mathbb{R}$ são determinados de maneira única, pelos $n + 1$ pontos de interpolação [6].

Exemplo 1.3.1. São exemplos de polinômios interpoladores: (a) um polinômio de primeiro grau (linear), que passa por dois pontos; (b) um polinômio de segundo grau (quadrático), que interpola três pontos; e (c) um polinômio de terceiro grau (cúbico), que ajusta quatro pontos.

Figura 2 – Exemplos de polinômios interpoladores: (a) linear (grau 1); (b) quadrático (grau 2) e (c) cúbico (grau 3).



Fonte: Adaptado de [7].

No exemplo a seguir, utilizaremos o método da Matriz de Vandermonde para encontrar o Polinômio Interpolador de grau $n = 2$ (polinômio quadrático) que passa por um conjunto de três pontos.

Exemplo 1.3.2. Suponha que dispomos de três medições exatas de uma função desconhecida f . Esses valores observados estão organizados na Tabela 1:

Tabela 1 – Conjunto de pontos utilizados na interpolação de f .

x_i	0	1	3
$f(x_i)$	1	3	1

Fonte: Autores.

O objetivo é encontrar o polinômio interpolador $p_2(x)$, que possui a forma geral:

$$p_2(x) = a_2 x^2 + a_1 x + a_0.$$

Ao impor que o polinômio passe exatamente pelos pontos amostrais, obtemos o seguinte sistema de equações lineares:

$$\begin{cases} a_0 + a_1(0) + a_2(0)^2 = 1 & (\text{Ponto } p_0) \\ a_0 + a_1(1) + a_2(1)^2 = 3 & (\text{Ponto } p_1) \\ a_0 + a_1(3) + a_2(3)^2 = 1 & (\text{Ponto } p_2) \end{cases}$$

que, após simplificações, assume a forma:

$$\begin{aligned} a_0 &= 1 \\ a_0 + a_1 + a_2 &= 3 \quad . \\ a_0 + 3a_1 + 9a_2 &= 1 \end{aligned} \tag{1.7}$$

O Sistema (1.7) pode ser escrito na forma matricial $V\mathbf{a} = \mathbf{y}$, onde V é a Matriz de Vandermonde associada aos nós (x_0, x_1, x_2) :

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 3 & 9 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \\ 1 \end{pmatrix}.$$

A resolução do sistema procede da seguinte forma:

Passo A: Pela primeira equação, concluímos que $a_0 = 1$.

Passo B: Substituindo $a_0 = 1$ nas equações restantes, obtemos:

$$1 + a_1 + a_2 = 3 \implies a_1 + a_2 = 2 \quad \text{e} \tag{1.8}$$

$$1 + 3a_1 + 9a_2 = 1 \implies a_1 + 3a_2 = 0. \tag{1.9}$$

Passo C: Subtraindo a Equação (1.8) da Equação (1.9):

$$(a_1 + a_2) - (a_1 + 3a_2) = 2 - 0,$$

que resulta em

$$-2a_2 = 2 \implies a_2 = -1.$$

Passo D: Substituindo o valor de $a_2 = -1$ na Equação (1.8):

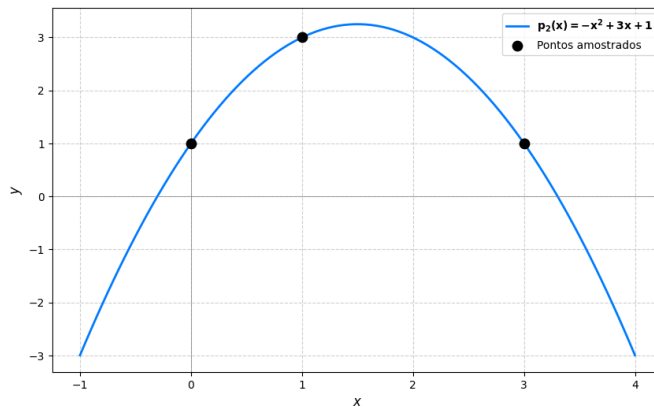
$$a_1 + (-1) = 2 \implies a_1 = 3.$$

Com os coeficientes determinados ($a_0 = 1$, $a_1 = 3$ e $a_2 = -1$), obtemos o polinômio interpolador desejado:

$$p_2(x) = -x^2 + 3x + 1. \quad (1.10)$$

A Figura 3 apresenta o gráfico do polinômio interpolador dado na Equação 1.10, o qual passa exatamente pelos pontos $(0, 1)$, $(1, 3)$ e $(3, 1)$. Trata-se, portanto, de uma interpolação quadrática que reproduz de forma precisa os valores observados nesses três pontos.

Figura 3 – Gráfico do polinômio interpolador $p_2(x) = -x^2 + 3x + 1$.



Fonte: Autores.

Convém destacar que, independentemente do método empregado para determinar o polinômio interpolador de uma função f em $n + 1$ pontos distintos, o resultado final é sempre o mesmo polinômio. Provaremos esta afirmação na próxima seção.

1.4 Teorema de existência e unicidade

Teorema 1.4.1. Seja dado um conjunto de $n + 1$ pontos $\{(x_i, y_i)\}_{i=0}^n$, com abscissas distintas, isto é, $x_i \neq x_j$ para $i \neq j$. Então, existe um único polinômio interpolador $p_n(x)$, de grau no máximo n , tal que

$$p_n(x_i) = y_i, \quad \text{para } i = 0, 1, 2, \dots, n.$$

Demonstração. Considere um polinômio $p_n(x)$ de grau no máximo n , com coeficientes reais a_i , $i = 0, 1, \dots, n$, dado por

$$p_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0.$$

A condição de que o polinômio p_n interpole o conjunto de pontos $\{(x_i, y_i)\}_{i=0}^n$ impõe que

$$p_n(x_i) = y_i, \quad \text{para } i = 0, 1, 2, \dots, n.$$

Isso conduz ao seguinte sistema de $n + 1$ equações lineares, cujas incógnitas são os coeficientes a_0, a_1, \dots, a_n :

$$\begin{aligned} p_n(x_0) &= a_n x_0^n + a_{n-1} x_0^{n-1} + \dots + a_1 x_0 + a_0 = y_0, \\ p_n(x_1) &= a_n x_1^n + a_{n-1} x_1^{n-1} + \dots + a_1 x_1 + a_0 = y_1, \\ p_n(x_2) &= a_n x_2^n + a_{n-1} x_2^{n-1} + \dots + a_1 x_2 + a_0 = y_2, \\ &\vdots \\ p_n(x_n) &= a_n x_n^n + a_{n-1} x_n^{n-1} + \dots + a_1 x_n + a_0 = y_n. \end{aligned} \tag{1.11}$$

O sistema (1.11) pode ser escrito em sua forma matricial $V \cdot \mathbf{a} = \mathbf{y}$:

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}.$$

A matriz V é denominada Matriz de Vandermonde.

Como as abscissas x_0, x_1, \dots, x_n são distintas, o determinante da matriz de Vandermonde é não nulo, isto é, $\det(V) \neq 0$, conforme demonstrado na Proposição 1.2.1. Pela regra de Cramer (Apêndice A.2.1), como o determinante é não nulo, o sistema é possível e determinado. Assim, os coeficientes a_i existem e são únicos, o que garante a existência e a unicidade do polinômio interpolador $p_n(x)$.

Embora a determinação do polinômio interpolador $p_n(x)$ via sistema linear seja conceitualmente direta, o procedimento carece de confiabilidade numérica na prática. Em muitos casos, a matriz de Vandermonde associada é mal condicionada, o que significa que pequenas imprecisões no arredondamento das casas decimais ou perturbações nos dados de entrada são amplificadas. Como consequência, os coeficientes a_k gerados podem sofrer variações drásticas, resultando em um polinômio que sequer satisfaz a condição de interpolação nos pontos tabelados.

Para contornar tal instabilidade, priorizam-se métodos numericamente mais robustos em aplicações práticas, como as formas de Lagrange ou de Newton (via diferenças divididas). Tais abordagens evitam a resolução direta de sistemas lineares mal condicionados, garantindo maior fidelidade aos dados originais. O exemplo a seguir ilustra como o mau condicionamento pode comprometer a solução em cenários de precisão limitada.

Exemplo 1.4.1. Determinaremos o polinômio interpolador de grau 3, denotado por $p_3(x)$, associado a uma função f cujos valores são apresentados na Tabela 2.

Tabela 2 – Dados de interpolação para obtenção de $p_3(x)$.

x	0.1	0.2	0.3	0.4
$f(x)$	5	13	-4	-8

Fonte: [5].

O polinômio interpolador é escrito na forma

$$p_3(x) = a_0 + a_1x + a_2x^2 + a_3x^3.$$

Impondo a condição de interpolação $p_3(x_i) = y_i$ para cada ponto da Tabela 2, obtém-se o sistema linear:

$$\begin{aligned} a_0 + 0.1a_1 + 0.01a_2 + 0.001a_3 &= 5, \\ a_0 + 0.2a_1 + 0.04a_2 + 0.008a_3 &= 13, \\ a_0 + 0.3a_1 + 0.09a_2 + 0.027a_3 &= -4, \\ a_0 + 0.4a_1 + 0.16a_2 + 0.064a_3 &= -8. \end{aligned}$$

Reescrevendo o sistema em termos de potências de 10, evidencia-se de forma explícita a estrutura da Matriz de Vandermonde:

$$\begin{cases} a_0 + 1 \cdot 10^{-1}a_1 + 1 \cdot 10^{-2}a_2 + 1 \cdot 10^{-3}a_3 = 5, \\ a_0 + 2 \cdot 10^{-1}a_1 + 4 \cdot 10^{-2}a_2 + 8 \cdot 10^{-3}a_3 = 13, \\ a_0 + 3 \cdot 10^{-1}a_1 + 9 \cdot 10^{-2}a_2 + 27 \cdot 10^{-3}a_3 = -4, \\ a_0 + 4 \cdot 10^{-1}a_1 + 16 \cdot 10^{-2}a_2 + 64 \cdot 10^{-3}a_3 = -8. \end{cases} \quad (1.12)$$

Esse sistema pode ser escrito na forma matricial $V\mathbf{a} = \mathbf{y}$, onde V é a Matriz de Vandermonde associada aos nós de interpolação:

$$\begin{pmatrix} 1 & 1 \cdot 10^{-1} & 1 \cdot 10^{-2} & 1 \cdot 10^{-3} \\ 1 & 2 \cdot 10^{-1} & 4 \cdot 10^{-2} & 8 \cdot 10^{-3} \\ 1 & 3 \cdot 10^{-1} & 9 \cdot 10^{-2} & 27 \cdot 10^{-3} \\ 1 & 4 \cdot 10^{-1} & 16 \cdot 10^{-2} & 64 \cdot 10^{-3} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 5 \\ 13 \\ -4 \\ -8 \end{pmatrix}.$$

Para a resolução do sistema associado à Matriz de Vandermonde (1.12), aplicaremos o método da eliminação de Gauss, conforme descrito por Ruggiero e Lopes [5, p. 119].

Após a aplicação do método, os coeficientes obtidos para o polinômio interpolador, apresentando as instabilidades de arredondamento, são dados por:

$$a_0 \approx -66$$

$$a_1 \approx 1150$$

$$a_2 \approx -5050$$

$$a_3 \approx 18666,67$$

Dessa forma, o polinômio interpolador associado aos dados da Tabela 2 é expresso por:

$$p_3(x) = 18666,67x^3 - 5050x^2 + 1150x - 66.$$

Observa-se que $p_3(0,4) \approx 780,67$, o que diverge drasticamente de $f(0,4) = -8$. Esse resultado não contradiz o teorema de existência e unicidade do polinômio interpolador, mas evidencia de forma extrema os efeitos de instabilidade numérica e de arredondamentos associados à resolução de sistemas lineares envolvendo matrizes de Vandermonde, que são, em geral, mal condicionadas.

Este exemplo crítico motiva a busca por métodos de interpolação numericamente mais estáveis. No próximo capítulo, serão apresentados e discutidos em detalhe os Métodos de Lagrange e de Newton para a determinação do polinômio interpolador de grau n .

2 Métodos de Interpolação Polinomial

Neste capítulo, serão apresentados os principais métodos construtivos utilizados na determinação explícita do polinômio interpolador associado a uma função f , a partir de um conjunto finito de pontos $\{(x_i, f(x_i))\}$. Em particular, serão estudados em detalhe os métodos clássicos de Lagrange e de Newton, os quais fornecem representações alternativas do polinômio interpolador, cada uma caracterizada por propriedades computacionais específicas e vantagens estruturais distintas. Ademais, será desenvolvida uma análise rigorosa do erro de interpolação, elemento fundamental para a compreensão da precisão das aproximações polinomiais e para o estudo do comportamento das soluções obtidas.

2.1 Método de Lagrange

Segundo a análise histórica apresentada por Burden et al. [8], a fórmula de interpolação classicamente atribuída a Joseph-Louis Lagrange (1736 – 1813) possui raízes históricas mais antigas. Há indícios de que Isaac Newton (1643 – 1727) já dominava o conceito em torno de 1675. No entanto, a primeira publicação formal e documentada dessa técnica é creditada a Edward Waring (1734 – 1798), que a divulgou em sua obra de 1779.

Lagrange, por sua vez, dedicou-se profundamente ao estudo da interpolação e publicou seus próprios resultados em 1795, influenciando de maneira decisiva o desenvolvimento posterior da matemática numérica. A principal vantagem metodológica da Fórmula de Lagrange reside na obtenção de uma forma explícita para o Polinômio Interpolador, eliminando a necessidade de resolver um sistema de equações lineares.

2.1.1 Polinômio Interpolador de Lagrange

Teorema 2.1.1 (Polinômio Interpolador de Lagrange). Sejam f uma função definida no conjunto de $n + 1$ abscissas distintas $\{x_i\}_{i=0}^n$ e

$$p_n(x) = \sum_{k=0}^n f(x_k)L_k(x), \quad (2.1)$$

onde

$$L_k(x) = \prod_{\substack{j=0 \\ j \neq k}}^n \frac{x - x_j}{x_k - x_j}, \quad (2.2)$$

para cada $k \in \{0, 1, \dots, n\}$. Então, $p_n(x)$ interpola f no conjunto $\{(x_i, f(x_i))\}_{i=0}^n$.

As formas $L_k(x)$ são conhecidas como os polinômios base de Lagrange [9, p. 174].

Demonstração. Observa-se que o polinômio $p_n(x)$, conforme definido, possui grau n , pois é construído como combinação linear dos polinômios base de Lagrange, cada um deles de grau exatamente n . Resta, portanto, demonstrar que esse polinômio satisfaz as condições de interpolação, isto é,

$$p_n(x_i) = f(x_i), \quad i = 0, 1, \dots, n.$$

Para isso, começaremos mostrando que $L_k(x)$, satisfaz a condição:

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

De fato, considerando $i = k$, temos:

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

Por outro lado, para $i \neq k$ ($i < k$ ou $i > k$), observamos que o numerador da expressão $L_k(x)$ exclui apenas o fator $(x - x_k)$. De modo que ao substituir x por x_i , algum destes fatores será $(x_i - x_i) = 0$ e, conseqüentemente, $L_k(x_i) = 0$.

Calculando $p_n(x_i)$, para $i = 0, \dots, n$, segue que:

$$\begin{aligned} p_n(x_i) &= \sum_{k=0}^n f(x_k) L_k(x_i) \\ &= f(x_i) L_i(x_i) + \sum_{\substack{k=0 \\ k \neq i}}^n f(x_k) L_k(x_i) \\ &= f(x_i) \cdot 1 + \sum_{\substack{k=0 \\ k \neq i}}^n f(x_k) \cdot 0 \\ &= f(x_i). \end{aligned}$$

Portanto, o polinômio $p_n(x)$ satisfaz a condição de interpolação em todos os nós.

□

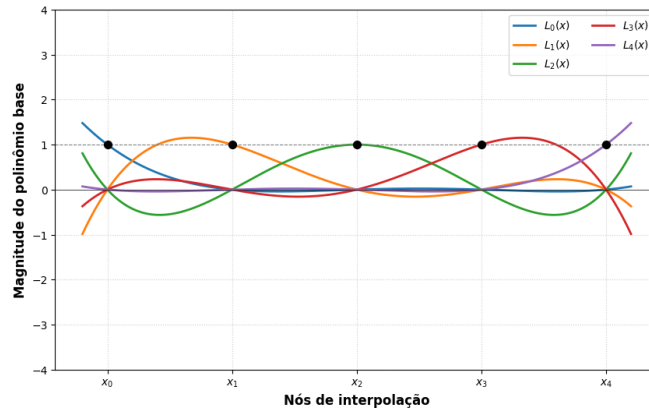
Uma vez estabelecida a expressão explícita de cada polinômio base $L_k(x)$, a construção do polinômio interpolador torna-se imediata. O polinômio assim obtido é denominado polinômio interpolador de Lagrange de ordem n [8]. Em particular, $p_n(x)$ é expresso como uma combinação linear dos valores da função nos nós, sendo cada termo ponderado pelo correspondente polinômio base $L_k(x)$.

Observa-se que a construção apresentada garante a existência do polinômio interpolador. A unicidade também decorre imediatamente de seus fundamentos. Com efeito, pela unicidade da solução do sistema linear associado (Seção 1.4), não podem existir dois polinômios distintos de grau no máximo n satisfazendo simultaneamente as mesmas condições de interpolação. Caso existisse um segundo polinômio interpolador, a diferença entre ambos seria um polinômio de grau no máximo n que se anula nos $n + 1$ nós. Pelo corolário do Teorema Fundamental da Álgebra, o qual garante que um polinômio de grau $n \geq 1$ tem exatamente n raízes complexas (contadas com multiplicidade) [1], tal polinômio deve ser identicamente nulo, o que confirma que o polinômio interpolador é único.

A análise da estrutura desses polinômios base revela, entretanto, uma limitação fundamental do método. Embora satisfaçam $L_k(x_k) = 1$ para $k = i$ e $L_k(x_i) = 0$ para $i \neq k$, suas magnitudes podem crescer significativamente entre os nós, sobretudo nas extremidades do intervalo. Esse comportamento implica que pequenos erros nos valores $f(x_k)$, ou erros de arredondamento, são multiplicados por fatores elevados, produzindo oscilações acentuadas no polinômio interpolador $p_n(x)$ — fenômeno conhecido como Fenômeno de Runge, o qual será discutido em seções posteriores.

Esse efeito pode ser visualmente compreendido por meio da observação direta dos polinômios base de Lagrange construídos sobre nós equidistantes. A Figura 4 ilustra o comportamento desses polinômios para um conjunto de 5 nós igualmente espaçados, evidenciando o crescimento acentuado de sua magnitude fora dos nós. Gráficos desse tipo, semelhantes aos apresentados em Burdem [8, p. 120], demonstram de forma clara a tendência oscilatória responsável pela instabilidade numérica da interpolação polinomial de alto grau.

Figura 4 – Comportamento dos polinômios base de Lagrange em nós equidistantes.



Fonte: Autores.

Ao examinar a Figura 4, observa-se que, embora cada polinômio base $L_k(x)$ cumpra rigorosamente a propriedade de assumir valor unitário em seu nó respectivo e zero nos demais, o comportamento entre esses pontos revela a fragilidade do método para ordens mais elevadas. Nota-se que, nos subintervalos periféricos — especificamente entre x_0 e x_1 , e entre x_3 e x_4 —, as curvas apresentam amplitudes significativamente maiores do que na região central do domínio.

Particularmente, os polinômios $L_0(x)$ e $L_4(x)$ demonstram uma tendência de crescimento abrupto ao se aproximarem das extremidades do intervalo. Como o polinômio interpolador final é uma combinação linear dessas bases, qualquer incerteza nos dados de entrada (y_k) será amplificada por esses fatores de escala elevados, resultando em uma representação que oscila violentamente. Essa visualização torna explícita a razão pela qual a interpolação por polinômios globais de alto grau é frequentemente substituída por técnicas de interpolação por partes, como os *splines*, que mantêm a estabilidade ao utilizar polinômios de baixo grau em cada subintervalo.

Exemplo 2.1.1. Este exemplo utiliza o método de Lagrange para determinar o polinômio interpolador $p_2(x)$ para o conjunto de dados apresentados na Tabela 3.

Tabela 3 – Dados de interpolação para obter $p_2(x)$.

x	-1	0	2
$f(x)$	4	1	-1

Fonte: [5].

Denotando as abscissas dos nós de interpolação por $x_0 = -1$, $x_1 = 0$ e $x_2 = 2$, bem como seus respectivos valores $y_0 = 4$, $y_1 = 1$ e $y_2 = -1$, seja $p_2(x)$ o polinômio interpolador obtido pelo método de Lagrange. Inicialmente, calculam-se os polinômios

base de Lagrange $L_k(x)$, para $k = 0, 1, 2$:

$$\begin{aligned} L_0(x) &= \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} = \frac{(x - 0)(x - 2)}{(-1 - 0)(-1 - 2)} = \frac{x^2 - 2x}{3}, \\ L_1(x) &= \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} = \frac{(x + 1)(x - 2)}{(0 + 1)(0 - 2)} = -\frac{x^2 - x - 2}{2}, \\ L_2(x) &= \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} = \frac{(x + 1)x}{(2 + 1) \cdot 2} = \frac{x^2 + x}{6}. \end{aligned}$$

Como

$$p_2(x) = \sum_{k=0}^2 y_k L_k(x),$$

substituindo os valores de y_k e dos polinômios base $L_k(x)$ e realizando as simplificações algébricas necessárias, obtém-se:

$$\begin{aligned} p_2(x) &= 4L_0(x) + L_1(x) - L_2(x) \\ &= 4 \left(\frac{x^2 - 2x}{3} \right) - \frac{1}{2} (x^2 - x - 2) - \frac{1}{6} (x^2 + x) \\ &= \left(\frac{4}{3}x^2 - \frac{8}{3}x \right) + \left(-\frac{1}{2}x^2 + \frac{1}{2}x + 1 \right) + \left(-\frac{1}{6}x^2 - \frac{1}{6}x \right) \\ &= \left(\frac{4}{3} - \frac{1}{2} - \frac{1}{6} \right) x^2 + \left(-\frac{8}{3} + \frac{1}{2} - \frac{1}{6} \right) x + 1 \\ &= \frac{2}{3}x^2 - \frac{7}{3}x + 1. \end{aligned}$$

Logo, a expressão final do polinômio interpolador é:

$$p_2(x) = \frac{2}{3}x^2 - \frac{7}{3}x + 1.$$

Pela unicidade do polinômio interpolador, esse resultado corresponde ao único polinômio de grau no máximo 2 que interpola os três pontos considerados.

2.1.2 Um modelo de calibração de termômetro

Segundo o VIM [10], o processo de calibração constitui uma etapa central na Metrologia, por garantir a confiabilidade dos instrumentos de medição, como termômetros. Calibrar um instrumento consiste em estabelecer, sob condições especificadas, a relação entre os valores por ele indicados e os valores correspondentes da grandeza fornecidos por padrões metrológicos. Essa operação permite determinar o erro e a incerteza de medição associados ao instrumento. O erro corresponde à diferença entre o valor indicado e o valor verdadeiro de referência. A incerteza, por sua vez, expressa o intervalo no qual o valor verdadeiro é estimado estar, levando em conta todas as fontes de variação do processo

de medição. Assim, a calibração assegura a rastreabilidade metrológica do instrumento. A exatidão obtida é particularmente relevante em contextos industriais, farmacêuticos e químicos, nos quais desvios podem comprometer tanto a qualidade dos produtos quanto a segurança operacional.

Em muitos casos, a relação entre a leitura do instrumento e o valor de referência não é perfeitamente linear. Assim, a calibração é realizada em pontos discretos e conhecidos ao longo da faixa de medição. Para obter o valor real em temperaturas intermediárias que não foram testadas (isto é, para construir uma curva de correção contínua), aplica-se a Interpolação Numérica. O Polinômio Interpolador de Lagrange é particularmente adequado para esse fim, pois fornece uma função que passa exatamente pelos pontos de calibração e permite estimar a temperatura real correspondente a qualquer leitura do termômetro dentro da faixa interpolada, garantindo a coerência e a correção da curva de calibração.

Para ilustrar a aplicação do método de Lagrange em um contexto metrológico, consideremos o processo de calibração de um termômetro cuja resposta apresenta comportamento não linear. Suponha que foram determinados valores pareados entre a temperatura real e a temperatura indicada pelo instrumento. A Tabela 4 apresenta os dados utilizados no procedimento de calibração, comparando as temperaturas reais fornecidas por um padrão rastreado e as temperaturas registradas pelo termômetro em análise. Nesse contexto, a interpolação é realizada a partir de um conjunto de pontos discretos $(x_i, f(x_i))$, onde definimos:

- x : a leitura do instrumento (temperatura medida em °C), que corresponde aos nós de interpolação;
- $f(x)$: a temperatura padrão (temperatura real em °C), que representa o valor de referência corrigido.

Tabela 4 – Comparação entre a temperatura medida (x) e a temperatura padrão ($f(x)$) ambas em °C.

x	0	10	20	30	40
$f(x)$	17	32	1	54	245

Fonte: Autores.

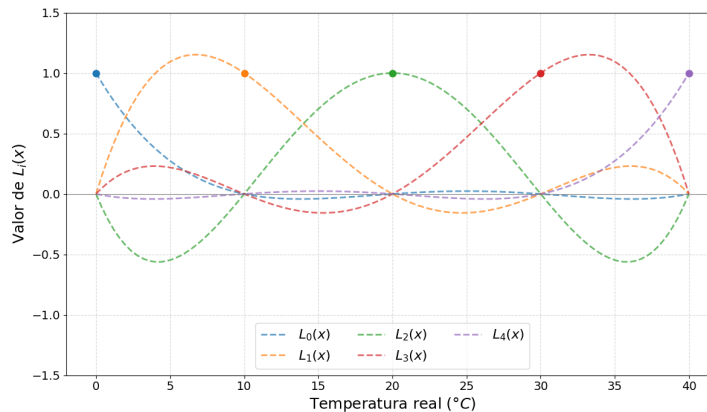
O processo de interpolação será realizado em três etapas. Inicialmente, calcularemos os polinômios de Lagrange de grau quatro para cada ponto de calibração. Em seguida, utilizaremos esses componentes para determinar a expressão final do Polinômio Interpolador. Por fim, apresentaremos o gráfico do polinômio resultante para permitir uma análise visual da sua adequação à não-linearidade dos dados originais.

Os polinômios de Lagrange, construídos com base nos dados da Tabela 4, são expressos por:

$$\begin{aligned} L_0(x) &= \frac{(x-10)(x-20)(x-30)(x-40)}{240000}, \\ L_1(x) &= \frac{x(x-20)(x-30)(x-40)}{-60000}, \\ L_2(x) &= \frac{x(x-10)(x-30)(x-40)}{40000}, \\ L_3(x) &= \frac{x(x-10)(x-20)(x-40)}{-60000}, \\ L_4(x) &= \frac{x(x-10)(x-20)(x-30)}{240000}. \end{aligned}$$

A Figura 5 ilustra as curvas dos polinômios de base de Lagrange ($L_i(x)$), destacando como cada polinômio assume o valor de 1 em seu respectivo nó x_i e zero nos demais.

Figura 5 – Polinômios de base de Lagrange ($L_k(x)$) para os nós $[0, 10, 20, 30, 40]$.



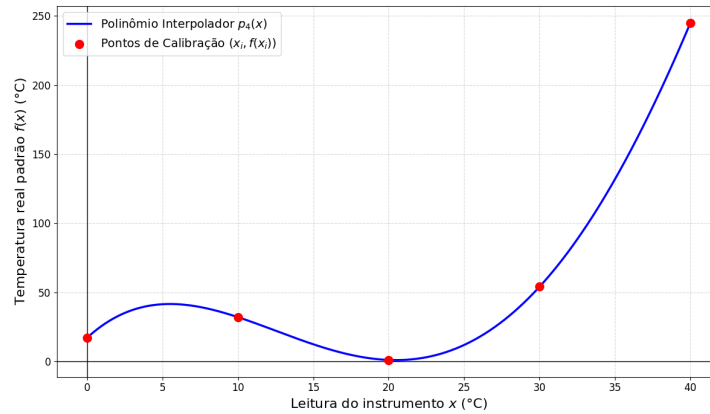
Fonte: Autores.

O polinômio interpolador de Lagrange, $p_4(x)$, é obtido pela soma ponderada dos polinômios de base ($L_k(x)$) pelos valores da função ($f(x_i)$). Sua expressão é dada por:

$$\begin{aligned} p_4(x) &= 17L_0(x) + 32L_1(x) + 1L_2(x) + 54L_3(x) + 245L_4(x) \\ &= -\frac{19}{60000}x^4 + \frac{61}{1500}x^3 - \frac{737}{600}x^2 + \frac{301}{30}x + 17. \end{aligned}$$

A Figura 6 apresenta a curva de calibração obtida a partir do polinômio interpolador de Lagrange $p_4(x)$.

Figura 6 – Curva de calibração do termômetro por Interpolação de Lagrange.



Fonte: Autores.

Para que a curva de correção possa ser utilizada na prática, calcularemos o valor da temperatura padrão $f(x)$ em pontos de leitura do instrumento x que não constam na Tabela 4.

A curva $p_4(x)$ foi determinada para passar exatamente pelos cinco pontos de calibração fornecidos. No entanto, a não-linearidade extrema dos dados (notadamente a queda de 32°C para 1°C entre $x = 10$ e $x = 20$) impõe uma tensão severa ao modelo polinomial de grau 4. Visualmente, o gráfico exibe um comportamento instável e oscilatório entre os nós, característico de polinômios de alta ordem que, ao tentarem satisfazer a continuidade global, introduzem oscilações espúrias em resposta a variações locais abruptas.

Percebe-se a inadequação deste método de interpolação (modelo) na previsão de valores intermediários. Ao testar o modelo para uma leitura de $x = 5^\circ\text{C}$, obtemos $p_4(5) \approx 42,05^\circ\text{C}$. Embora este valor devesse situar-se entre os limites dos nós vizinhos ($f(0) = 17$ e $f(10) = 32$), a curvatura imposta pelo polinômio resulta em um valor que excede ambos, criando uma “corcova” artificial. O modelo assume um pico de leitura que não reflete o crescimento linear esperado ($\approx 24,5^\circ\text{C}$) e que dificilmente representaria o comportamento térmico de um sensor real, reforçando a tese de que a interpolação global de Lagrange falha em capturar a física do processo por trás dos dados de calibração devido às oscilações espúrias.

Em termos metrológicos, essa distorção não apenas compromete a estimativa, mas também aumenta a incerteza de medição do instrumento, pois o modelo de calibração introduz um erro sistemático substancial em leituras fora dos nós. A alternativa mais robusta para modelar tal comportamento não-linear e garantir a suavidade da curva seria a utilização de *splines* cúbicos, que utilizam polinômios de baixo grau em segmentos, minimizando a oscilação e fornecendo um ajuste local mais fiel aos dados. Este método será abordado ainda neste trabalho.

Sobre o método de Lagrange, observa-se que ele permite determinar de forma explícita o polinômio interpolador de grau n a partir de $n + 1$ nós distintos. Sua formulação direta, baseada nos polinômios fundamentais $L_k(x)$, facilita tanto a construção inicial do polinômio quanto a verificação da condição de interpolação. Entretanto, esse método apresenta limitações operacionais relevantes, destacando-se a necessidade de recalcular integralmente o polinômio sempre que um novo ponto é acrescentado ao conjunto de dados, bem como a possibilidade de instabilidade numérica associada à sua aplicação, especialmente em problemas com maior número de nós. Tais aspectos restringem seu uso em situações que exigem maior eficiência computacional e robustez numérica.

Na sequência, apresentamos o método de Newton, que oferece uma abordagem mais eficiente para contornar essas limitações.

2.2 Método de Newton

Diferentemente da formulação de Lagrange, o método de Newton para interpolação polinomial permite a construção do polinômio interpolador de forma incremental, possibilitando a atualização do polinômio à medida que novos nós de interpolação são adicionados, sem a necessidade de refazer todos os cálculos anteriores.

Nessa formulação, o Polinômio Interpolador de Newton $p_n(x)$ é expresso de modo sequencial por meio das diferenças divididas, assumindo a forma:

$$p_n(x) = d_0 + d_1(x - x_0) + d_2(x - x_0)(x - x_1) + \cdots + d_n(x - x_0)(x - x_1) \cdots (x - x_{n-1}), \quad (2.3)$$

em que os coeficientes d_k , para $k = 0, 1, \dots, n$, correspondem às diferenças divididas de ordem k da função f associadas aos nós x_0, x_1, \dots, x_k .

2.2.1 Diferenças divididas

De forma engenhosa, a fórmula do Polinômio de Newton na Equação 2.3 é estruturada de modo que as constantes d_0, d_1, \dots, d_n possam ser determinadas sucessivamente pelas condições de interpolação.

Por exemplo, para determinar o coeficiente d_0 , avaliamos $p_n(x)$ no primeiro nó x_0 , isto é:

$$d_0 = p_n(x_0) = f(x_0). \quad (2.4)$$

De modo análogo, ao calcularmos $p_n(x)$ em x_1 , temos que:

$$p_n(x_1) = f(x_0) + d_1(x_1 - x_0) = f(x_1).$$

Dessa forma, o coeficiente d_1 é definido como:

$$d_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}. \quad (2.5)$$

Dando continuidade, ao calcular $p_n(x)$ em x_2 , obtemos:

$$\begin{aligned} p_n(x_2) &= d_0 + d_1(x_2 - x_0) + d_2(x_2 - x_0)(x_2 - x_1) \\ &= f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x_2 - x_0) + d_2(x_2 - x_0)(x_2 - x_1) \\ &= f(x_2). \end{aligned}$$

Ao isolar d_2 , temos que:

$$\begin{aligned} d_2 &= \frac{f(x_2) - \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x_2 - x_0) - f(x_0)}{(x_2 - x_0)(x_2 - x_1)} \\ &= \frac{f(x_2) - (f(x_1) - f(x_0))\frac{x_2 - x_0}{x_1 - x_0} - f(x_0)}{(x_2 - x_0)(x_2 - x_1)} \end{aligned}$$

Além disso, se considerarmos:

$$\frac{x_2 - x_0}{x_1 - x_0} = \frac{x_2 - x_1 + x_1 - x_0}{x_1 - x_0} = \frac{x_2 - x_1}{x_1 - x_0} + 1.$$

ao substituir em d_2 , segue que:

$$\begin{aligned} d_2 &= \frac{f(x_2) - (f(x_1) - f(x_0))\left(\frac{x_2 - x_1}{x_1 - x_0} + 1\right) - f(x_0)}{(x_2 - x_0)(x_2 - x_1)} \\ &= \frac{f(x_2) - (f(x_1) - f(x_0))\frac{x_2 - x_1}{x_1 - x_0} - \cancel{f(x_1)} + \cancel{f(x_0)} - f(x_0)}{(x_2 - x_0)(x_2 - x_1)} \\ &= \frac{f(x_2) - f(x_1) - (f(x_1) - f(x_0))\frac{x_2 - x_1}{x_1 - x_0}}{(x_2 - x_0)(x_2 - x_1)} \end{aligned}$$

Finalmente, passando $(x_2 - x_1)$ para o numerador, chegamos à seguinte expressão:

$$d_2 = \frac{\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0},$$

que pode ser reescrita como:

$$d_2 = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}.$$

Para generalizar a representação desses coeficientes, apresentamos a notação formal para as diferenças divididas. Dito isso, a diferença dividida de ordem zero da função f em relação a x_i , denotada por $f[x_i]$, é simplesmente o valor da função nesse ponto:

$$f[x_i] = f(x_i). \quad (2.6)$$

As demais diferenças divididas são definidas de forma indutiva. A primeira diferença dividida, denotada por $f[x_i, x_{i+1}]$, é definida como:

$$f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}. \quad (2.7)$$

A segunda diferença dividida é uma extensão recursiva, usando diferenças de primeira ordem:

$$f[x_i, x_{i+1}, x_{i+2}] = \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i}. \quad (2.8)$$

De forma geral, a diferença dividida de ordem k sobre os $k + 1$ nós $x_i, x_{i+1}, \dots, x_{i+k}$ é definida recursivamente como:

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

O processo finaliza com a única n -ésima diferença dividida:

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

Portanto, os coeficientes do polinômio de Newton, $d_k = f[x_0, x_1, \dots, x_k]$, são definidos como:

$$\begin{aligned} d_0 &= f[x_0] = f(x_0) && \text{(Ordem 0)} \\ d_1 &= f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0} = \frac{f(x_1) - f(x_0)}{x_1 - x_0} && \text{(Ordem 1)} \\ d_2 &= f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} && \text{(Ordem 2)} \\ &\vdots && \vdots \\ d_n &= f[x_0, x_1, x_2, \dots, x_n] = \frac{f[x_1, x_2, \dots, x_n] - f[x_0, x_1, x_2, \dots, x_{n-1}]}{x_n - x_0} && \text{(Ordem } n\text{).} \end{aligned}$$

A recursividade da fórmula das diferenças divididas confere ao método de Newton uma grande vantagem computacional em relação à forma de Lagrange. Ao introduzir um novo ponto de interpolação, x_{n+1} , o novo polinômio $p_{n+1}(x)$ pode ser obtido simplesmente adicionando o termo $d_{n+1}(x - x_0) \cdots (x - x_n)$ ao polinômio $p_n(x)$ já calculado, de modo que os coeficientes d_0, \dots, d_n permanecem inalterados.

Conhecidos os valores da função f nos $n + 1$ nós, x_0, x_1, \dots, x_n , os coeficientes d_k são calculados sucessivamente através da construção de uma tabela de diferenças divididas. Esta tabela organiza o cálculo recursivo, disposta convencionalmente da esquerda para a direita, conforme esboçado na Tabela 5.

Tabela 5 – Estrutura do cálculo das diferenças divididas.

x	Ordem 0	Ordem 1	Ordem 2	...	Ordem n
x_0	$f[x_0]$				
x_1	$f[x_1]$	$f[x_0, x_1]$			
x_2	$f[x_2]$	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$		
x_3	$f[x_3]$	$f[x_2, x_3]$	$f[x_1, x_2, x_3]$		
x_4	$f[x_4]$	$f[x_3, x_4]$	$f[x_2, x_3, x_4]$...	
\vdots	\vdots	\vdots	\vdots	\ddots	\vdots
x_n	$f[x_n]$	$f[x_{n-1}, x_n]$	$f[x_{n-2}, x_{n-1}, x_n]$...	$f[x_0, x_1, \dots, x_n]$

Fonte: Adaptado de [5].

Além disso, embora estabelecida a unicidade do Polinômio Interpolador, podemos perceber que o processo construtivo de Newton é invariante à ordem dos nós de interpolação x_0, x_1, \dots, x_n . Para esta demonstração, veja a Proposição A.3.1 no Apêndice A.

2.2.2 Polinômio Interpolador de Newton

Teorema 2.2.1 (Polinômio Interpolador de Newton). Seja f uma função definida em um conjunto de $n + 1$ abscissas distintas $\{x_i\}_{i=0}^n$. Considere o polinômio

$$p_n(x) = f(x_0) + \sum_{k=1}^n d_k \prod_{i=0}^{k-1} (x - x_i), \quad (2.9)$$

em que o coeficiente $d_k = f[x_0, x_1, \dots, x_k]$ representa a diferença dividida de ordem k associada aos nós x_0, x_1, \dots, x_k . Então, o polinômio $p_n(x)$ interpola a função f no conjunto de pontos $\{(x_i, f(x_i))\}_{i=0}^n$.

Demonstração. Precisamos verificar que $p_n(x_j) = f(x_j)$ para todo $j \in \{0, 1, \dots, k\}$. Isso é feito por substituição direta na Equação (2.9), aproveitando o fato de que, em um nó x_j , todos os termos que contém o fator $(x - x_j)$ se anulam.

Ao substituirmos no primeiro nó, $x = x_0$, todos os termos, exceto o primeiro, contém o fator $(x_0 - x_0)$. Dessa forma:

$$p_n(x_0) = f(x_0) + \sum_{k=1}^n d_k \prod_{i=0}^{k-1} (x_0 - x_i) = f(x_0).$$

Avaliando em $x = x_1$, todos os termos de ordem $k \geq 2$ se anulam devido ao fator

$(x - x_1)$:

$$\begin{aligned}
 p_n(x_1) &= f(x_0) + f[x_0, x_1](x_1 - x_0) + \sum_{k=2}^n d_k \prod_{i=0}^{k-1} (x_1 - x_i) \\
 &= f(x_0) + \left(\frac{f(x_1) - f(x_0)}{x_1 - x_0} \right) (x_1 - x_0) \\
 &= f(x_0) + f(x_1) - f(x_0) \\
 &= f(x_1) \\
 &= y_1.
 \end{aligned}$$

Analogamente, ao avaliarmos em $x = x_2$, todos os termos de ordem $k \geq 3$ se anulam devido ao fator $(x - x_2)$:

$$\begin{aligned}
 p_n(x_2) &= f(x_0) + f[x_0, x_1](x_2 - x_0) + f[x_0, x_1, x_2](x_2 - x_0)(x_2 - x_1) + \sum_{k=3}^n d_k \prod_{i=0}^{k-1} (x_2 - x_i) \\
 &= f(x_0) + f[x_0, x_1](x_2 - x_0) + \left(\frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} \right) (x_2 - x_0)(x_2 - x_1) \\
 &= f(x_0) + f[x_0, x_1](x_2 - x_0) + (f[x_1, x_2] - f[x_0, x_1])(x_2 - x_1) \\
 &= f(x_0) + f[x_0, x_1]((x_2 - x_0) - (x_2 - x_1)) + f[x_1, x_2](x_2 - x_1) \\
 &= f(x_0) + f[x_0, x_1](x_1 - x_0) + f[x_1, x_2](x_2 - x_1) \\
 &= f(x_0) + \left(\frac{f(x_1) - f(x_0)}{x_1 - x_0} \right) (x_1 - x_0) + \left(\frac{f(x_2) - f(x_1)}{x_2 - x_1} \right) (x_2 - x_1) \\
 &= f(x_0) + f(x_1) - f(x_0) + f(x_2) - f(x_1) \\
 &= f(x_2) \\
 &= y_2.
 \end{aligned}$$

De modo geral, para qualquer nó x_j , onde $j \in \{0, 1, \dots, k\}$, ao avaliarmos a Equação (2.9) em $x = x_j$, todos os termos da soma cujo índice k é maior que j ($k \geq j + 1$) se anulam. Isso ocorre porque o produtório $\prod_{i=0}^{k-1} (x_j - x_i)$ conterà o fator $(x_j - x_j)$, que é zero. Portanto, temos:

$$p_n(x_j) = f(x_0) + \sum_{k=1}^n d_k \prod_{i=0}^{k-1} (x_j - x_i),$$

de modo que a soma se restringe aos termos que não contêm o fator $(x_j - x_j)$, ou seja, $k \leq j$, resultando em:

$$p_n(x_j) = \underbrace{f(x_0) + \sum_{k=1}^j d_k \prod_{i=0}^{k-1} (x_j - x_i)}_{p_j(x_j)} + \sum_{k=j+1}^n d_k \cdot 0.$$

A expressão restante coincide, por definição, com o polinômio interpolador de Newton de grau j , denotado por $p_j(x)$, avaliado no ponto x_j . Como $p_j(x)$ é construído de modo a

interpolando os nós x_0, x_1, \dots, x_j , tem-se que essa condição é satisfeita e, conseqüentemente,

$$p_n(x_j) = p_j(x_j) = f(x_j).$$

Isso prova que o polinômio de Newton de ordem n preserva as condições de interpolação de todas as ordens anteriores, consolidando sua natureza incremental.

□

Exemplo 2.2.1. Iremos obter o polinômio interpolador de grau no máximo 2 que interpola a função $f(x)$ nos nós dados pela Tabela 6, utilizando a fórmula das diferenças divididas de Newton.

Tabela 6 – Dados de interpolação para obter $p_2(x)$ pelo método de Newton.

x	-1	0	3
$f(x)$	15	8	-1

Fonte: [5].

Os nós de interpolação são definidos como $x_0 = -1$, $x_1 = 0$, e $x_2 = 3$, com os respectivos valores da função $f(x_0) = 15$, $f(x_1) = 8$, e $f(x_2) = -1$.

A forma geral para o polinômio de Newton de grau $n = 2$ é:

$$p_2(x) = f[x_0] + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2].$$

Os coeficientes d_k são determinados pela construção da tabela de diferenças divididas.

$$\text{Ordem 0: } f[x_0] = f(x_0) = 15$$

$$\text{Ordem 1: } f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{8 - 15}{0 - (-1)} = -7$$

$$\text{Ordem 2: } f[x_1, x_2] = \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{-1 - 8}{3 - 0} = -3$$

$$\text{Ordem 3: } f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{-3 - (-7)}{3 - (-1)} = 1.$$

A tabela de diferenças divididas, contendo os coeficientes na primeira diagonal, é apresentada na Tabela 7:

Tabela 7 – Tabela de diferenças divididas para $f(x)$.

x	Ordem 0	Ordem 1	Ordem 2
-1	15		
		-7	
0	8		1
		-3	
3	-1		

Fonte: [5].

Substituindo os coeficientes ($d_0 = 15$, $d_1 = -7$, $d_2 = 1$) na fórmula de Newton, obtemos o Polinômio Interpolador de grau no máximo 2 que interpola $f(x)$ em todos os seus nós:

$$\begin{aligned}
 p_2(x) &= 15 + (x - (-1))(-7) + (x - (-1))(x - 0) \cdot 1 \\
 &= 15 - 7(x + 1) + x(x + 1) \\
 &= 15 - 7x - 7 + x^2 + x \\
 &= x^2 - 6x + 8.
 \end{aligned}$$

2.2.3 Um modelo para a média móvel dos casos de COVID-19

Nesta seção, apresentamos uma aplicação prática da interpolação polinomial pelo Método de Newton na análise do registro semanal de casos de COVID-19 no estado de Pernambuco (PE). Para isso, utilizamos o *Python* como ferramenta computacional, aplicando o método aos valores da média móvel de casos referentes à segunda semana dos meses de fevereiro a agosto nos anos de 2021, 2023 e 2025. Esses dados foram obtidos a partir das bases oficiais disponibilizadas pelo Ministério da Saúde [11].

O conjunto de dados de casos semanais manifesta flutuações de alta frequência (ruído) e um efeito de sazonalidade de curto prazo, sendo este último tipicamente induzido por vieses operacionais, como o atraso na notificação e a redução na testagem e no processamento laboratorial em fins de semana e feriados. Para mitigar esse ruído, suavizar a série temporal e, assim, isolar a tendência epidemiológica fundamental, aplicou-se o método da Média Móvel.

Conforme definido por Morettin e Tolo [12, p. 90], a Média Móvel é uma técnica de suavização utilizada em séries temporais que consiste em substituir o valor de um ponto (período) pela média aritmética dos pontos adjacentes. Dito isto, a Média Móvel Simples (MMS_t) de ordem k para uma série Y_t no tempo t é definida como:

$$MMS_t = \frac{1}{k} \sum_{i=0}^{k-1} Y_{t-i}.$$

A principal função deste procedimento é eliminar as flutuações de curto prazo e destacar a tendência de longo prazo ou ciclos, tornando a série mais estável e adequada para a análise de interpolação subsequente com o Método de Newton. Desta maneira, obtém-se uma visão mais estável e confiável sobre a velocidade da transmissão, permitindo às autoridades de saúde identificar se a epidemia está em uma fase de crescimento, declínio ou estabilidade.

Considere os dados apresentados na Tabela 8, os quais correspondem aos valores da Média Móvel Semanal (MMS) de casos de COVID-19 referentes às segundas semanas epidemiológicas dos meses selecionados ao longo dos três anos analisados.

Tabela 8 – Média Móvel Semanal do número de casos de COVID-19 por mês e ano (2ª Semana).

Mês	Semana	Número de casos		
		2021	2023	2025
Fevereiro	6	9 700	1 920	615
Março	10	8 440	1 944	26
Abril	14 / 15	13 718	1 255	91
Mai	18 / 19	14 926	3 677	121
Junho	23	20 515	1 203	67
Julho	27 / 28	11 647	1 456	110
Agosto	31 / 32	4 734	709	308

Fonte: Ministério da Saúde [11].

Os dados desta tabela foram obtidos diretamente do portal de informações do Ministério da Saúde, onde estão disponíveis os registros de casos por semana epidemiológica. Especificamente, utilizamos os valores da MMS previamente calculada pelo próprio órgão para o monitoramento da epidemia. Selecionamos estrategicamente as segundas semanas epidemiológicas dos meses de interesse a fim de capturar pontos relevantes nas fases de crescimento, pico e declínio da curva de casos. A utilização da MMS nesses pontos garante que os valores de y_i usados na interpolação já estão suavizados, oferecendo uma base mais confiável para o modelo preditivo.

Ao analisar os dados apresentados, observa-se que as semanas epidemiológicas correspondentes à segunda semana de cada mês não são as mesmas em todos os anos considerados. Isso ocorre porque o calendário epidemiológico é definido anualmente com base em critérios específicos, o que pode levar a variações na numeração das semanas entre diferentes anos. Assim, por exemplo, em abril a segunda semana corresponde à 14ª semana em 2021 e 2023, mas à 15ª em 2025; situação semelhante ocorre em maio, julho e agosto, cujas semanas epidemiológicas também apresentam diferenças entre os anos analisados.

Essas variações foram mantidas na tabela para preservar a fidelidade aos dados originais e garantir a consistência da análise.

Com base nos valores discretos e suavizados de MMS apresentados na Tabela 8, procederemos à construção do polinômio interpolador pelo Método de Newton, com o processo implementado integralmente em código *Python* (disponível no Apêndice B). Para cada ano analisado, será exibida a respectiva tabela de diferenças divididas, a expressão do polinômio resultante e o seu gráfico. Em seguida, examinaremos a adequação do modelo, isto é, em que medida o polinômio obtido representa de forma coerente a tendência observada nos dados, evitando oscilações excessivas associadas ao Fenômeno de Runge. Também discutiremos a validade das estimativas interpoladas no contexto epidemiológico, reconhecendo tanto seu potencial descritivo quanto suas limitações no tratamento de fenômenos reais.

Para o ano de 2021, a tabela de diferenças divididas é apresentada a seguir:

Tabela 9 – Tabela de diferenças divididas da média móvel de casos de COVID-19 — Ano de 2021.

x	Ordem 0	Ordem 1	Ordem 2	Ordem 3	Ordem 4	Ordem 5	Ordem 6
6	9700	-315	$\frac{3269}{16}$	$\frac{221}{8}$	$\frac{415409}{159120}$	$\frac{37679}{139230}$	$\frac{4309}{176800}$
10	8440	$\frac{2639}{2}$	$\frac{2035}{16}$	$\frac{156839}{9360}$	$\frac{488887}{159120}$	$\frac{754471}{2227680}$	
14	13718	302	$\frac{7493}{240}$	$\frac{76357}{9360}$	$\frac{104093}{79560}$		
18	14926	$\frac{5589}{5}$	$\frac{5558}{15}$	$\frac{15937}{480}$			
23	20515	-2217	$\frac{1955}{32}$				
27	11647	$\frac{6913}{4}$					
31	4734						

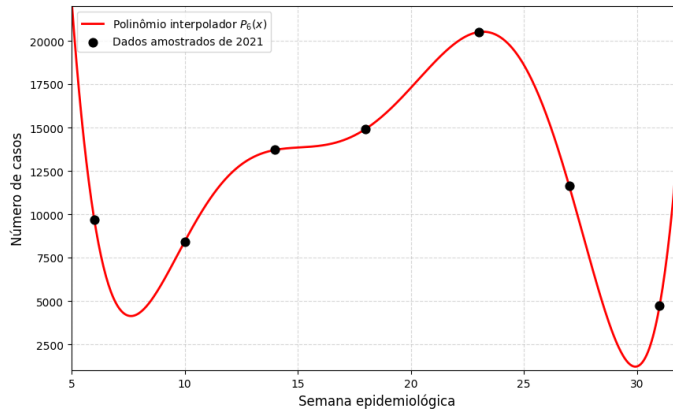
Fonte: Elaborada pelos autores com base nos dados do Ministério da Saúde [11].

Assim, o polinômio que modela os dados de 2021, $p_6^{2021}(x)$, a partir dos pontos de dados selecionados, é dado por:

$$p_6^{2021}(x) = \frac{4309}{176800}x^6 - \frac{14809043}{5569200}x^5 + \frac{257377193}{2227680}x^4 - \frac{1420359091}{556920}x^3 + \frac{167169994879}{5569200}x^2 - \frac{40983967153}{232050}x + \frac{1809911557}{4420}.$$

A curva do polinômio interpolador para esse conjunto de dados é dada pelo gráfico:

Figura 7 – Polinômio interpolador de Newton para a média móvel dos casos de COVID-19 em 2021.



Fonte: Elaborada pelos autores com base nos dados do Ministério da Saúde [11].

Para o ano de 2023, a tabela de diferenças divididas é dada por:

Tabela 10 – Tabela de diferenças divididas da média móvel de casos de COVID-19 — Ano de 2023.

x	Ordem 0	Ordem 1	Ordem 2	Ordem 3	Ordem 4	Ordem 5	Ordem 6
6	1920	6	$\frac{713}{32}$	$\frac{137149}{18720}$	$\frac{139559}{106880}$	$\frac{41641}{278460}$	$\frac{64835503}{5734048320}$
10	1944	$\frac{689}{4}$	$\frac{13133}{180}$	$\frac{2707}{180}$	$\frac{64523}{35360}$	$\frac{4550833}{31505760}$	
14	1255	$\frac{2422}{5}$	$\frac{11029}{90}$	$\frac{299179}{18720}$	$\frac{227959}{168480}$		
19	3677	$\frac{1237}{2}$	$\frac{2727}{32}$	$\frac{18720}{156739}$			
23	1203	$\frac{253}{4}$	$\frac{4253}{180}$	$\frac{18720}{18720}$			
27	1456	$\frac{747}{5}$					
32	709						

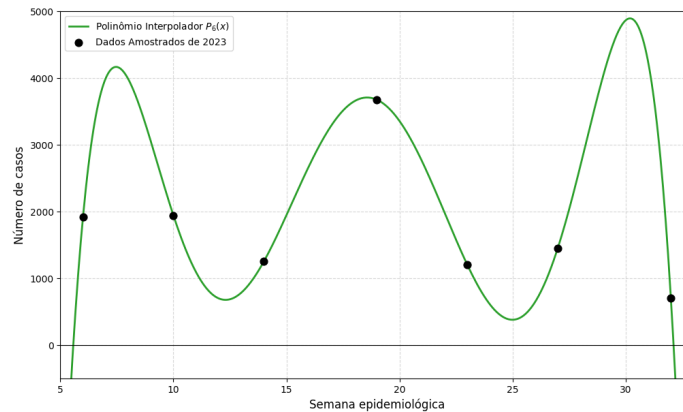
Fonte: Elaborada pelos autores com base nos dados do Ministério da Saúde [11].

O polinômio que modela os dados de 2023, $p_6^{2023}(x)$ refletindo a dinâmica da doença naquele ano, é expresso por:

$$p_6^{2023}(x) = -\frac{64835503}{5734048320}x^6 + \frac{4323343}{3407040}x^5 - \frac{64752203489}{1146809664}x^4 + \frac{3382737823}{2673216}x^3 - \frac{21413747555623}{1433512080}x^2 + \frac{3791973258253}{43439760}x - \frac{6117197247}{31603}.$$

O gráfico do polinômio interpolador para esse conjunto de dados é:

Figura 8 – Polinômio interpolador de Newton para a média móvel dos casos de COVID-19 em 2023.



Fonte: Elaborada pelos autores com base nos dados do Ministério da Saúde [11].

Para o ano de 2025, a tabela de diferenças divididas é apresentada a seguir:

Tabela 11 – Tabela de diferenças divididas da média móvel de casos de COVID-19 — Ano de 2025.

x	Ordem 0	Ordem 1	Ordem 2	Ordem 3	Ordem 4	Ordem 5	Ordem 6
6	615	$-\frac{589}{4}$	$\frac{641}{36}$	$-\frac{17}{12}$	$\frac{1181}{15912}$	$-\frac{7859}{3938220}$	$\frac{1}{12046320}$
10	26	13	$-\frac{11}{18}$	$-\frac{145}{936}$	$\frac{42120}{1277}$	$-\frac{15701}{7876440}$	
15	91	$\frac{15}{2}$	$-\frac{21}{8}$	$\frac{1829}{4680}$	$-\frac{359}{26520}$		
19	121	$-\frac{27}{2}$	$\frac{221}{90}$	$\frac{94}{585}$			
23	67	$\frac{43}{5}$	$\frac{409}{90}$				
28	110	$\frac{99}{2}$					
32	308						

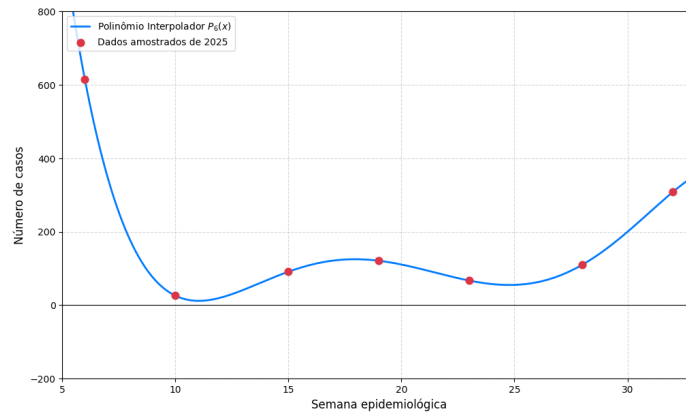
Fonte: Elaborada pelos autores com base nos dados do Ministério da Saúde [11].

Por fim, o polinômio para 2025, $p_6^{2025}(x)$ que representa o cenário de baixa incidência, é:

$$p_6^{2025}(x) = \frac{1}{12046320}x^6 - \frac{27359}{13652496}x^5 + \frac{9020329}{40957488}x^4 - \frac{4653811}{505648}x^3 + \frac{18610626889}{102393720}x^2 - \frac{1437328702}{853281}x + \frac{1677203452}{284427}.$$

A curva do polinômio interpolador para esse conjunto de dados é dado pelo gráfico:

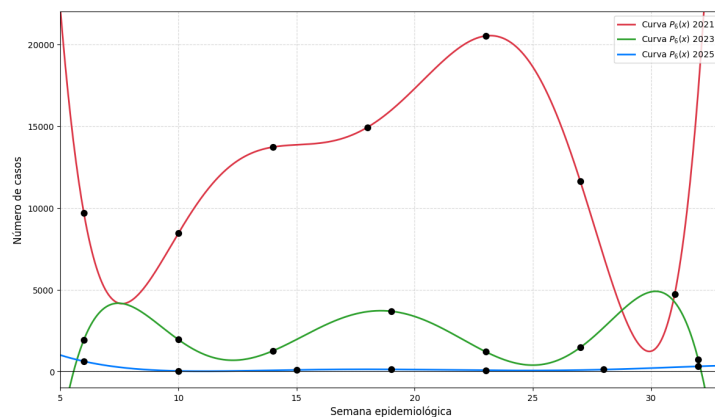
Figura 9 – Polinômio interpolador de Newton para a média móvel dos casos de COVID-19 em 2025.



Fonte: Elaborada pelos autores com base nos dados do Ministério da Saúde [11].

A visualização conjunta das três curvas polinomiais permite uma análise clara da evolução epidemiológica da COVID-19.

Figura 10 – Gráfico comparativo dos polinômios interpoladores de Newton para casos de COVID-19, nos anos de 2021, 2023 e 2025.



Fonte: Elaborada pelos autores com base nos dados do Ministério da Saúde [11].

A curva correspondente ao ano de 2021 apresenta a maior magnitude, com picos superiores a 20.000 casos, evidenciando um período marcado pela baixa cobertura vacinal e pela intensa circulação viral. O polinômio $p_6^{2021}(x)$ reproduz essa dinâmica, refletindo elevada variabilidade nos registros epidemiológicos. Em contraste, a curva de 2023 mostra uma redução substancial no número de casos, com valores máximos em torno de 4.000 e um comportamento mais suavizado, resultado direto do avanço da vacinação e do aumento da imunidade populacional. No ano de 2025, observa-se um patamar significativamente inferior, com casos raramente ultrapassando 600, caracterizando um cenário de estabilização em que a COVID-19 passa a apresentar comportamento endêmico, com oscilações de baixa expressividade estatística.

A interpolação polinomial de Newton, embora reproduza exatamente os pontos observados e forneça uma representação contínua da tendência, apresenta limitações importantes quando aplicada a séries epidemiológicas, especialmente com polinômios de alto grau como $p_6(x)$. Esses polinômios são suscetíveis ao Fenômeno de Runge, o que gera oscilações exageradas e artificiais entre os pontos (visíveis na curva de 2021) e torna a extrapolação fora do intervalo de dados pouco confiável, visto que a curva tende a divergir rapidamente. Além disso, o método é essencialmente descritivo, pois não incorpora mecanismos epidemiológicos causais — como taxas de infecção ou recuperação — ao contrário de modelos mecanísticos como o SIR, que se baseiam na dinâmica biológica da doença [13].

Portanto, a aplicação do Método de Newton permitiu reconstruir e comparar de forma clara a evolução dos diferentes cenários epidemiológicos nos anos analisados. Contudo, seus resultados devem ser interpretados com a devida cautela e dentro das limitações inerentes à interpolação polinomial. Seu principal papel neste trabalho é ilustrar a técnica de análise numérica e a visualização de tendências, mas não o de fornecer um modelo preditivo robusto ou substituir modelos epidemiológicos mais adequados para a elaboração de políticas de saúde pública.

Vale ressaltar que a integração entre a Matemática do Ensino Médio e a análise de fenômenos reais, como a dinâmica das epidemias, confere à construção de modelos matemáticos um elevado valor pedagógico. Ao trabalhar com dados concretos e métodos numéricos — como o Método de Newton e a média móvel — o estudante deixa de atuar apenas como receptor de conteúdo e passa a assumir um papel ativo na investigação e na construção do conhecimento, superando a fragmentação tradicional do currículo. Tal perspectiva dialoga diretamente com a concepção freireana de aprendizagem, conforme destacado a seguir.

Pelo contrário, nas condições de verdadeira aprendizagem os educandos vão se transformando em reais sujeitos da construção e da reconstrução do saber ensinado, ao lado do educador, igualmente sujeitos ao processo [14, p. 26].

Essa abordagem, portanto, constitui uma prática didática altamente motivadora, capaz de promover não apenas a compreensão de conceitos matemáticos, mas também o desenvolvimento crítico de professores e alunos ao relacionar o conhecimento científico às questões contemporâneas.

3 Análise do erro de interpolação

A essência do nosso estudo em métodos de interpolação não reside apenas em encontrar um polinômio $p_n(x)$ que satisfaça os pontos dados de uma função $f(x)$, como fizemos com os polinômios de Lagrange ou de Newton, mas sim em entender quão boa é essa aproximação.

Neste capítulo, apresentaremos o Teorema do Erro de Interpolação, que é a chave para essa garantia. Este teorema fundamental nos fornecerá um limitante superior, uma cota, para a magnitude do erro $E(x) = f(x) - p_n(x)$. Ao conhecermos este limitante, podemos afirmar, com a necessária confiança matemática, que a aproximação se encontra dentro de uma margem aceitável de incerteza.

Além disso, observamos que o estudo da fórmula do erro não é um fim em si. Ele ocupa um lugar de destaque para a otimização do processo de aproximação. A fórmula do erro revela que a precisão é influenciada pelo comportamento da função original, ou seja, por suas derivadas. E, crucialmente, pela distribuição dos nós de interpolação. Nota-se que ao entendermos esses fatores, podemos tomar decisões informadas para minimizar o erro global. Isso inclui a escolha de melhores métodos, como utilizar *splines* cúbicas em vez de um único polinômio de grau elevado para controlar oscilações ou a seleção de melhores nós de interpolação, como os pontos de Chebyshev, para combater a divergência observada em nós uniformemente espaçados.

O objetivo deste capítulo é, portanto, abordar as ferramentas necessárias para não apenas interpolar, mas fundamentalmente para quantificar, controlar e, principalmente, minimizar o erro inerente a qualquer processo de aproximação polinomial.

3.1 Erro de interpolação

A aproximação de uma função f por um polinômio interpolador p_n , de grau menor ou igual a n , está naturalmente associada a um erro de aproximação.

Definição 3.1.1 (Erro de Interpolação). Seja $f: [x_0, x_n] \rightarrow \mathbb{R}$ uma função real e seja $p_n(x)$ o seu polinômio interpolador de grau menor ou igual a n , construído a partir dos

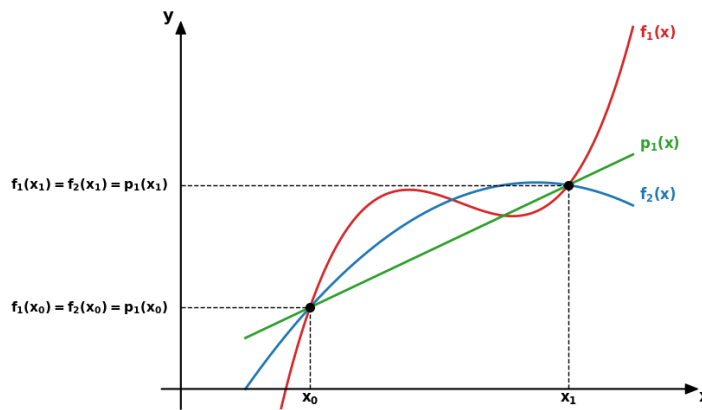
nós $\{x_i\}_{i=0}^n$. O erro de interpolação, denotado por $E_n(x)$, é definido por

$$E_n(x) = f(x) - p_n(x), \quad \text{para todo } x \in [x_0, x_n]. \quad (3.1)$$

O exemplo a seguir evidencia que a magnitude do erro de interpolação não depende exclusivamente do grau n do polinômio interpolador, mas também das características da função original $f(x)$.

Exemplo 3.1.1. Considere duas funções distintas, $f_1(x)$ e $f_2(x)$, interpoladas pelo mesmo polinômio linear $p_1(x)$ nos nós x_0 e x_1 , conforme ilustrado na Figura 11.

Figura 11 – Interpolação das funções $f_1(x)$ e $f_2(x)$ pelo polinômio linear $p_1(x)$.



Fonte: Adaptada de [5].

O erro de interpolação associado a cada função é dado, respectivamente, por $E_1^{(1)}(x) = f_1(x) - p_1(x)$ e $E_1^{(2)}(x) = f_2(x) - p_1(x)$. Observa-se, pela representação gráfica, que $|E_1^{(1)}(x)| > |E_1^{(2)}(x)|$ ao longo do intervalo considerado. Esse comportamento indica que o erro de interpolação não depende apenas do grau do polinômio interpolador, mas está diretamente relacionado à curvatura das funções originais, isto é, ao comportamento de suas derivadas de ordem superior, em particular da segunda derivada $f''(x)$.

Proposição 3.1.1. Seja f uma função real definida no intervalo $[a, b]$, e sejam x_0, x_1, \dots, x_n pontos distintos em $[a, b]$. Seja $p_n(x)$ o polinômio interpolador de f nesses pontos. Então, para todo $x \in [a, b]$, o erro de interpolação satisfaz:

$$E_n(x) = f[x_0, x_1, \dots, x_n, x] \prod_{i=0}^n (x - x_i). \quad (3.2)$$

Demonstração. Desejamos obter a expressão do erro de interpolação $E_n(x) = f(x) - p_n(x)$ utilizando a estrutura da Forma de Newton.

Seja $x \in [a, b]$ um ponto arbitrário, distinto dos nós x_0, x_1, \dots, x_n . Podemos estender o polinômio $p_n(t)$ ao incluir x como um novo nó de interpolação, definindo assim um

polinômio $p_{n+1}(t)$ de grau no máximo $n + 1$. Pela forma de Newton, esta extensão é expressa por:

$$p_{n+1}(t) = p_n(t) + f[x_0, x_1, \dots, x_n, x] \prod_{i=0}^n (t - x_i).$$

Visto que $p_{n+1}(t)$ é construído para interpolar f também no ponto x , temos por definição que $p_{n+1}(x) = f(x)$. Ao avaliarmos a identidade acima especificamente para $t = x$, obtemos:

$$f(x) = p_n(x) + f[x_0, x_1, \dots, x_n, x] \prod_{i=0}^n (x - x_i).$$

O erro de interpolação $E_n(x)$ é definido como a diferença entre o valor real da função e o valor aproximado pelo polinômio de grau n . Assim, subtraindo $p_n(x)$ de ambos os lados da igualdade anterior, resulta:

$$E_n(x) = f(x) - p_n(x) = f[x_0, x_1, \dots, x_n, x] \prod_{i=0}^n (x - x_i). \quad (3.3)$$

□

Essa expressão fornece uma forma unificada para o erro de interpolação associado a polinômios interpoladores de grau no máximo n , uma vez que, para valores específicos de n , obtêm-se os seguintes casos particulares:

$$\begin{aligned} E_0(x) &= (x - x_0) f[x_0, x], \\ E_1(x) &= (x - x_0)(x - x_1) f[x_0, x_1, x], \\ E_2(x) &= (x - x_0)(x - x_1)(x - x_2) f[x_0, x_1, x_2, x], \\ &\vdots \\ E_n(x) &= \prod_{i=0}^n (x - x_i) f[x_0, x_1, \dots, x_n, x]. \end{aligned}$$

A Fórmula (3.3) é poderosa por expressar o erro em termos de diferenças divididas. Contudo, para estimar sua magnitude de forma prática, é necessário o resultado do Teorema do Erro. A relação entre a diferença dividida $f[x_0, x_1, \dots, x_n, x]$ e a $(n + 1)$ -ésima derivada da função será o foco da análise a seguir. Para isso, introduziremos dois lemas auxiliares, o Teorema de Rolle e o Teorema de Rolle Generalizado, cujas demonstrações podem ser encontradas em [3].

Lema 1 (Teorema de Rolle). Seja $f \in C[a, b]$ e derivável em (a, b) . Se $f(a) = f(b) = 0$, então existe um número $\xi \in (a, b)$ tal que $f'(\xi) = 0$.

Lema 2 (Teorema de Rolle Generalizado). Seja $f \in C[a, b]$ e n vezes derivável em (a, b) . Se $f(x)$ possui $n + 1$ raízes distintas x_0, x_1, \dots, x_n no intervalo $[a, b]$, então existe um número $\xi \in (a, b)$ tal que $f^{(n)}(\xi) = 0$.

Veamos a seguir o teorema que nos dá a expressão do erro.

Teorema 3.1.1 (Teorema do Erro). Seja $\{x_i\}_{i=0}^n$ um conjunto de $n + 1$ pontos distintos em um intervalo $[a, b]$ e $f \in C^{n+1}[a, b]$. Para cada $x \in [a, b]$, existe um número $\xi(x)$ entre $\min\{x_0, x_1, \dots, x_n\}$ e $\max\{x_0, x_1, \dots, x_n\}$ e conseqüentemente em (a, b) , tal que o erro de interpolação é dado por:

$$E_n(x) = f(x) - p_n(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \prod_{i=0}^n (x - x_i) \quad (3.4)$$

onde $p_n(x)$ é o polinômio interpolador dado na equação (2.9).

Demonstração. A demonstração será realizada para qualquer $x \in [x_0, x_n]$ tal que $x \neq x_i$ para todo $i = 0, 1, \dots, n$. Note que se x coincidir com algum nó x_i , o erro é igual a zero, e a expressão do teorema também será igual a zero.

Primeiramente, definimos o polinômio que contém os termos nodais:

$$g(t) = \prod_{i=0}^n (t - x_i) = (t - x_0)(t - x_1) \cdots (t - x_n). \quad (3.5)$$

Para o ponto fixo x , definimos a função auxiliar $\phi(t)$ da seguinte forma:

$$\phi(t) = E_n(x)g(t) - E_n(t)g(x). \quad (3.6)$$

Note que $E_n(x)$ e $g(x)$ são constantes em relação à variável t . A função $\phi(t)$ possui derivadas de ordem até $n + 1$ no intervalo (x_0, x_n) , pois:

1. A função erro $E_n(t) = f(t) - p_n(t)$ é de classe C^{n+1} . Por hipótese, $f(t) \in C^{n+1}$ e $p_n(t)$ é um polinômio de grau n .
2. A função $g(t) \in C^{n+1}$. Ela foi definida na Equação (3.5) e é um polinômio de grau $n + 1$.

Como $\phi(t)$ é uma combinação linear de $g(t)$ e $E_n(t)$ com coeficientes constantes (em relação a t), a propriedade de diferenciabilidade é preservada, garantindo $\phi(t) \in C^{n+1}$.

A seguir, verificaremos que $\phi(t)$ possui pelo menos $n + 2$ zeros no intervalo que contém $[x_0, x_n]$.

1. Para $t = x_i$, onde $i = 0, 1, \dots, n$: Como $p_n(t)$ interpola $f(t)$ em x_i , o erro $E_n(x_i) = f(x_i) - p_n(x_i)$ é zero. Além disso, $g(x_i) = 0$ por definição na Equação (3.5). Portanto,

$$\phi(x_i) = E_n(x) \cdot 0 - 0 \cdot g(x) = 0.$$

2. Para $t = x$: Substituindo $t = x$ na função auxiliar, temos:

$$\phi(x) = E_n(x)g(x) - E_n(x)g(x) = 0.$$

Assim, x_0, x_1, \dots, x_n, x são $n + 2$ zeros distintos de $\phi(t)$ no intervalo $[x_0, x_n]$.

A função auxiliar $\phi(t)$ possui $n + 2$ zeros ($n + 1$ nós x_i e o ponto x). Pelo Lema 1, a primeira derivada, $\phi'(t)$, deve ter pelo menos um zero em cada um dos $n + 1$ subintervalos formados por esses zeros, totalizando pelo menos $n + 1$ zeros distintos. Ao aplicarmos sucessivamente o Lema 2, a ordem da derivada com zeros garantidos é reduzida em uma unidade a cada passo:

- $\phi'(t)$ tem $n + 1$ zeros distintos.
- $\phi''(t)$ tem n zeros distintos.
- ...
- $\phi^{(n)}(t)$ tem 2 zeros distintos.
- $\phi^{(n+1)}(t)$ tem pelo menos um zero, que denotamos por $\xi(x) = \xi$, no intervalo aberto (a, b) .

Assim, o próximo passo é calcular a $(n + 1)$ -ésima derivada de $\phi(t)$, a qual é dada por:

$$\phi^{(n+1)}(t) = E_n(x)g^{(n+1)}(t) - g(x)E_n^{(n+1)}(t).$$

Analisando a derivada $\phi^{(n+1)}(t)$ em termos de seus componentes:

1. A função erro é definida como $E_n(t) = f(t) - p_n(t)$. Como $p_n(t)$ é um polinômio de grau n , sua $(n + 1)$ -ésima derivada é identicamente nula. Portanto:

$$E_n^{(n+1)}(t) = f^{(n+1)}(t) - 0 = f^{(n+1)}(t).$$

2. O termo $g(t) = \prod_{i=0}^n (t - x_i)$ é um polinômio de grau $n + 1$. Ao expandir $g(t)$, o termo de maior grau é t^{n+1} :

$$g(t) = t^{n+1} + a_n t^n + \dots + a_1 t + a_0.$$

Ao derivar $g(t)$ exatamente $n + 1$ vezes, todos os termos de grau inferior a $n + 1$ se anulam. A derivada do termo líder é:

$$g^{(n+1)}(t) = \frac{d^{n+1}}{dt^{n+1}}(t^{n+1}) = (n + 1) \cdot n \cdot (n - 1) \cdot \dots \cdot 1.$$

Portanto, sua $(n + 1)$ -ésima derivada é igual a:

$$g^{(n+1)}(t) = (n + 1)!.$$

Substituindo estas identidades na expressão de $\phi^{(n+1)}(t)$ e avaliando no zero ξ :

$$\phi^{(n+1)}(\xi) = E_n(x) \cdot (n+1)! - g(x) \cdot f^{(n+1)}(\xi) = 0.$$

Isolando $E_n(x)$, obtemos a fórmula do erro:

$$E_n(x) = \frac{g(x) \cdot f^{(n+1)}(\xi)}{(n+1)!}.$$

Substituindo $g(x) = \prod_{i=0}^n (x - x_i)$, a expressão final do erro é:

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

□

A expressão do erro na Equação (3.4) demonstra formalmente que a magnitude do erro $E_n(x)$ na interpolação está diretamente relacionada à $(n+1)$ -ésima derivada da função, $f^{(n+1)}(\xi)$. Este resultado generaliza o princípio intuitivo observado na Figura 11, confirmando que a precisão da aproximação depende da suavidade da função f no intervalo de interesse.

Exemplo 3.1.2. Consideremos a aproximação numérica do valor $\ln(3.7)$ por meio da interpolação linear. Para isso, selecionam-se dois nós de interpolação x_0 e x_1 da função $f(x) = \ln(x)$ que contenham o ponto de interesse no intervalo definido. Os valores tabelados correspondentes, utilizados na construção do polinômio interpolador de grau 1, são apresentados na Tabela 12.

Tabela 12 – Valores da função $\ln(x)$.

x	1	2	3	4
$\ln(x)$	0	0.6931	1.0986	1.3863

Fonte: [5].

Como 3.7 pertence ao intervalo $(3, 4)$, utilizaremos os nós $x_0 = 3$ e $x_1 = 4$ para a interpolação linear, $n = 1$. O polinômio interpolador de grau 1, $p_1(x)$, é dado pela Fórmula de Newton:

$$\begin{aligned} p_1(x) &= f(x_0) + (x - x_0)f[x_0, x_1] \\ &= \ln(3) + (x - 3) \left(\frac{\ln(4) - \ln(3)}{4 - 3} \right) \\ &= 1.0986 + (x - 3) \frac{1.3863 - 1.0986}{1} \\ &= 1.0986 + (x - 3)0.2877. \end{aligned}$$

Ao calcularmos a aproximação em $x = 3.7$:

$$p_1(3.7) = 1.0986 + (3.7 - 3) \cdot 0.2877 \approx 1.3000.$$

Considerando que o valor real, com maior precisão, é $\ln(3.7) \approx 1.3083$, o erro real cometido é:

$$\begin{aligned} E_1(3.7) &= \ln(3.7) - p_1(3.7) \\ &= 1.3083 - 1.3000 \\ &= 0.0083 = 8.3 \cdot 10^{-3}. \end{aligned}$$

Pelo Teorema 3.1.1, podemos afirmar que para $n = 1$:

$$E_1(x) = (x - x_0)(x - x_1) \frac{f''(\xi)}{2!}, \quad \text{onde } \xi \in (x_0, x_1). \quad (3.7)$$

Para $f(x) = \ln(x)$, a segunda derivada é $f''(x) = \frac{d^2}{dx^2}(\ln x) = -\frac{1}{x^2}$. Substituindo $x = 3.7$, $x_0 = 3$, $x_1 = 4$ e o erro calculado $E_1(3.7) = 0.0083$, da Equação (3.7) temos:

$$0.0083 = (3.7 - 3)(3.7 - 4) \left(\frac{-1}{2\xi^2} \right).$$

Simplificando a expressão:

$$0.0083 = (0.7)(-0.3) \left(\frac{-1}{2\xi^2} \right) \implies 0.0083 = 0.21 \left(\frac{1}{2\xi^2} \right).$$

Isolando ξ :

$$2\xi^2 = \frac{0.21}{0.0083} \approx 25.3012 \implies \xi^2 \approx 12.6506$$

Consequentemente, $\xi = \sqrt{12.6506} \approx 3.5568$.

Como $\xi \approx 3.5568$, concluímos que ξ está contido no intervalo dos nós $(3, 4)$, validando a forma do erro conforme estabelecido no Teorema 3.1.1.

Vale ressaltar que a importância de encontrar o valor de ξ reside apenas como um artifício teórico para garantir a forma da expressão do erro, e não como um procedimento prático para seu cálculo.

O Teorema do Erro mostra que a magnitude do erro de interpolação está ligada à derivada de ordem superior da função. O teorema seguinte estabelece a relação entre a diferença dividida de ordem $n + 1$ a $(n + 1)$ -ésima derivada da função $f(x)$.

Teorema 3.1.2. Seja f uma função que possui $n + 1$ derivadas contínuas no intervalo $[a, b]$, e sejam x_0, x_1, \dots, x_n pontos distintos em $[a, b]$. Então, para qualquer x no intervalo

aberto delimitado pelos nós, isto é, $x \in (x_0, x_n)$, existe um ponto ξ pertencente ao menor intervalo aberto que contém x_0, x_1, \dots, x_n e x , tal que a diferença dividida de ordem $n + 1$ satisfaz:

$$f[x_0, x_1, \dots, x_n, x] = \frac{f^{(n+1)}(\xi)}{(n+1)!}. \quad (3.8)$$

Demonstração. Seja $p_n(x)$ o polinômio interpolador de grau no máximo n associado à função f , construído a partir dos nós distintos x_0, x_1, \dots, x_n . O erro de interpolação, definido pela Equação (3.1), admite duas representações equivalentes, as quais serão apresentadas a seguir.

Pelo Teorema 3.1.1, o erro pode ser expresso em termos da derivada de ordem $n + 1$:

$$E_n(x) = \prod_{i=0}^n (x - x_i) \frac{f^{(n+1)}(\xi)}{(n+1)!}, \quad \xi \in (a, b). \quad (3.9)$$

Alternativamente, a partir da construção do Polinômio de Newton, o erro é o termo de ordem $n + 1$ que seria adicionado para incluir o ponto x :

$$E_n(x) = \prod_{i=0}^n (x - x_i) f[x_0, x_1, \dots, x_n, x]. \quad (3.10)$$

Como ambas as expressões representam o mesmo erro $E_n(x)$, e assumindo $x \neq x_i$ para que o produto seja não nulo, podemos igualar as Equações 3.9 e 3.10:

$$\prod_{i=0}^n (x - x_i) \frac{f^{(n+1)}(\xi)}{(n+1)!} = \prod_{i=0}^n (x - x_i) f[x_0, x_1, \dots, x_n, x].$$

Dividindo ambos os lados pelo termo $\prod_{i=0}^n (x - x_i)$, obtemos a relação desejada:

$$f[x_0, x_1, \dots, x_n, x] = \frac{f^{(n+1)}(\xi)}{(n+1)!}.$$

□

Esse resultado fornece um meio de estimar o comportamento da derivada de ordem $(n + 1)$ de uma função f (quando existente), a partir das diferenças divididas de ordem $(n + 1)$ dessa mesma função no intervalo $[a, b]$ [15].

3.2 Limitante superior para o erro de interpolação

A fórmula do erro na Interpolação Polinomial, proveniente do Teorema 3.1.1, é dada por:

$$E_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^n (x - x_i), \quad (3.11)$$

onde ξ pertence ao menor intervalo aberto contendo os nós x_0, x_1, \dots, x_n e o ponto x .

Segundo Ruggiero e Lopes [5], essa expressão tem caráter predominantemente teórico, pois, na prática, raramente se conhece a derivada de ordem $(n+1)$ da função f em todo o intervalo, além de ser impossível determinar com exatidão o ponto ξ sem o conhecimento do erro real. Ainda assim, apesar dessa limitação, a forma exata do erro $E_n(x)$ possui grande relevância teórica, servindo como fundamento para a dedução de estimativas de erro em diversas técnicas da Análise Numérica, como na diferenciação e na integração numérica.

A seguir, será estabelecido um limitante superior para o módulo do erro de interpolação, $|E_n(x)|$.

Ruggiero e Lopes [5] apresentam dois corolários decorrentes do Teorema 3.1.1, os quais fornecem tais estimativas a partir de um limitante superior para o módulo da derivada de ordem $(n+1)$ da função f no intervalo considerado.

Corolário 3.2.1 (Limitante do erro de interpolação). Seja f uma função cuja derivada de ordem $(n+1)$, $f^{(n+1)}(x)$, seja contínua no intervalo I , e seja $p_n(x)$ o polinômio interpolador de f nos pontos distintos $x_0, x_1, \dots, x_n \in I$. Então, o erro de interpolação satisfaz a desigualdade:

$$|E_n(x)| = |f(x) - p_n(x)| \leq \frac{M_{n+1}}{(n+1)!} \left| \prod_{i=0}^n (x - x_i) \right|, \quad (3.12)$$

onde $M_{n+1} = \max_{x \in I} |f^{(n+1)}(x)|$. Define-se a função:

$$\mathcal{L}_n(x) = \frac{M_{n+1}}{(n+1)!} \left| \prod_{i=0}^n (x - x_i) \right| \quad (3.13)$$

como um limitante superior para o erro de interpolação.

Demonstração. O erro associado à aproximação da função $f(x)$ pelo polinômio interpolador $p_n(x)$ é caracterizado pelo Teorema 3.1.1. Em particular, o erro $E_n(x)$ pode ser expresso na forma:

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

onde ξ pertence ao menor intervalo aberto que contém os nós x_0, \dots, x_n e o ponto x .

Como, por hipótese, $f^{(n+1)}(x)$ é contínua no intervalo I , pelo Teorema do Valor Extremo, o módulo $|f^{(n+1)}(x)|$ atinge seu valor máximo no intervalo. Este valor é definido como M_{n+1} :

$$M_{n+1} = \max_{x \in I} |f^{(n+1)}(x)|.$$

Como o ponto ξ deve estar contido no intervalo I , a desigualdade $|f^{(n+1)}(\xi)| \leq M_{n+1}$ é válida.

Tomando o módulo da expressão do erro e aplicando a propriedade multiplicativa de módulos, seguida pela desigualdade acima, obtemos:

$$\begin{aligned} |E_n(x)| &= \left| \frac{1}{(n+1)!} f^{(n+1)}(\xi) \prod_{i=0}^n (x - x_i) \right| \\ &= \frac{1}{(n+1)!} |f^{(n+1)}(\xi)| \left| \prod_{i=0}^n (x - x_i) \right| \\ &\leq \frac{M_{n+1}}{(n+1)!} \left| \prod_{i=0}^n (x - x_i) \right|. \end{aligned}$$

□

O Corolário 3.2.2 apresenta uma estimativa mais concreta para o erro quando os nós de interpolação estão uniformemente distribuídos no intervalo, simplificando a Equação 3.12.

Corolário 3.2.2. Seja f uma função cuja derivada de ordem $(n+1)$, $f^{(n+1)}(x)$, seja contínua no intervalo $[a, b]$, e seja $p_n(x)$ o polinômio interpolador de f nos pontos distintos x_0, x_1, \dots, x_n pertencentes a $[a, b]$. Suponha que os nós de interpolação sejam igualmente espaçados, isto é, dados x_0 e $h > 0$, tem-se $x_{i+1} = x_i + h$, com $i = 0, \dots, n-1$. Então, para todo $x \in [x_0, x_n]$, o erro de interpolação satisfaz a desigualdade:

$$|E_n(x)| \leq \frac{h^{n+1}}{4(n+1)} M_{n+1}, \quad (3.14)$$

onde $M_{n+1} = \max_{x \in [a, b]} |f^{(n+1)}(x)|$.

Demonstração. Partimos do resultado obtido pelo Corolário do Limitante do Erro (3.2.1).

O objetivo é encontrar um limitante superior para o termo do polinômio nodal, $|\prod_{i=0}^n (x - x_i)|$, no intervalo de interpolação $[x_0, x_n]$.

Pela hipótese de que os nós estão uniformemente espaçados com passo h , ou seja, $x_{i+1} = x_i + h$, um resultado clássico da teoria de aproximação numérica [16] garante que o máximo valor absoluto desse produto é limitado por:

$$\max_{x \in [x_0, x_n]} \left| \prod_{i=0}^n (x - x_i) \right| \leq \frac{h^{n+1} n!}{4}. \quad (3.15)$$

Substituindo este limitante na expressão do Corolário do Limitante do Erro (3.2.1), temos:

$$|E_n(x)| \leq \frac{M_{n+1}}{(n+1)!} \cdot \left(\max_{x \in [x_0, x_n]} \left| \prod_{i=0}^n (x - x_i) \right| \right) \leq \frac{M_{n+1}}{(n+1)!} \cdot \left(\frac{h^{n+1} n!}{4} \right).$$

Utilizando a propriedade do fatorial $(n+1)! = (n+1) \cdot n!$, psimplificamos os fatoriais:

$$|E_n(x)| \leq \frac{M_{n+1}}{(n+1)n!} \cdot \frac{h^{n+1} n!}{4} = \frac{h^{n+1}}{4(n+1)} M_{n+1},$$

onde $M_{n+1} = \max_{x \in [a,b]} |f^{(n+1)}(x)|$.

□

Exemplo 3.2.1. Seja a função $f(x) = e^x + x - 1$ definida no intervalo $[0, 2]$, e os pontos de interpolação dados de acordo com a Tabela 13.

Tabela 13 – Valores da função $f(x) = e^x + x - 1$ em nós igualmente espaçados.

x	0	0,5	1	1.5	2
$f(x)$	0	1,1487	2,7183	4,9811	8,3890

Fonte: [5].

Iremos obter uma aproximação de $f(0.7)$ utilizando um polinômio interpolador linear ($n = 1$) e, em seguida, estimar o erro cometido com base no Corolário 3.2.1 e no Corolário 3.2.2.

Como o ponto a ser interpolado é $x = 0.7 \in [0.5, 1]$, escolheremos $x_0 = 0.5$ e $x_1 = 1$. Usando a forma de Newton para determinar o polinômio interpolador linear ($p_1(x)$), temos:

$$p_1(x) = f(x_0) + (x - x_0) \frac{f(x_1) - f(x_0)}{x_1 - x_0}.$$

Substituindo os valores:

$$\begin{aligned} p_1(x) &= f(0.5) + (x - 0.5) \frac{f(1) - f(0.5)}{1 - 0.5} \\ &= 1.1487 + (x - 0.5) \frac{2.7183 - 1.1487}{0.5} \\ &= 1.1487 + (x - 0.5) \cdot 3.1392. \end{aligned}$$

O valor da função interpoladora no ponto $x = 0.7$ é:

$$p_1(0.7) = 1.1487 + (0.7 - 0.5) \cdot 3.1392 = 1.1487 + 0.2 \cdot 3.1392 \approx 1.7765.$$

O erro da interpolação linear ($n = 1$) depende da segunda derivada $f''(x)$. A função é $f(x) = e^x + x - 1$, e sua segunda derivada é $f''(x) = e^x$. Como e^x é crescente, o valor máximo de $f''(x)$ no intervalo $I = [0.5, 1]$ é:

$$M_2 = \max_{x \in [0.5, 1]} |f''(x)| = e^1 \approx 2.7183.$$

Substituindo na expressão do majorante do Corolário 3.2.1:

$$\begin{aligned} |E_1(0.7)| &\leq \frac{M_2}{2!} |(0.7 - x_0)(0.7 - x_1)| \\ &\leq \frac{2.7183}{2} \cdot |(0.7 - 0.5)(0.7 - 1.0)| \\ &\leq 1.35915 \cdot |0.2 \cdot (-0.3)| \\ &\leq 1.35915 \cdot 0.06 \approx 0.0816. \end{aligned}$$

Observamos que os pontos de interpolação fornecidos estão igualmente espaçados com passo $h = 0.5$. Assim, podemos aplicar o Corolário 3.2.2 (para $n = 1$):

$$|E_1(x)| \leq \frac{h^{1+1}M_2}{4(1+1)} = \frac{h^2M_2}{8}.$$

Substituindo os valores $h = 0.5$ e $M_2 \approx 2.7183$:

$$|E_1(x)| \leq \frac{(0.5)^2 \cdot 2.7183}{8} = \frac{0.25 \cdot 2.7183}{8} \approx 0.0850.$$

O erro real da aproximação é (usando $f(0.7) \approx 1.71375$):

$$|E_1(0.7)| = |f(0.7) - p_1(0.7)| = |1.71375 - 1.7765| \approx 0.0628.$$

Constatamos que:

- $0.0628 < 0.0816$ (Limite do Corolário 3.2.1).
- $0.0628 < 0.0850$ (Limite do Corolário 3.2.2).

Ambas as estimativas confirmam que o erro real cometido é inferior ao limite teórico.

3.3 Estimativa do erro por meio de diferenças divididas

Embora os Corolários 3.2.1 e 3.2.2 forneçam estimativas úteis para o erro de interpolação, sua aplicação prática é limitada pela necessidade de conhecer o limitante

$$M_{n+1} = \max |f^{(n+1)}(x)|,$$

o que, em geral, não é viável. Conforme observam Ruggiero e Lopes [5], quando se dispõe apenas de um conjunto de dados tabelados, o valor exato do erro $|E_n(x)|$ não pode ser determinado, pois não é possível calcular diretamente a derivada de ordem $n+1$ da função f .

Nessa situação, uma alternativa consiste em utilizar a tabela de diferenças divididas de ordem $(n+1)$. O maior valor, em módulo, das diferenças divididas de ordem $(n+1)$ fornece uma aproximação para o fator $\frac{M_{n+1}}{(n+1)!}$ no intervalo $[x_0, x_n]$. Assim, obtemos a estimativa:

$$|E_n(x)| \approx \left| \prod_{i=0}^n (x - x_i) \right| \max |f[x_0, x_1, \dots, x_n, x]|. \quad (3.16)$$

Essa expressão possui grande relevância prática, pois o termo $f[x_0, x_1, \dots, x_n, x]$ atua como uma estimativa direta para $\frac{f^{(n+1)}(\xi)}{(n+1)!}$, dispensando o cálculo explícito da derivada

de ordem $n + 1$ da função f . Dessa forma, o uso das diferenças divididas revela-se um método eficiente e acessível para avaliar a precisão da interpolação quando apenas dados tabelados estão disponíveis.

Exemplo 3.3.1. Seja f uma função na qual os valores são tabelados na Tabela 14. Iremos obter uma aproximação para $f(0.47)$ por meio de um polinômio interpolador de grau 2 ($p_2(x)$) e, em seguida, estimar o erro $|E_2(0.47)|$ associado a essa aproximação utilizando a técnica de Diferenças Divididas.

Tabela 14 – Valores da função $f(x)$ utilizados na interpolação polinomial e na estimativa do erro.

x	0.2	0.34	0.4	0.52	0.6	0.72
$f(x)$	0.16	0.22	0.27	0.29	0.32	0.37

Fonte: [5].

Ao calcularmos as diferenças divididas até ordem 3 a partir dos valores tabelados de $f(x)$, com o auxílio da implementação em *python*, obtemos a Tabela 15:

Tabela 15 – Tabela de diferenças divididas para $p_2(x)$.

x	Ordem 0	Ordem 1	Ordem 2	Ordem 3
0.20	0.16			
		0.4286		
0.34	0.22		2.0235	
		0.8333		-17.8963
0.40	0.27		-3.7033	
		0.1667		18.2492
0.52	0.29		1.0415	
		0.3750		-2.6031
0.60	0.32		0.2085	
		0.4167		
0.72	0.37			

Fonte: [5].

Para aproximar $f(0.47)$, escolhemos os nós $x_0 = 0.40$, $x_1 = 0.52$ e $x_2 = 0.60$. O polinômio interpolador de grau 2 na forma de Newton é:

$$p_2(x) = f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1).$$

Substituindo os coeficientes destacados:

$$p_2(x) = 0.27 + 0.1667(x - 0.40) + 1.0415(x - 0.40)(x - 0.52).$$

Logo, o valor da aproximação em $x = 0.47$ é:

$$\begin{aligned} p_2(0.47) &= 0.27 + 0.1667(0.07) + 1.0415(0.07)(-0.05) \\ &\approx 0.27 + 0.011669 - 0.003645 \\ &\approx 0.2780. \end{aligned}$$

Portanto, $f(0.47) = p_2(0.47) \approx 0.2780$.

Por fim, utilizaremos a estimativa majorante da Equação (3.16) com a diferença dividida de ordem 3. O termo do produto é:

$$|(0.47 - 0.40)(0.47 - 0.52)(0.47 - 0.60)| = |0.07 \cdot (-0.05) \cdot (-0.13)| = 0.000455.$$

Assim, a partir do valor absoluto do produto e da maior diferença dividida de ordem 3 correspondente, ao substituirmos esses valores na fórmula para a estimativa do erro:

$$|E_2(0.47)| \approx 0.000455 \cdot 18.2492 \approx 0.008303.$$

Dessa forma, a estimativa para o erro é $|E_2(0.47)| \approx 8.3 \times 10^{-3}$.

4 Interpolação por Splines

Nos capítulos precedentes, exploramos as técnicas de Interpolação Polinomial, focando especificamente nos métodos de sistemas lineares, Lagrange e de Newton, visando determinar um polinômio que interpole uma função em um subconjunto finito de seu domínio.

Embora essas abordagens se revelem eficazes na aproximação da função em um intervalo fechado, especialmente quando o número de nós é reduzido e a função possui um comportamento relativamente regular, observamos uma limitação crucial: a introdução de um maior número de nós nem sempre resulta em uma aproximação superior. Percebemos que polinômios de grau elevado são suscetíveis a oscilações indesejadas, um fenômeno ilustrado, por exemplo, pelo fenômeno de Runge, que ocorre ao se considerar conjuntos arbitrários de pontos.

Diante dessa instabilidade e da necessidade de representar funções de forma mais suave e estável – particularmente ao lidar com um elevado número de dados ou com funções de comportamento complexo – recorreremos à interpolação por partes. Nessa técnica, o intervalo de interesse é particionado em subintervalos menores, e em cada um deles é construído um polinômio aproximador de grau baixo. Esta abordagem conduz ao estudo das *splines*.

Um *spline* de grau k é uma função S definida por partes, tal que $S \in \mathcal{C}^{k-1}$, onde cada subintervalo é um polinômio de grau menor ou igual a k . Em particular, um *spline* cúbico utiliza polinômios cúbicos entre cada par de nós sucessivos. A construção por *splines* cúbicos garante que a função interpoladora resultante pertença ao espaço das funções \mathcal{C}^2 (possuam segunda derivada contínua) sobre o intervalo de interesse, conferindo a desejada suavidade global.

Nos tópicos a seguir, discutiremos em detalhes o Fenômeno de Runge, introduziremos formalmente a interpolação linear por partes e, subsequentemente, a interpolação por *spline* cúbico.

4.1 O fenômeno de Runge

Adicionar mais nós a uma interpolação polinomial nem sempre resulta em uma aproximação mais precisa da função f dada. Embora seja intuitivo pensar que um polinômio de grau elevado se ajuste melhor aos pontos da função, o aumento do grau tende a gerar oscilações relevantes. Essas divergências ocorrem principalmente nas extremidades do intervalo, fenômeno conhecido como fenômeno de Runge, o qual evidencia os limites da interpolação polinomial de alto grau.

Para ilustrar o fenômeno de Runge, escolhemos o exemplo a seguir. Ele mostra como polinômios de alto grau, com nós igualmente espaçados, podem gerar oscilações indesejadas nas extremidades do intervalo.

Exemplo 4.1.1. Considere o problema de interpolar a função

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

no intervalo $[-1, 1]$ utilizando 11 pontos igualmente espaçados, dados por $x_i = -1 + hi$, com $i = 0, 1, \dots, 10$ e $h = 0.2$. Dessa forma, os valores de $f(x)$ tabelados no intervalo $[-1, 1]$ são:

Tabela 16 – Valores tabelados de $f(x)$ no intervalo $[-1, 1]$.

x	-1	-0.8	-0.6	-0.4	-0.2	-0	0.2	0.4	0.6	0.8	1
$f(x)$	0.03846	0.05882	0.1	0.2	0.5	1	0.5	0.2	0.1	0.05882	0.03846

Fonte: Elaborada com base em [6].

A tabela a seguir apresenta os resultados do cálculo das diferenças divididas, obtidos por meio de uma implementação do algoritmo em *python* para a interpolação polinomial de Newton:

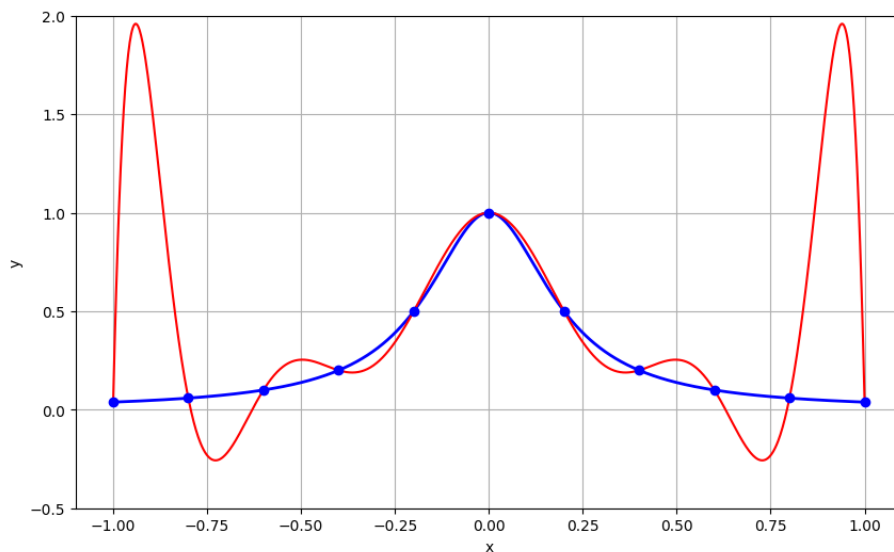
Tabela 17 – Tabela de diferenças divididas para $f(x)$ no intervalo $[-1, 1]$.

x	Ordem 0	Ordem 1	Ordem 2	Ordem 3	Ordem 4	Ordem 5	Ordem 6	Ordem 7	Ordem 8	Ordem 9	Ordem 10
-1.0	0.03846										
-0.8		0.1018									
-0.6			0.2602								
-0.4				0.7919							
-0.2					2.687						
0.0						-6.363					
0.2							-17.68				
0.4								84.84			
0.6									-167.9		
0.8										220.9	
1.0											-220.9

Fonte: Elaborada com base em [6].

A figura 12 ilustra a função $f(x)$ construída com base em 11 pontos nodais (ou dados) distribuídos no intervalo de interesse. Na figura, a função original, $f(x)$, é representada pela curva azul, e o polinômio interpolador pela curva vermelha. Observa-se que, nos extremos do intervalo de interpolação, o ajuste polinomial apresenta uma acentuada divergência (ou erro de interpolação) em relação à função $f(x)$. Essa divergência constitui uma clara manifestação do fenômeno de Runge, conforme previsto anteriormente.

Figura 12 – Gráficos da função $f(x)$ (em azul) e do seu polinômio interpolador de grau 10 (em vermelho) — Fenômeno de Runge.



Fonte: Elaborada com base em [6].

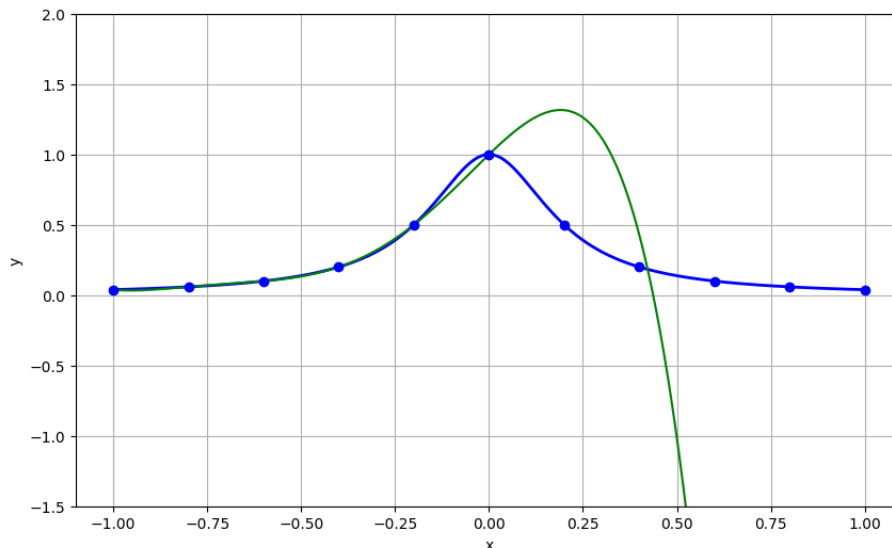
Neste estudo, verificou-se que a tabela de diferenças divididas, aliada à relação entre as diferenças divididas de ordem k e a derivada de ordem k da função, constitui um instrumento eficaz para identificar o grau apropriado do polinômio interpolador.

Com efeito, pelo Teorema 3.1.2, Neide Franco [15, p. 304] observa que, ao examinarmos uma tabela de diferenças divididas, se as diferenças de ordem k permanecem aproximadamente constantes, isso sugere que a função pode ser bem aproximada por um polinômio de grau k . Assim, torna-se adequado empregar um polinômio desse grau no processo de interpolação.

Essa análise permite determinar o ponto a partir do qual o acréscimo de novos termos deixa de produzir melhorias significativas na precisão da aproximação, evitando oscilações desnecessárias ou incremento do erro de interpolação.

Retomando o Exemplo 4.1.1, nota-se, por meio da tabela de diferenças divididas, que a partir do grau 5 os valores começam a oscilar de maneira expressiva. Esse comportamento indica que, nesse caso, polinômios de grau elevado não necessariamente aprimoram a aproximação e podem, inclusive, intensificar oscilações indesejadas. Dessa forma, um polinômio de grau moderado, como o de grau 5, revela-se uma escolha mais estável e adequada.

Figura 13 – Gráficos da função $f(x)$ (em azul) e do seu polinômio interpolador de grau 5 (em verde).



Fonte: Autores.

Embora o polinômio de grau 5 já reduza algumas oscilações em relação ao polinômio de grau 10, ainda existem pequenas irregularidades, especialmente próximas às extremidades do intervalo.

Para contornar estes problemas relativos ao fenômeno de Runge, algumas alternativas

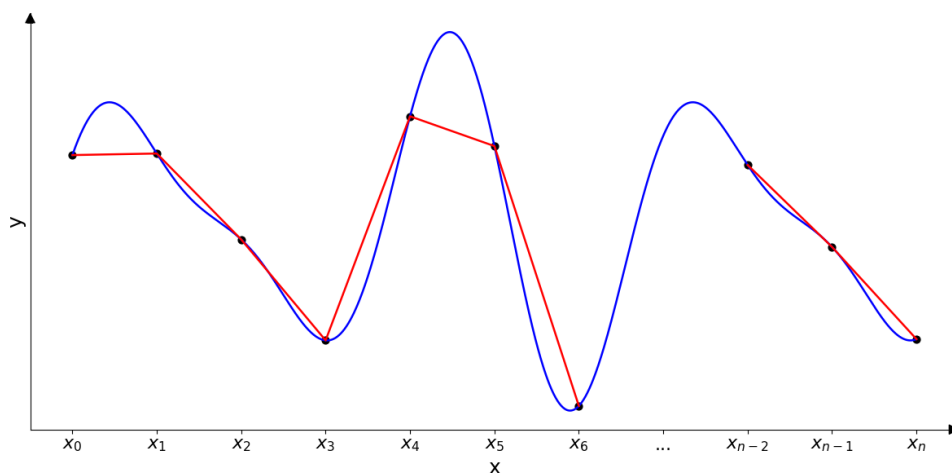
podem ser adotadas. Uma delas, como foi feito no exemplo 4.1.1, é recorrer a polinômios de menor grau por meio do truncamento da interpolação, utilizando apenas os primeiros termos da tabela de diferenças divididas. Essa estratégia busca manter uma aproximação local mais estável, reduzindo as oscilações que surgem em aproximações de ordem elevada. Outra possibilidade consiste no uso dos nós de Chebyshev, em que os pontos são distribuídos de forma não equidistante, e que contribui para amenizar o efeito de borda do fenômeno de Runge [6].

A seguir, apresentaremos a interpolação por *splines*, a qual se estabelece como uma metodologia robusta na aproximação de funções. O princípio fundamental do método é a imposição de condições de continuidade paramétrica nas conexões entre os segmentos (os nós), garantindo que a função resultante pertença a uma classe de suavidade C^k apropriada, geralmente C^2 . Esta abordagem elimina o problema das instabilidades numéricas e das oscilações espúrias, o fenômeno de Runge, característicos da interpolação polinomial de grau elevado, resultando em uma aproximação de elevada fidelidade e estabilidade numérica superior.

4.2 Interpolação por spline linear

Segundo Burden et al. [8], a forma mais simples de aproximação polinomial por partes é a interpolação linear por partes. Essa técnica, também conhecida como interpolação por *spline* linear, consiste em construir uma função interpoladora formada por segmentos de reta que conectam pares consecutivos de pontos de dados, $(x_0, f(x_0)), (x_1, f(x_1)), \dots, (x_n, f(x_n))$, conforme mostrado na Figura 14.

Figura 14 – Interpolação linear por partes.



Fonte: Autores.

De modo mais preciso, considere $\{(x_i, y_i)\}_{i=0}^n$ um conjunto formado por $n + 1$ pontos de dados, de modo que $x_{i+1} > x_i$, isto é, as abscissas são distintas e estão em ordem

estritamente crescente. Aproxima-se f por uma função contínua, *spline* linear, que em cada intervalo $I_i = [x_i, x_{i+1}]$ está definida pelo segmento de reta que une os dois pontos consecutivos $(x_i, f(x_i))$ e $(x_{i+1}, f(x_{i+1}))$. A função linear que interpola esses dois extremos no intervalo I_i é o polinômio interpolador de Lagrange de primeiro grau dado por:

$$P_i(x) = y_i \frac{x_{i+1} - x}{x_{i+1} - x_i} + y_{i+1} \frac{x - x_i}{x_{i+1} - x_i}, \forall x \in I_i. \quad (4.1)$$

Portanto, o resultado da interpolação linear é uma função contínua definida por partes em cada intervalo I_i :

$$S(x) = \begin{cases} P_0(x), & x \in [x_0, x_1] \\ P_1(x), & x \in [x_1, x_2] \\ \vdots \\ P_{n-1}(x), & x \in [x_{n-1}, x_n]. \end{cases} \quad (4.2)$$

ou ainda:

$$S(x) = P_i(x), \quad \text{para } x \in [x_i, x_{i+1}], \quad \text{com } i = 0, 1, \dots, n-1. \quad (4.3)$$

Optamos por não detalhar uma aplicação numérica para o caso da interpolação por spline de grau um, visto que sua simplicidade computacional é evidente. A construção do polinômio interpolador para cada subintervalo é reduzida a um cálculo local e explícito da reta secante.

Definição 4.2.1. Uma função f é continuamente diferenciável ou de classe C^1 quando ela é diferenciável e a sua derivada f' é contínua.

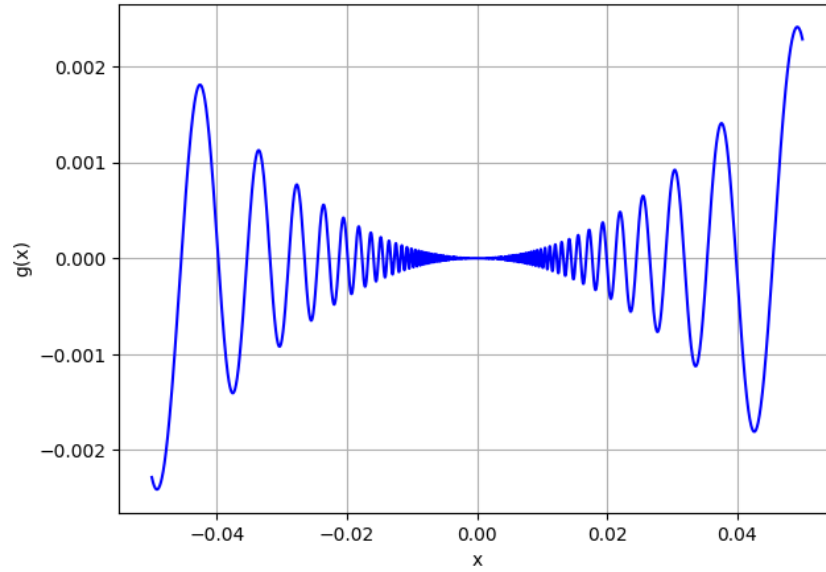
Mais claramente, se a função f é diferenciável em seu domínio, isso significa que a derivada f' existe em todos os pontos do domínio. Ou ainda, a diferenciabilidade de f garante que o gráfico da função não tem "bicos" (cantos, cúspides) ou tangentes verticais. Ela é suave no sentido de que a reta tangente é bem definida em todos os pontos. Além disso, se f' é função contínua, então a inclinação do gráfico varia de forma gradual, sem saltos abruptos.

Como este conceito é importante para o estudo sobre *spline* cúbico que veremos na próxima seção, a seguir apresentaremos uma função que é diferenciável e a sua derivada não é contínua. Isto é, uma função que não satisfaz a propriedade de ser continuamente diferenciável.

Exemplo 4.2.1. Consideremos a função $g : \mathbb{R} \rightarrow \mathbb{R}$ definida por:

$$g(x) = \begin{cases} x^2 \sin\left(\frac{1}{x}\right) & \text{se } x \neq 0 \\ 0 & \text{se } x = 0 \end{cases},$$

cujos gráfico, obtido através do *python*, tem a seguinte forma:

Figura 15 – Função $g(x)$ diferenciável com derivada descontínua em $x = 0$.

Fonte: Autores.

O objetivo é demonstrar que $g(x)$ é diferenciável em $x = 0$, mas sua derivada $g'(x)$ não é contínua em $x = 0$, o que implica que $g \notin C^1$.

Para verificar se $g(x)$ é diferenciável em $x = 0$, calculamos a derivada através da definição por limite:

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

Substituindo a definição de $g(x)$, temos que:

$$g'(0) = \lim_{h \rightarrow 0} \frac{h^2 \sin\left(\frac{1}{h}\right) - 0}{h} = \lim_{h \rightarrow 0} h \sin\left(\frac{1}{h}\right)$$

Pelo Teorema do Confronto (ou Sanduíche), uma vez que $-|h| \leq h \sin\left(\frac{1}{h}\right) \leq |h|$ e $\lim_{h \rightarrow 0} |h| = 0$, segue que $g'(0) = 0$. Portanto, a função $g(x)$ é diferenciável em $x = 0$.

Para verificarmos a continuidade de $g(x)$ em $x \neq 0$, calcularemos a derivada $g'(x)$, usando a regra do produto e a regra da cadeia.

$$\begin{aligned} g'(x) &= \frac{d}{dx} \left(x^2 \sin\left(\frac{1}{x}\right) \right) \\ g'(x) &= 2x \sin\left(\frac{1}{x}\right) + x^2 \left[\cos\left(\frac{1}{x}\right) \cdot \left(-\frac{1}{x^2}\right) \right] \\ g'(x) &= 2x \sin\left(\frac{1}{x}\right) - \cos\left(\frac{1}{x}\right). \end{aligned}$$

Para que $g'(x)$ seja contínua em $x = 0$, é necessário que $\lim_{x \rightarrow 0} g'(x) = g'(0) = 0$. Note que:

$$\lim_{x \rightarrow 0} g'(x) = \lim_{x \rightarrow 0} \left(2x \sin\left(\frac{1}{x}\right) - \cos\left(\frac{1}{x}\right) \right)$$

Analisando os termos de $g'(x)$, obtemos:

1. $\lim_{x \rightarrow 0} 2x \sin\left(\frac{1}{x}\right) = 0$ (pelo Teorema do Confronto).
2. $\lim_{x \rightarrow 0} \cos\left(\frac{1}{x}\right)$ não existe, pois o argumento $1/x$ tende ao infinito, fazendo o cosseno oscilar infinitamente entre -1 e 1 .

Logo, $\lim_{x \rightarrow 0} g'(x)$ não existe, ou seja, a derivada $g'(x)$ é descontínua em $x = 0$. Consequentemente, a função $g(x)$ é diferenciável, mas não é continuamente diferenciável.

Uma limitação da interpolação linear, destacada por Burden et al. [8], reside na potencial ausência de diferenciabilidade nos pontos de união dos subintervalos. Em diversas aplicações, entretanto, a continuidade da derivada da função aproximadora é um requisito fundamental. Por exemplo, em Engenharia Mecânica, na Robótica: a descontinuidade da primeira derivada, que representa a velocidade do braço do robô, significa que o braço teria que mudar de velocidade instantaneamente, ou acelerar infinitamente, em cada ponto de conexão para seguir a sua trajetória. Na prática, isso causaria solavancos, desgaste no motor e vibrações. Um sistema de controle real exige que a função de velocidade, a primeira derivada, seja, no mínimo, contínua e, idealmente, a aceleração, a segunda derivada, também seja contínua.

Um procedimento alternativo para a interpolação linear por partes consiste na utilização de polinômios por partes do tipo Hermite, conforme apresentado por Burden *et al.* [8]. Por exemplo, se os valores de f e de f' forem conhecidos em cada um dos pontos $x_0 < x_1 < \dots < x_n$, é possível empregar um polinômio cúbico de Hermite em cada subintervalo $[x_0, x_1], [x_1, x_2], \dots, [x_{n-1}, x_n]$, de modo a obter uma função interpoladora com derivada contínua no intervalo $[x_0, x_n]$ [6].

Contudo, para aplicar os polinômios por partes de Hermite para interpolação de dados é necessário conhecer previamente a sua derivada da função que está sendo aproximada, uma informação que, na prática, muitas vezes não está disponível. O que limita a aplicação direta desse método.

Em situações nas quais a diferenciabilidade e continuidade da derivada da interpolação é essencial, mas as derivadas da função a ser aproximada não são conhecidas, recorre-se as *splines* de ordem superior, tipicamente as *splines* cúbicas, que fornecem uma solução eficiente para a falta de suavidade presente na interpolação linear.

4.3 Interpolação por *spline* cúbico

A interpolação por *splines* cúbicos utiliza polinômios de grau três em cada intervalo formado por dois nós consecutivos. Esse método oferece flexibilidade suficiente para que

a função interpoladora não apenas seja continuamente diferenciável em todo o intervalo, mas também apresente continuidade em sua segunda derivada. Vale destacar que, na construção do *spline* cúbico, não se impõe que as derivadas da função interpoladora coincidam com as derivadas da função original aproximada, mesmo nos pontos de interpolação.

A seguir, apresentamos a definição geral de um *spline* de ordem p e, posteriormente, a definição para o *spline* cúbico.

Definição 4.3.1. (*Spline* de ordem p) Seja f uma função definida em um intervalo $\mathcal{I} = [a, b]$, em particular, num conjunto de nós $a = x_0 < x_1 < \dots < x_n = b$. Um *spline* de ordem p que interpola esses pontos é uma função S que satisfaz as seguintes condições:

- (i) $S_i(x)$ é um polinômio de grau no máximo p em cada intervalo $[x_i, x_{i+1}]$, para $i = 0, 1, 2, \dots, n - 1$;
- (ii) $S_i(x_i) = f(x_i)$ e $S_i(x_{i+1}) = f(x_{i+1})$, para $i = 0, 1, 2, \dots, n - 1$;
- (iii) $S_{i+1}(x_{i+1}) = S_i(x_{i+1})$, para $i = 0, 1, 2, \dots, n - 2$;
- (iv) $S_{i+1}^{(k)}(x_{i+1}) = S_i^{(k)}(x_{i+1})$ para $k = 1, 2, \dots, p - 1$ e $i = 0, 1, 2, \dots, n - 2$. Isto é, S é uma função $p - 1$ vezes continuamente diferenciável.

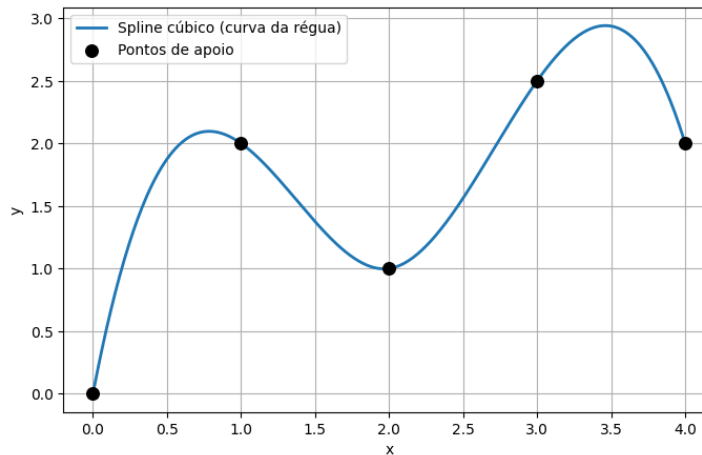
Há n intervalos para o conjunto de $n + 1$ pontos. Em cada um desses intervalos, um polinômio de grau p possui $p + 1$ coeficientes a serem determinados. Dessa forma, o número total de coeficientes a serem determinados é $n \cdot (p + 1)$.

Pela condição (ii) da Definição 4.3.1, a *spline* deve passar por cada um dos $n + 1$ pontos, pois se trata da condição de interpolação, gerando assim $n + 1$ equações. Por outro lado, a condição (iv) implica a continuidade das derivadas de ordem 0 até $p - 1$ nos pontos onde os segmentos de polinômios se encontram. A continuidade da própria função já é garantida pelas condições de interpolação nos pontos x_i . As condições de continuidade das derivadas aplicam-se nos pontos internos, ou seja, os pontos de \mathcal{I} com índices $i = 1, 2, \dots, n - 1$, que geram um total de $n - 1$ pontos internos. Em cada um desses $n - 1$ pontos internos, exigimos a continuidade de $p - 1$ derivadas, isto é, da primeira até a $(p - 1)$ -ésima derivada. Dessa forma, esta condição de suavidade gera $(p - 1) \cdot (n - 1)$ equações. Portanto, a partir do número de equações da condição de interpolação e as de suavidade das derivadas, obtemos um total de $(n + 1) + (p - 1) \cdot (n - 1)$ equações iniciais.

Para $p > 1$, ou seja, para polinômios de grau superior ao linear, o número de coeficientes a serem determinados é, em geral, maior do que o número de equações fornecidas por essas condições iniciais. Isso torna o problema subdeterminado, o que significa que existem infinitas splines de ordem p que satisfazem as condições básicas de interpolação e suavidade para o conjunto de pontos dados.

Em particular, o caso $p = 3$, denominado *spline* cúbico, apresenta grande relevância, pois sua formulação reproduz o comportamento físico de réguas delgadas com estrutura elástica homogênea e perfil uniforme, como apresentado na Figura 16, cujos vínculos são representados pelos pontos do conjunto do intervalo considerado. Esse comportamento é descrito por uma equação diferencial de quarta ordem, que constitui um caso particular da equação da viga de Euler-Bernoulli, cuja solução geral é um polinômio de grau três.

Figura 16 – *Spline* cúbico imitando o comportamento de uma régua delgada.



Fonte: Autor.

Definição 4.3.2. Seja f uma função definida em um intervalo $[a, b]$, em particular, num conjunto de nós $a = x_0 < x_1 < \dots < x_n = b$. Um *spline* cúbico interpolador S para f é uma função que satisfaz as seguintes condições:

- (i) $S_i(x)$ é um polinômio cúbico em cada subintervalo $[x_i, x_{i+1}]$, para $i = 0, 1, 2, \dots, n-1$;
- (ii) $S_i(x_i) = f(x_i)$ e $S_i(x_{i+1}) = f(x_{i+1})$, para $i = 0, 1, 2, \dots, n-1$;
- (iii) $S_{i+1}(x_{i+1}) = S_i(x_{i+1})$, para $i = 0, 1, 2, \dots, n-2$; (Implicado por (ii));
- (iv) $S'_{i+1}(x_{i+1}) = S'_i(x_{i+1})$, para $i = 0, 1, 2, \dots, n-2$;
- (v) $S''_{i+1}(x_{i+1}) = S''_i(x_{i+1})$, para $i = 0, 1, 2, \dots, n-2$.

Além disso, um dos seguintes conjuntos de condições de contorno deve ser satisfeito:

- (a) $S''(x_0) = S''(x_n) = 0$ (condições de contorno **livres** ou **naturais**);
- (b) $S'(x_0) = f'(x_0)$ e $S'(x_n) = f'(x_n)$ (condições de contorno **fixadas**).

O *spline* cúbico natural é obtido sob a imposição de condições de contorno livres, onde a segunda derivada nas extremidades é nula. A analogia física para esta configuração é

o formato que uma haste longa e flexível assumiria ao ser flexionada para passar pelos pontos de dados.

Apesar deste estudo se restringir à análise de *splines* cúbicos naturais, é importante notar que o emprego de condições de contorno fixadas, como os valores das derivadas nos extremos, geralmente proporciona aproximações superiores. Isso ocorre porque tais condições incorporam informações adicionais sobre a função, otimizando o ajuste.

Adiante, apresentaremos um exemplo que descreve o cálculo de um *spline* cúbico para três pontos dados, mostrando que a construção de um *spline* natural é um processo relativamente simples.

Exemplo 4.3.1. Vamos construir um *spline* cúbico natural que passa pelos pontos $(1, 2)$, $(2, 3)$, e $(3, 5)$.

Este *spline* é definido por dois polinômios cúbicos distintos, $S_0(x)$ e $S_1(x)$. O primeiro polinômio, $S_0(x)$, é válido no subintervalo $[1, 2]$ e é dado por:

$$S_0(x) = a_0 + b_0(x - 1) + c_0(x - 1)^2 + d_0(x - 1)^3. \quad (4.4)$$

O segundo polinômio, $S_1(x)$, é válido no subintervalo $[2, 3]$ e é dado por:

$$S_1(x) = a_1 + b_1(x - 2) + c_1(x - 2)^2 + d_1(x - 2)^3. \quad (4.5)$$

Para a determinação das oito constantes nestas equações, são necessárias oito condições. Quatro delas vêm do fato de que os splines devem coincidir com os dados nos nós, conforme a condição (ii) da Definição 4.3.2. Dessa forma, temos que:

$$\begin{aligned} S_0(1) = f(1) &\implies a_0 = 2, \\ S_0(2) = f(2) &\implies a_0 + b_0 + c_0 + d_0 = 3, \\ S_1(2) = f(2) &\implies a_1 = 3 \text{ e} \\ S_1(3) = f(3) &\implies a_1 + b_1 + c_1 + d_1 = 5. \end{aligned}$$

Outras duas condições decorrem do fato de que $S'_0(2) = S'_1(2)$ e $S''_0(2) = S''_1(2)$, correspondendo, respectivamente, às condições (iv) e (v) da Definição 4.3.2. Assim, obtemos:

$$\begin{aligned} S'_0(2) = S'_1(2) &\implies b_0 + 2c_0 + 3d_0 = b_1 \text{ e} \\ S''_0(2) = S''_1(2) &\implies 3c_0 + 6d_0 = 2c_1. \end{aligned}$$

Por fim, as duas últimas vêm das condições de contorno naturais:

$$\begin{aligned} S''_0(1) = 0 &\implies 2c_0 = 0 \text{ e} \\ S''_1(3) = 0 &\implies 2c_1 + 6d_1 = 0. \end{aligned}$$

Portanto, a partir das condições apresentadas, obtém-se três coeficientes já determinados, a saber: $a_0 = 2$, $a_1 = 3$ e $c_0 = 0$. Para determinar os coeficientes restantes, escreveremos o sistema de equações derivado das condições de continuidade e derivabilidade, o qual é dado por:

$$\begin{cases} b_0 + d_0 = 1 \\ b_1 + c_1 + d_1 = 2 \\ b_0 + 3d_0 = b_1 \\ 6d_0 = 2c_1 \\ 2c_1 + 6d_1 = 0 \end{cases} \implies \begin{cases} b_0 + 0b_1 + 0c_1 + d_0 + 0d_1 = 1 \\ 0b_1 + b_1 + c_1 + 0d_0 + d_1 = 2 \\ b_0 - b_1 + 0c_1 + 3d_0 + 0d_1 = 0 \\ 0b_0 + 0b_1 - 2c_1 + 6d_0 + 0d_1 = 0 \\ 0b_0 + 0b_1 + 2c_1 + 0d_0 + 6d_1 = 0 \end{cases}$$

Este sistema pode ser escrito em forma matricial $Ax = b$, o que permite uma manipulação mais direta usando álgebra linear:

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & -1 & 0 & 3 & 0 \\ 0 & 0 & -2 & 6 & 0 \\ 0 & 0 & 2 & 0 & 6 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ c_1 \\ d_0 \\ d_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Para a resolução deste sistema utilizamos o *python* como recurso computacional e obtemos os seguintes valores para os coeficientes:

$$b_0 = \frac{3}{4}, \quad b_1 = \frac{3}{2}, \quad c_1 = \frac{3}{4}, \quad d_0 = \frac{1}{4} \quad \text{e} \quad d_1 = -\frac{1}{4}.$$

Substituindo esses valores nas Equações (4.4) e (4.5), provenientes da solução do sistema associado às oito condições impostas, obtém-se finalmente o *spline*:

$$S(x) = \begin{cases} 2 + \frac{3}{4}(x-1) + \frac{1}{4}(x-1)^3, & \text{para } x \in [1, 2] \\ 3 + \frac{3}{2}(x-2) + \frac{3}{4}(x-2)^2 - \frac{1}{4}(x-2)^3, & \text{para } x \in [2, 3]. \end{cases}$$

Como ilustrado no Exemplo 4.3.1, um *spline* cúbico que interpola $n+1$ pontos consiste em n polinômios cúbicos. Cada polinômio tem 4 coeficientes, totalizando $4n$ coeficientes a serem determinados. Portanto, este problema exige um total de $4n$ condições independentes. Essas condições são necessárias para montar e resolver o sistema linear que define a solução. Percebe-se que $2n$ condições são fornecidas pela propriedade (ii) da Definição 4.3.2, a qual exige que, em cada um dos n subintervalos, o polinômio cúbico coincida com

os valores dos dados nos dois nós das extremidades, isto é:

$$\begin{aligned} S_0(x_0) &= f(x_0) , S_0(x_1) = f(x_1); \\ S_1(x_1) &= f(x_1) , S_1(x_2) = f(x_2); \\ S_2(x_2) &= f(x_2) , S_2(x_3) = f(x_3); \\ &\vdots \\ S_{n-1}(x_{n-1}) &= f(x_{n-1}) , S_{n-1}(x_n) = f(x_n). \end{aligned}$$

Além disso, cada uma das propriedades (iv) e (v) da Definição 4.3.2 se aplica nos $n - 1$ nós internos, o que nos dá um total de $2(n - 1) = 2n - 2$ condições:

$$\begin{aligned} S'_0(x_1) &= S'_1(x_1) , S''_0(x_1) = S''_1(x_1); \\ S'_1(x_2) &= S'_2(x_2) , S''_1(x_2) = S''_2(x_2); \\ S'_2(x_3) &= S'_3(x_3) , S''_2(x_3) = S''_3(x_3); \\ &\vdots \\ S'_{n-2}(x_{n-1}) &= S'_{n-1}(x_{n-1}) , S''_{n-2}(x_{n-1}) = S''_{n-1}(x_{n-1}). \end{aligned}$$

Portanto, as condições de interpolação e suavidade nos pontos de conexão fornecem $2n + 2(n - 1) = 4n - 2$ equações. Para um *spline* cúbico natural, as condições de contorno são definidas de modo que a segunda derivada nas extremidades do intervalo seja igual a zero, isto é:

$$S''_0(x_0) = S''_{n-1}(x_n) = 0.$$

Assim, temos mais duas condições, resultando em um total de $4n - 2 + 2 = 4n$. Portanto, há $4n$ coeficientes serem determinados e $4n$ condições, formando um sistema linear onde suas variáveis são os coeficientes dos polinômios cúbicos para cada um dos n subintervalos. Isto permite investigar a existência e unicidade da solução que resultará, na possível determinação do *spline* cúbico. Mas, antes iremos analisar o formato da matriz dos coeficientes deste sistema.

Note que a solução deste sistema de equações resultará no *spline*:

$$S(x) = \begin{cases} a_0 + b_0(x - x_0) + c_0(x - x_0)^2 + d_0(x - x_0)^3, & x \in [x_0, x_1] \\ a_1 + b_1(x - x_1) + c_1(x - x_1)^2 + d_1(x - x_1)^3, & x \in [x_1, x_2] \\ \vdots \\ a_{n-1} + b_{n-1}(x - x_{n-1}) + c_{n-1}(x - x_{n-1})^2 + d_{n-1}(x - x_{n-1})^3, & x \in [x_{n-1}, x_n]. \end{cases}$$

Consequentemente, para construir um *spline* cúbico interpolador S que passa por um conjunto de $n + 1$ pontos $\mathcal{I} = \{(x_i, y_i)\}_{i=0}^n$, onde $x_{i+1} > x_i$, definimos S por partes. Em

cada intervalo $[x_i, x_{i+1}]$, o *spline* S é um polinômio cúbico $S_i(x)$ da forma:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \quad (4.6)$$

para cada $i = 0, 1, \dots, n - 1$. Os coeficientes a_i, b_i, c_i, d_i são determinados pelas condições de interpolação e continuidade descritas na Definição 4.3.2. Como $S_i(x_i) = a_i = f(x_i)$, a condição (iii) pode ser aplicada para obter:

$$a_{i+1} = S_{i+1}(x_{i+1}) = S_i(x_{i+1}) = a_i + b_i(x_{i+1} - x_i) + c_i(x_{i+1} - x_i)^2 + d_i(x_{i+1} - x_i)^3,$$

para cada $i = 0, 1, \dots, n - 2$. Para termos de simplificação, adotemos a notação $h_i = x_{i+1} - x_i$, para cada $i = 0, 1, \dots, n - 1$. Além disso, tomarmos a condição $a_i = f(x_i)$, então a equação

$$a_{i+1} = a_i + b_i h_i + c_i h_i^2 + d_i h_i^3 \quad (4.7)$$

será válida para cada $i = 0, 1, \dots, n - 1$. Analogamente, ao tomarmos $b_i = S'(x_i)$ observe que

$$S'_i(x) = b_i + 2c_i(x - x_i) + 3d_i(x - x_i)^2$$

implica $S'_i(x_i) = b_i$, para cada $i = 0, 1, \dots, n - 1$. Ao aplicar a condição (iv), teremos

$$b_{i+1} = S'_{i+1}(x_{i+1}) = S'_i(x_{i+1}) \implies b_{i+1} = b_i + 2c_i h_i + 3d_i h_i^2 \quad (4.8)$$

para cada $i = 0, 1, \dots, n - 1$. Podemos obter uma outra relação entre os coeficientes de S_i ao observar que $c_i = S''_i(x_i)/2$ e aplicar a condição (v). Então, para cada $i = 0, 1, \dots, n - 1$:

$$c_{i+1} = \frac{S''_{i+1}(x_{i+1})}{2} = \frac{S''_i(x_{i+1})}{2} = \frac{2c_i + 6d_i h_i}{2} = c_i + 3d_i h_i. \quad (4.9)$$

Ao isolar d_i na Equação (4.9) e substituir esse valor nas Equações (4.7) e (4.8) iremos obter, para cada $i = 0, 1, \dots, n - 1$, as equações:

$$a_{i+1} = a_i + b_i h_i + \frac{h_i^2}{3}(2c_i + c_{i+1}) \quad (4.10)$$

e

$$b_{i+1} = b_i + h_i(c_i + c_{i+1}). \quad (4.11)$$

A relação final entre os coeficientes pode ser obtida isolando primeiro b_i na equação (4.10), o que resulta:

$$b_i = \frac{1}{h_i}(a_{i+1} - a_i) - \frac{h_i}{3}(2c_i + c_{i+1}), \quad (4.12)$$

e em seguida, reduzimos do índice b_i em um, o que nos dá:

$$b_{i-1} = \frac{1}{h_{i-1}}(a_i - a_{i-1}) - \frac{h_{i-1}}{3}(2c_{i-1} + c_i). \quad (4.13)$$

Substituindo os valores das Equações (4.12) e (4.13) na equação deduzida da equação (4.8), com o índice reduzido em um, resulta no sistema linear de equações

$$h_{i-1}c_{i-1} + 2(h_{i-1} + h_i) + h_i c_{i+1} = \frac{3}{h_i}(a_{i+1} - a_i) - \frac{3}{h_{i-1}}(a_i - a_{i-1}), \quad (4.14)$$

para cada $i = 1, 2, \dots, n - 1$.

Este sistema envolve apenas os coeficientes $\{c_i\}_{i=0}^n$ como incógnitas, pois os valores de $\{h_i\}_{i=0}^{n-1}$ são determinados, respectivamente, pelo espaçamento dos nós $\{x_i\}_{i=0}^n$ e pelos valores de f nesses nós. Uma vez determinados os valores de $\{c_i\}_{i=0}^n$, as constantes restantes podem ser obtidas de forma simples. Os coeficientes $\{b_i\}_{i=0}^{n-1}$ são determinados a partir da Equação (4.12) e $\{d_i\}_{i=0}^{n-1}$ pela Equação (4.9), o que permite, por fim, a construção dos polinômios cúbicos $\{S_i(x)\}_{i=0}^{n-1}$.

A principal questão que emerge a respeito desta construção é determinar se os valores dos coeficientes $\{c_i\}_{i=0}^n$ podem ser obtidos a partir do sistema de equações fornecido em (4.14) e, em caso afirmativo, se tais valores são únicos. Os teoremas subsequentes demonstram que esta condição é satisfeita quando qualquer uma das condições de contorno apresentadas na Definição 4.3.2 é imposta. Em particular, iremos mostrar que satisfaz para as condições de contorno naturais.

Teorema 4.3.1. Seja f uma função definida em um conjunto de $n + 1$ pontos $\lambda = \{(x_i, y_i)\}_{i=0}^n$, tais que $x_{i+1} > x_i$ para $i = 0, 1, \dots, n - 1$. Então, existe um único *spline* cúbico natural S que interpola f nos nós x_0, x_1, \dots, x_n , isto é, $S(x_i) = y_i$ para $i = 0, 1, \dots, n$, e que satisfaz as condições de contorno $S''(x_0) = S''(x_n) = 0$.

Demonstração. As condições de contorno impostas, nesse caso, implicam em:

$$c_n = S''(x_n)/2 = 0 \quad e \quad S''(x_0) = 2c_0 + 6d_0(x_0 - x_0) = 0.$$

Consequentemente, as duas equações, $c_0 = 0$ e $c_n = 0$, juntamente com as equações produzidas a partir da equação (4.14), produzem o sistema linear:

$$\left\{ \begin{array}{l} c_0 = 0 \\ h_0 c_0 + 2(h_0 + h_1)c_1 + h_1 c_2 = \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ h_1 c_1 + 2(h_1 + h_2)c_2 + h_2 c_3 = \frac{3}{h_2}(a_3 - a_2) - \frac{3}{h_1}(a_2 - a_1) \\ \vdots \\ h_{i-1} c_{i-1} + 2(h_{i-1} + h_i)c_i + h_i c_{i+1} = \frac{3}{h_i}(a_{i+1} - a_i) - \frac{3}{h_{i-1}}(a_i - a_{i-1}) \\ \vdots \\ h_{n-2} c_{n-2} + 2(h_{n-2} + h_{n-1})c_{n-1} + h_{n-1} c_n = \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ c_n = 0 \end{array} \right.$$

o qual pode ser descrito pela equação vetorial $A\mathbf{x} = b$, onde A é a matriz de ordem $(n+1) \times (n+1)$:

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ h_0 & 2(h_0 + h_1) & h_1 & 0 & \cdots & 0 & 0 \\ 0 & h_1 & 2(h_1 + h_2) & h_2 & \cdots & 0 & 0 \\ 0 & 0 & h_2 & 2(h_2 + h_3) & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 2(h_{n-2} + h_{n-1}) & h_{n-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix},$$

e b e \mathbf{x} são os vetores:

$$B = \begin{pmatrix} 0 \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \frac{3}{h_2}(a_3 - a_2) - \frac{3}{h_1}(a_2 - a_1) \\ \frac{3}{h_3}(a_4 - a_3) - \frac{3}{h_2}(a_3 - a_2) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 0 \end{pmatrix} \quad \text{e} \quad \mathbf{x} = \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_{n-1} \\ c_n \end{pmatrix}.$$

A formulação do sistema acima reduz o problema da existência e unicidade do *spline* à análise das propriedades da matriz A . Para garantir que o sistema possua uma solução única, é necessário assegurar que A seja não-singular. A seguir, apresentamos os conceitos de dominância diagonal que fornecerão a base teórica para essa verificação.

Definição 4.3.3. Seja $B = (b_{ij})$ uma matriz $n \times n$ com entradas complexas. B é dita diagonal dominante (ou diagonal dominante por linhas), para cada $i = 1, 2, \dots, n$, se:

$$|b_{ii}| \geq \sum_{j \neq i}^n |b_{ij}|.$$

Além disso, se tivermos $(>)$ ao invés de (\geq) , então B é dita estritamente dominante (por linhas).

A estrutura do sistema de equações derivado da construção dos *splines* cúbicos naturais herda essa propriedade, conforme enunciado a seguir.

Proposição 4.3.1. A matriz A dos coeficientes do sistema linear obtida na construção de um *spline* cúbico é estritamente diagonal dominante.

A demonstração detalhada desta proposição, que verifica a condição de dominância tanto para as linhas internas quanto para as condições de contorno, encontra-se no Apêndice A.

A relevância da Proposição 4.3.1 reside na sua conexão direta com a invertibilidade da matriz A . Essa garantia é fundamentada pelo Teorema de Lévy-Desplanques, enunciado a seguir, cuja demonstração detalhada pode ser consultada no Apêndice A.

Teorema 4.3.2 (Teorema de Lévy-Desplanques). Se uma matriz é estritamente diagonal dominante, então ela é não-singular.

Pelo fato de a matriz A satisfazer tal condição de dominância, o Teorema 4.3.2 assegura que seu determinante é não nulo. Consequentemente, o sistema linear possui uma solução única, o que estabelece, de forma definitiva, a unicidade do *spline* cúbico interpolador para o conjunto de dados fornecido.

□

4.4 Reconstrução de imagens digitais

A reconstrução digital e aproximação do contorno de uma folha foi realizada por meio de interpolação por *splines* cúbicos. Para gerar uma representação contínua e suave do contorno, foram primeiramente selecionados pontos discretos de amostragem escolhidos de forma estratégica, distribuídos ao longo de toda a borda superior da folha. A escolha do *spline* cúbico garante que a função interpoladora resultante não apenas passe exatamente por cada um desses pontos de discretização (propriedade de interpolação), mas também que possua continuidade de sua primeira e segunda derivadas nos nós (os pontos de amostragem). Essa condição garantirá a suavidade da curva reconstruída e é essencial para uma representação fiel de um contorno orgânico.

Os pontos utilizados na interpolação foram obtidos por meio de um procedimento composto por quatro etapas principais. Primeiramente, realizou-se o registro fotográfico da folha. Em seguida, a imagem foi editada através do site *remove.bg*, a fim de remover o plano de fundo e facilitar a identificação precisa da borda. Posteriormente, a fotografia

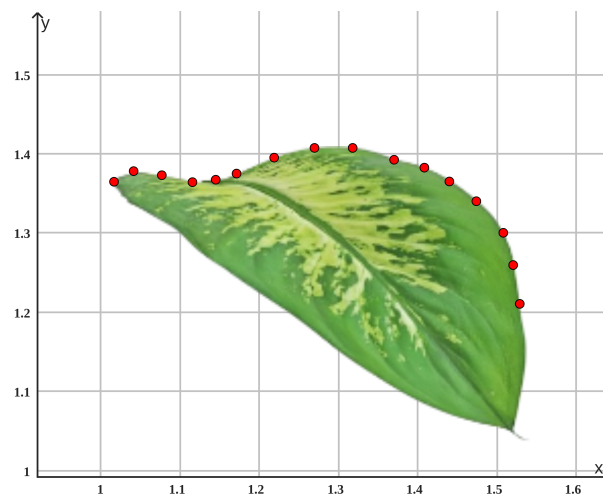
tratada foi importada para o *software GeoGebra*, no qual foram marcados manualmente 16 pontos distribuídos ao longo do contorno. Por fim, o arquivo foi exportado em formato *.ggb*, possibilitando o *download* e a extração das coordenadas dos pontos, que posteriormente foram utilizadas no ambiente *Python* para o cálculo do polinômio interpolador; suas coordenadas encontram-se na Tabela 18, enquanto a sua disposição gráfica sobre a folha é apresentada na Figura 17. Ressalta-se que foi empregada uma maior densidade de pontos nas regiões de maior curvatura, de modo a capturar adequadamente as variações geométricas locais, em conformidade com as recomendações presentes em Burden *et al.*[8].

Tabela 18 – Coordenadas dos pontos selecionados no *GeoGebra* para a reconstrução do contorno da folha por *splines* cúbicas.

x	y	x	y
1.01644	1.36468	1.26950	1.40738
1.04119	1.37809	1.31766	1.40738
1.07677	1.37293	1.36998	1.39243
1.11545	1.36416	1.40818	1.38247
1.14484	1.36726	1.43974	1.36503
1.17114	1.37499	1.47379	1.34012
1.21858	1.39510	1.50784	1.30025
1.52860	1.21056	1.52030	1.25956

Fonte: Autores.

Figura 17 – Pontos utilizados na interpolação superpostos ao contorno da folha.



Fonte: Autores.

Após a obtenção do conjunto de coordenadas, aplicou-se o algoritmo desenvolvido para o cálculo do *spline* cúbico associado aos dados (Apêndice B). Os coeficientes resultantes desse procedimento, que definem cada trecho da curva interpoladora, encontram-se apresentados na Tabela 19. A partir desses coeficientes, obteve-se um polinômio cúbico

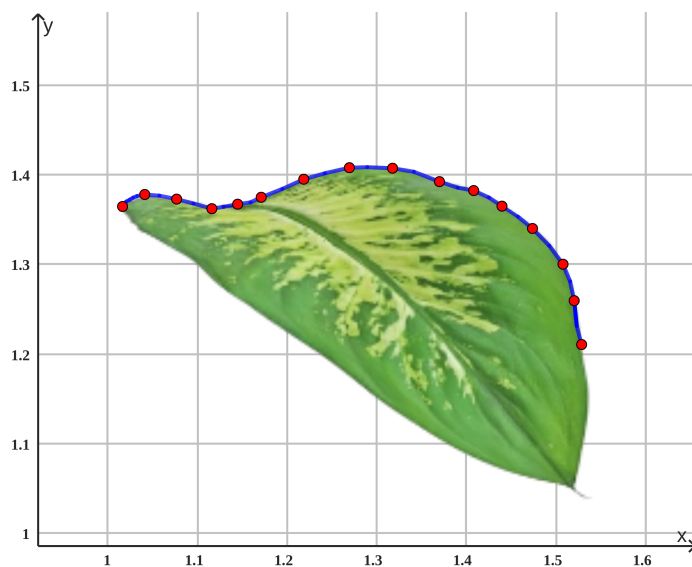
em cada intervalo, cujos coeficientes a_i, b_i, c_i, d_i foram determinados pelas condições de interpolação e continuidade descritas na Definição 4.3.2. A curva final reproduziu com elevada fidelidade o formato da folha, demonstrando a eficiência dos *splines* cúbicos na representação de contornos suaves e geometricamente consistentes, conforme ilustrado na Figura 18.

Tabela 19 – Coeficientes do *spline* cúbico interpoladora $S_i(x)$.

i	x_i	a_i	b_i	c_i	d_i
1	1.01644	1.364680	0.448530	0.000000	-13.626779
2	1.04119	1.378090	0.063715	-1.026402	1.139873
3	1.07677	1.372930	-0.116644	0.902919	-4.689693
4	1.11545	1.364160	-0.063321	0.354145	3.659976
5	1.14484	1.367260	0.113279	0.686568	-3.719717
6	1.17114	1.374990	0.231904	0.160100	0.140889
7	1.21858	1.395100	0.435868	0.359050	-0.584762
8	1.26950	1.407380	0.035417	-0.490787	3.193245
9	1.31766	1.407380	-0.045667	0.007621	-0.999696
10	1.36998	1.392430	-0.224160	-0.157833	1.428766
11	1.40818	1.382470	-0.370502	0.038481	1.320498
12	1.43974	1.365030	-0.662243	0.162788	1.332308
13	1.47379	1.340120	-0.941913	0.301552	-2.316812
14	1.50784	1.300250	-1.530960	0.046835	3.864388
15	1.52030	1.259560	-2.138805	0.191384	-25.263595

Fonte: Autores.

Figura 18 – Contorno da parte superior da folha por *spline* cúbico.



Fonte: Autor.

Vale destacar algumas limitações para o uso do *spline* cúbico. Conforme observado

por Burden *et al.*, ao aplicar o método ao contorno do perfil superior de um pato em pleno vôo [8, p. 171], a representação da curva na forma tradicional $y = f(x)$ torna-se impraticável em situações nas quais a geometria apresenta trechos verticais, dobras acentuadas ou regiões nas quais a curva deixa de ser função de x . Nesse cenário, a exigência de monotonicidade dos nós e de unicidade da imagem y para cada valor de x é violada, impedindo a construção de um *spline* cúbico suave ao longo de todo o contorno.

Além disso, Burden *et al.* destacam que uma solução inicial para esse problema consiste em dividir a curva em porções que satisfaçam localmente a condição de ser função de x , ajustando um *spline* cúbico separado para cada trecho. Essa fragmentação, no entanto, introduz um problema crítico de continuidade. Ao tratar os trechos de forma isolada, as condições de suavidade nas junções (pontos de transição) deixam de ser garantidas. Na prática, isso resulta em uma perda da classe de continuidade C^2 , manifestando-se como "quinas" ou mudanças abruptas de curvatura que invalidam o uso da curva em contextos que exigem suavidade rigorosa, como na computação gráfica ou no controle de trajetórias robóticas.

Por essa razão, eles enfatizam que uma abordagem mais eficiente e matematicamente consistente para curvas desse tipo é o uso de *splines* cúbicos em sua forma paramétrica. Nesse modelo, cada coordenada é descrita como uma função de um parâmetro independente t :

$$S(t) = (S_x(t), S_y(t)),$$

reduzindo o problema bidimensional a dois problemas unidimensionais independentes. Essa parametrização elimina a necessidade de monotonicidade em x , garante suavidade global e permite reconstruir contornos arbitrariamente complexos — incluindo curvas fechadas e formatos orgânicos que não podem ser descritos na forma funcional usual.

Assim, embora não seja foco de nosso trabalho, os *splines* paramétricos constituem a técnica mais apropriada para representar curvas cuja geometria viola as hipóteses da interpolação tradicional.

5 Conclusão

Este trabalho investigou a interpolação polinomial como uma ferramenta principal para a aproximação de funções no âmbito da Análise Numérica e da Teoria da Aproximação. Estabeleceu-se uma base teórica sólida, com a demonstração do Teorema da Existência e Unicidade do Polinômio Interpolador para dados discretos sob certas condições, contemplando métodos clássicos, como os de Lagrange e de Newton, e técnicas mais avançadas, como a interpolação por *splines*. Além disso, foi estudado o Erro de Interpolação discutindo-se a precisão, os limitantes superiores e a estimativa via diferenças divididas. A implementação computacional dessas técnicas, utilizando a linguagem *python*, permitiu validar os conceitos teóricos e demonstrar a viabilidade das diferentes abordagens na modelagem de funções e na representação de diversos conjuntos de dados. Observamos, na demonstração da existência e unicidade do polinômio interpolador, entretanto, que a resolução direta por meio do sistema linear associado à matriz de Vandermonde pode ser numericamente instável devido ao seu mau condicionamento.

A análise detalhada dos métodos construtivos evidenciou características e limitações próprias de cada abordagem. Como abordado ao longo do Capítulo 2, o Método de Lagrange apresenta uma forma explícita do polinômio, dispensando a resolução de sistemas lineares; porém, mostrou-se pouco eficiente em processos incrementais, pois qualquer adição de novos pontos exige o recálculo completo da expressão. Em contrapartida, o Método de Newton, baseado em diferenças divididas, revelou-se computacionalmente mais eficiente ao permitir a ampliação do polinômio de grau n para $n + 1$ reutilizando cálculos anteriores.

A aplicabilidade prática das técnicas estudadas foi verificada por meio de dois modelos específicos, sendo que o Método de Lagrange foi empregado na elaboração de um modelo de calibração de termômetros, enquanto o Método de Newton foi utilizado na construção de um modelo para a média móvel dos sasos de COVID-19.

Ademais, o estudo dedicou atenção importante à Análise do Erro de Interpolação, presente no Capítulo 3, componente fundamental para avaliar a precisão da aproximação. O Teorema do Erro relaciona a magnitude do erro interpolatório à derivada de ordem $(n + 1)$ da função original, possibilitando a obtenção de limitantes superiores para $|E_n(x)|$.

Adicionalmente, no Capítulo 4 discutimos uma limitação central da interpolação polinomial de alto grau: o Fenômeno de Runge, caracterizado por oscilações acentuadas nas extremidades do intervalo. Como alternativa, introduziu-se a interpolação por *splines*, com destaque para os *splines* cúbicos. Ao utilizarem polinômios de baixo grau definidos por partes, esses modelos garantem continuidade até a segunda derivada (C^2), o que proporciona aproximações mais estáveis, precisas e com alto grau de suavidade. A eficácia dessa abordagem foi ilustrada na aplicação envolvendo a reconstrução de imagens digitais, especificamente no contorno da folhagem.

Ao modelar perfis em que a curva não pode ser expressa como uma função única de x ou apresenta descontinuidades de suavidade, a interpolação por *splines* cúbicas padrão revela limitações significativas. Para superar essas restrições em contornos complexos — como o desenho completo de uma folha — destaca-se, como perspectiva para trabalhos futuros, o uso de *splines* paramétricas. Essa abordagem permite a representação eficiente de geometrias e trajetórias complexas por meio de curvas parametrizadas, ampliando substancialmente as possibilidades de modelagem em áreas como *design* gráfico, animação computacional e robótica, além de reforçar que a escolha do método de interpolação deve considerar cuidadosamente a natureza dos dados e os objetivos da aproximação.

Conclui-se que a interpolação polinomial é uma ferramenta versátil e de grande relevância, com aplicações que abrangem desde a engenharia de precisão até a análise de dados epidemiológicos. As implementações e comparações apresentadas fornecem uma base sólida para estudos futuros, especialmente no desenvolvimento de métodos que visam mitigar o fenômeno de Runge e aumentar a estabilidade das aproximações.

Referências

- 1 GONÇALVES, A. *Introdução à Álgebra*. 5. ed.. ed. Rio de Janeiro: Instituto de Matemática Pura e Aplicada (IMPA), 2011. (Projeto Euclides). ISBN 978-85-244-0108-4.
- 2 FLEMMING, D. M.; GONÇALVES, M. B. *Cálculo A: Funções, limite, derivação e integração*. 6. ed. rev. e ampl.. ed. São Paulo: Pearson Prentice Hall, 2006. ISBN 978-85-7605-115-2.
- 3 BOLDRINI, J. L. et al. *Álgebra Linear*. 3. ed.. ed. São Paulo: Harper & Row do Brasil, 1980.
- 4 ATKINSON, K. E. *An Introduction to Numerical Analysis*. 2. ed.. ed. [S.l.]: John Wiley & Sons, 2008. ISBN 978-0471433378.
- 5 RUGGIERO, M. A. G.; LOPES, V. L. da R. *Cálculo Numérico: Aspectos Teóricos e Computacionais*. 2. ed.. ed. São Paulo: Pearson, 2000.
- 6 PEREIRA, H. W. de L. *Estudo e Implementação de Técnicas de Interpolação Polinomial*. Dissertação (Trabalho de Conclusão de Curso de Bacharelado em Matemática) — Universidade Federal de Uberlândia, Uberlândia-MG, 2024. ORCID do autor: <https://orcid.org/0009-0000-6407-0145>. Disponível em: <<https://repositorio.ufu.br/handle/123456789/41921>>.
- 7 CHAPRA, S. C.; CANALE, R. P. *Métodos Numéricos para Engenharia*. 5. ed.. ed. São Paulo: AMGH Editora Ltda., 2011. Tradução técnica de Helena Castro. ISBN 978-85-86804-87-8.
- 8 BURDEN, R. L.; FAIRES, J. D.; BURDEN, A. M. *Análise Numérica*. 3. ed.. ed. São Paulo: Cengage Learning, 2015. ISBN 978-85-221-2341-4.
- 9 JUSTO, D. A. R. et al. *Cálculo Numérico: Um Livro Colaborativo*. Porto Alegre, RS: Projeto REAMAT (UFRGS), 2020. Versão Scilab. Disponível em <<https://www.ufrgs.br/reatmat>>.
- 10 Instituto Nacional de Metrologia, Qualidade e Tecnologia (INMETRO). *Vocabulário Internacional de Metrologia(VIM): Conceitos fundamentais e gerais e termos associados*. 1. ed. luso-brasileira. ed. Rio de Janeiro: INMETRO, 2012. Tradução do JCGM 200:2012. ISBN 978-85-86920-09-7.
- 11 Brasil. Ministério da Saúde. *Painel Coronavírus COVID-19*. 2025. <https://infoms.saude.gov.br/extensions/covid-19_html/covid-19_html.html#>. Acesso em: 22 nov. 2025.

- 12 MORETTIN, P. A.; TOLOI, C. M. d. C. *Análise de Séries Temporais: Modelos lineares univariados*. 3. ed. São Paulo: Editora Blucher, 2018. Volume 1. ISBN 978-85-212-1352-9.
- 13 SANTOS, L. M. M. *Modelos Matemáticos Epidemiológicos do Tipo SIS e SIR e o Segundo Método de Lyapunov*. Dissertação (Monografia de Graduação em Matemática) — Universidade Federal Rural de Pernambuco (UFRPE), Recife-PE, 2023. Disponível em: <<https://arandu.ufrpe.br/>>.
- 14 FREIRE, P. *Pedagogia da autonomia: saberes necessários à prática educativa*. 54. ed.. ed. Rio de Janeiro: Paz & Terra, 1996. ISBN 978-8577531639.
- 15 FRANCO, N. M. B. *Cálculo Numérico*. 1. ed.. ed. São Paulo: Pearson Prentice Hall, 2006. ISBN 978-85-7605-087-2.
- 16 QUARTERONI, A.; SALERI, F. *álculo Científico com MATLAB e Octave*. Milano, Itália: Springer, 2007. Tradução da obra italiana.
- 17 ROSA, C. F. *Série de Taylor e Aplicações*. Dissertação (Trabalho de Conclusão de Curso (Licenciatura em Matemática)) — Universidade Federal de Santa Catarina, Florianópolis - SC, 2013.
- 18 LIMA, E. L. *Espaços métricos*. 6. ed.. ed. Rio de Janeiro: IMPA, 2020. 308 p. ISBN 978-65-990528-7-3.
- 19 OLIVEIRA, M. V. d. *Cálculo Numérico da Forma Normal de Floquet e Aplicações em Controle de Sistemas Dinâmicos*. Dissertação (Dissertação (Mestrado em Matemática)) — Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, 2013. Disponível em: <<https://hdl.handle.net/1843/EABA-9DEHQY>>.

Apêndice A — Resultados Complementares

A.1 Resultados sobre Aproximações de Funções Reais

Definição A.1.1 (Polinômio de Taylor de grau n [17]). Seja $f : I \rightarrow \mathbb{R}$ uma função definida no intervalo I , e seja $x_0 \in I$ um ponto onde f é n vezes diferenciável. O n -ésimo polinômio de Taylor da função f centrado em $x = x_0$ é o polinômio

$$T_n(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \cdots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n.$$

Teorema A.1.1 (Fórmula de Taylor com Resto de Lagrange [17]). Se $f : [a, b] \rightarrow \mathbb{R}$ uma função com n derivadas contínuas e $f^{(n+1)}$ definida em todo (a, b) . Seja $x_0 \in [a, b]$ então existe ξ entre x_0 e x tal que

$$f(x) = \sum_{k=0}^n \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k + \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)^{n+1}.$$

Neste caso,

$$R_n(x) = f(x) - T_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)^{n+1}$$

é o resto (ou erro de truncamento) da série de Taylor.

Teorema A.1.2 (Teorema da Aproximação de Weierstrass [18]). Seja $f : [a, b] \rightarrow \mathbb{R}$ uma função contínua. Para todo $\epsilon > 0$, existe um polinômio $p(t)$ tal que $|f(t) - p(t)| < \epsilon$, para todo $t \in [a, b]$.

A.2 Resultados de Álgebra Linear

Proposição A.2.1 (Multiplicação de fila por escalar [3]). Se A' é a matriz obtida ao multiplicar uma fila (linha ou coluna) de uma matriz quadrada A por um escalar k , então

$$\det(A') = k \cdot \det(A).$$

Demonstração. Seja A uma matriz quadrada de ordem $n \times n$. Considere a matriz A e a matriz A' obtida pela multiplicação da i -ésima linha de A pelo escalar k :

$$A = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_{i1} & \alpha_{i2} & \dots & \alpha_{in} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_{n1} & \alpha_{n2} & \dots & \alpha_{nn} \end{pmatrix} \quad \text{e} \quad A' = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ k\alpha_{i1} & k\alpha_{i2} & \dots & k\alpha_{in} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_{n1} & \alpha_{n2} & \dots & \alpha_{nn} \end{pmatrix}.$$

Calculamos o determinante de A' pelo desenvolvimento de Laplace na i -ésima linha. Note que os cofatores c_{ij} dos elementos da i -ésima linha de A' são os mesmos de A , pois eles dependem apenas dos elementos das outras linhas:

$$\begin{aligned} \det A' &= k\alpha_{i1}c_{i1} + k\alpha_{i2}c_{i2} + \dots + k\alpha_{in}c_{in} \\ &= k(\alpha_{i1}c_{i1} + \alpha_{i2}c_{i2} + \dots + \alpha_{in}c_{in}) \\ &= k\det A. \end{aligned}$$

□

Proposição A.2.2. [3] O determinante de uma matriz real $A = (a_{ij})$ de ordem $n \times n$ pode ser calculado pelo desenvolvimento de Laplace ao longo de qualquer linha i (para $i = 1, \dots, n$):

$$\det(A) = \sum_{j=1}^n a_{ij}C_{ij},$$

onde a_{ij} é o elemento na i -ésima linha e j -ésima coluna de A , e C_{ij} é o cofator (ou complemento algébrico) do elemento a_{ij} . O cofator C_{ij} é definido por:

$$C_{ij} = (-1)^{i+j} \det(M_{ij}),$$

sendo M_{ij} a submatriz de A obtida pela remoção da i -ésima linha e da j -ésima coluna.

Proposição A.2.3 (Teorema de Jacobi [3]). Seja A uma matriz quadrada $n \times n$. Se a matriz A' é obtida a partir de A pela adição de um múltiplo de uma fila (linha ou coluna) a outra fila paralela, então os determinantes são iguais:

$$\det(A) = \det(A').$$

Demonstração. Demonstraremos primeiramente para o caso em que a operação é realizada em colunas.

Seja A uma matriz quadrada de ordem n , e A' a matriz obtida pela adição de k vezes a k -ésima coluna à l -ésima coluna, onde $k \neq l$. As matrizes A e A' são dadas por:

$$A = \begin{pmatrix} \alpha_{11} & \dots & \alpha_{1k} & \dots & \alpha_{1l} & \dots & \alpha_{1n} \\ \vdots & & \vdots & & \vdots & & \vdots \\ \alpha_{n1} & \dots & \alpha_{nk} & \dots & \alpha_{nl} & \dots & \alpha_{nn} \end{pmatrix}$$

e

$$A' = \begin{pmatrix} \alpha_{11} & \cdots & \alpha_{1k} & \cdots & \alpha_{1l} + k\alpha_{1k} & \cdots & \alpha_{1n} \\ \vdots & & \vdots & & \vdots & & \vdots \\ \alpha_{n1} & \cdots & \alpha_{nk} & \cdots & \alpha_{nl} + k\alpha_{nk} & \cdots & \alpha_{nn} \end{pmatrix}.$$

Pela propriedade de linearidade do determinante em relação à l -ésima coluna [3], o determinante de A' pode ser escrito como a soma de dois determinantes:

$$\begin{aligned} \det(A') &= \begin{vmatrix} \alpha_{11} & \cdots & \alpha_{1k} & \cdots & \alpha_{1l} & \cdots & \alpha_{1n} \\ \vdots & & \vdots & & \vdots & & \vdots \\ \alpha_{n1} & \cdots & \alpha_{nk} & \cdots & \alpha_{nl} & \cdots & \alpha_{nn} \end{vmatrix} + \begin{vmatrix} \alpha_{11} & \cdots & \alpha_{1k} & \cdots & k\alpha_{1k} & \cdots & \alpha_{1n} \\ \vdots & & \vdots & & \vdots & & \vdots \\ \alpha_{n1} & \cdots & \alpha_{nk} & \cdots & k\alpha_{nk} & \cdots & \alpha_{nn} \end{vmatrix} \\ &= \det(A) + k \begin{vmatrix} \alpha_{11} & \cdots & \alpha_{1k} & \cdots & \alpha_{1k} & \cdots & \alpha_{1n} \\ \vdots & & \vdots & & \vdots & & \vdots \\ \alpha_{n1} & \cdots & \alpha_{nk} & \cdots & \alpha_{nk} & \cdots & \alpha_{nn} \end{vmatrix}. \end{aligned}$$

No segundo determinante (à direita), as colunas k e l são idênticas. Uma propriedade fundamental do determinante é que, se uma matriz possui duas colunas idênticas (ou proporcionais), seu determinante é nulo. Portanto,

$$\det(A') = \det(A) + k \cdot 0 = \det(A).$$

O caso em que as filas paralelas são linhas é análogo, utilizando a linearidade do determinante em relação às linhas. Alternativamente, como $\det(A) = \det(A^T)$ e a transposição preserva esta operação elementar, a prova para linhas segue diretamente da prova para colunas. \square

Teorema A.2.1 (Regra de Cramer [3]). Seja $Ax = b$ um sistema linear de n equações e n incógnitas, onde A é a matriz dos coeficientes ($n \times n$), x é o vetor das incógnitas (x_1, x_2, \dots, x_n) e b é o vetor dos termos independentes ($n \times 1$). Se o determinante da matriz dos coeficientes, $\det(A)$ (ou D), for diferente de zero ($\det(A) \neq 0$), então o sistema $Ax = b$ tem solução única dada por:

$$x_i = \frac{\det(A_i)}{\det(A)}, \quad \text{para } i = 1, 2, \dots, n,$$

onde A_i é a matriz obtida substituindo a i -ésima coluna de A pelo vetor coluna dos termos independentes b .

Proposição A.2.4 (Propriedade Fundamental dos Autovalores [3]). O determinante de uma matriz quadrada A de ordem n é igual ao produto de seus n autovalores $(\lambda_1, \lambda_2, \dots, \lambda_n)$, incluindo suas multiplicidades algébricas.

Demonstração. O polinômio característico $p(\lambda)$ de uma matriz A é definido pelo determinante da matriz $(A - \lambda I)$:

$$p(\lambda) = \det(A - \lambda I),$$

no qual os autovalores $\lambda_1, \lambda_2, \dots, \lambda_n$ são as n raízes do polinômio $p(\lambda)$. Assim, ele pode ser fatorado como:

$$p(\lambda) = (-1)^n (\lambda - \lambda_1)(\lambda - \lambda_2) \cdots (\lambda - \lambda_n). \quad (\text{A.1})$$

O determinante de A é obtido avaliando o polinômio característico em $\lambda = 0$, isto é:

$$\det(A) = \det(A - 0I) = p(0).$$

Ao substituir $\lambda = 0$ na Equação (A.1) e simplificamos a multiplicação dos sinais negativos:

$$p(0) = (-1)^n (0 - \lambda_1)(0 - \lambda_2) \cdots (0 - \lambda_n)$$

$$p(0) = (-1)^n (-\lambda_1)(-\lambda_2) \cdots (-\lambda_n)$$

$$p(0) = (-1)^n \cdot [(-1)^n \lambda_1 \lambda_2 \cdots \lambda_n]$$

Como $(-1)^n \cdot (-1)^n = (-1)^{2n} = 1$, a expressão se simplifica para:

$$\det(A) = p(0) = \lambda_1 \lambda_2 \cdots \lambda_n.$$

□

Proposição A.2.5. [3] Seja A uma matriz quadrada de ordem n . A matriz A é singular (não-invertível) se, e somente se, $\lambda = 0$ é um autovalor de A .

Demonstração. Suponha que a matriz A seja singular. A condição para que λ seja um autovalor de A é dada pela equação característica:

$$\det(A - \lambda I) = 0. \quad (\text{A.2})$$

Queremos verificar se $\lambda = 0$ é um autovalor. Substituindo $\lambda = 0$ na Equação (A.2), obtemos:

$$\det(A - 0I) = \det(A) = 0. \quad (\text{A.3})$$

Como $\det(A)$ é igual a zero, pois A é singular, o valor $\lambda = 0$ satisfaz a equação característica. Pela definição da equação característica, isso significa que $\lambda = 0$ é uma raiz do polinômio característico e, conseqüentemente, um autovalor de A .

Agora, suponha que $\lambda = 0$ é um autovalor de A . Pela definição de autovalor, $\lambda = 0$ é um autovalor de A se satisfaz a equação característica (A.2). Substituindo $\lambda = 0$ na equação característica, encontramos:

$$\det(A - 0I) = \det(A) = 0. \quad (\text{A.4})$$

Portanto, como $\det(A) = 0$, a matriz A é, por definição, singular (não-invertível). \square

Proposição A.2.6. [3] A matriz A dos coeficientes do sistema linear obtida na construção de um *spline* cúbico é estritamente diagonal dominante.

Demonstração. Para verificar a dominância diagonal estrita, analisamos a estrutura da matriz A em dois casos:

1. Linhas internas: Para cada $i = 1, 2, \dots, n - 1$, os elementos da i -ésima linha são definidos pela equação:

$$h_{i-1}c_{i-1} + 2(h_{i-1} + h_i)c_i + h_i c_{i+1} = D_i,$$

com $D_i = \frac{3}{h_i}(a_{i+1} - a_i) - \frac{3}{h_{i-1}}(a_i - a_{i-1})$. Neste caso, o elemento da diagonal principal é $a_{i,i} = 2(h_{i-1} + h_i)$, enquanto os elementos fora da diagonal (adjacentes) são $a_{i,i-1} = h_{i-1}$ e $a_{i,i+1} = h_i$. Visto que os nós estão ordenados ($x_0 < x_1 < \dots < x_n$), os comprimentos dos subintervalos h_i são estritamente positivos. Assim, a condição de dominância diagonal estrita se traduz em:

$$|2(h_{i-1} + h_i)| > |h_{i-1}| + |h_i| \implies 2h_{i-1} + 2h_i > h_{i-1} + h_i.$$

Simplificando a expressão, obtemos $h_{i-1} + h_i > 0$, o que é garantido pela positividade de cada h_i .

2. Linhas de contorno: As equações para a primeira ($i = 0$) e última ($i = n$) linhas decorrem das condições de contorno naturais impostas ($c_0 = 0$ e $c_n = 0$). Na linha 0, temos $a_{0,0} = 1$ e $a_{0,j} = 0$ para todo $j \neq 0$. Analogamente, na linha n , temos $a_{n,n} = 1$ e $a_{n,j} = 0$ para todo $j \neq n$. Em ambos os casos, a condição $|a_{i,i}| > \sum_{j \neq i} |a_{i,j}|$ simplifica-se para $1 > 0$, o que é trivialmente satisfeito.

Portanto, como a condição de dominância diagonal estrita é verificada em todas as linhas, a matriz A é estritamente diagonal dominante.

\square

A.3 Simetria das Diferenças Divididas

Proposição A.3.1. [5] A diferença dividida $f[x_0, x_1, \dots, x_k]$ é simétrica em relação a seus argumentos. Isto é, o valor de $f[x_0, x_1, \dots, x_k]$ é invariante sob qualquer permutação dos nós x_0, x_1, \dots, x_k .

Demonstração. A demonstração da simetria é realizada por indução sobre a ordem k da diferença dividida.

Note que a diferença dividida de primeira ordem, a simetria é imediata:

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{-(f(x_0) - f(x_1))}{-(x_0 - x_1)} = \frac{f(x_0) - f(x_1)}{x_0 - x_1} = f[x_1, x_0].$$

Para a segunda ordem, mostraremos que a diferença dividida $f[x_0, x_1, x_2]$ é invariante sob a transposição dos argumentos x_0 e x_1 , provando que $f[x_0, x_1, x_2] = f[x_1, x_0, x_2]$.

Partimos da definição recursiva de $f[x_0, x_1, x_2]$:

$$\begin{aligned} f[x_0, x_1, x_2] &= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} \\ &= \frac{1}{x_2 - x_0} \left[\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0} \right] \\ &= \frac{f(x_0)(x_2 - x_1) + f(x_1)(x_0 - x_2) + f(x_2)(x_1 - x_0)}{(x_1 - x_0)(x_2 - x_1)(x_2 - x_0)}. \end{aligned}$$

Agora, reescrevemos o numerador para corresponder à forma recursiva de $f[x_1, x_0, x_2]$:

$$\begin{aligned} f[x_0, x_1, x_2] &= \frac{f(x_0)(x_2 - x_1) + f(x_1)(x_0 - x_2) + f(x_2)(x_1 - x_0)}{(x_1 - x_0)(x_2 - x_1)(x_2 - x_0)} \\ &= \frac{1}{x_2 - x_1} \cdot \left[\frac{f(x_2)(x_1 - x_0) - f(x_0)(x_1 - x_2) + f(x_1)(x_0 - x_2)}{(x_1 - x_0)(x_2 - x_0)} \right] \\ &= \frac{1}{x_2 - x_1} \cdot \left[\frac{f(x_2)(x_1 - x_0) - f(x_0)(x_1 - x_0 + x_0 - x_2) + f(x_1)(x_0 - x_2)}{(x_1 - x_0)(x_2 - x_0)} \right] \\ &= \frac{1}{x_2 - x_1} \cdot \left[\frac{f(x_2) - f(x_0)}{x_2 - x_0} - \frac{f(x_1) - f(x_0)}{x_1 - x_0} \right] \\ &= \frac{f[x_0, x_2] - f[x_1, x_0]}{x_2 - x_1} \\ &= f[x_1, x_0, x_2]. \end{aligned}$$

Portanto, a propriedade de simetria é verificada para $k = 2$.

Agora, iremos mostrar que $f[x_0, x_1, \dots, x_k]$ é invariante sob qualquer troca de dois de seus argumentos, o que é suficiente para provar a simetria total, usando a hipótese de indução.

Consideramos, em particular, a troca de dois argumentos quaisquer, x_i e x_j , onde $i, j \in \{0, 1, \dots, k\}$. Como a diferença dividida de ordem k é definida recursivamente como:

$$f[x_0, x_1, \dots, x_k] = \frac{f[x_1, \dots, x_k] - f[x_0, \dots, x_{k-1}]}{x_k - x_0}.$$

Se trocarmos o primeiro e o último nó, x_0 e x_k , a nova diferença dividida é $f[x_k, x_1, \dots, x_0]$. Usando a mesma definição recursiva, mas com x_k como primeiro nó e x_0 como último:

$$f[x_k, x_1, \dots, x_0] = \frac{f[x_1, \dots, x_0] - f[x_k, \dots, x_1]}{x_0 - x_k}.$$

Pela hipótese indutiva, sabemos que as diferenças divididas de ordem $k - 1$ (o numerador) são simétricas:

$$\begin{aligned} f[x_1, \dots, x_0] &= f[x_0, x_1, \dots, x_{k-1}] \\ f[x_k, x_1, \dots, x_{k-1}] &= f[x_1, \dots, x_{k-1}, x_k]. \end{aligned}$$

Substituindo na expressão de $f[x_k, x_1, \dots, x_0]$:

$$\begin{aligned} f[x_k, x_1, \dots, x_0] &= \frac{f[x_0, \dots, x_{k-1}] - f[x_1, \dots, x_k]}{-(x_k - x_0)} \\ &= \frac{-(f[x_1, \dots, x_k] - f[x_0, \dots, x_{k-1}])}{-(x_k - x_0)} \\ &= \frac{f[x_1, \dots, x_k] - f[x_0, \dots, x_{k-1}]}{x_k - x_0} \\ &= f[x_0, x_1, \dots, x_k]. \end{aligned}$$

Portanto, a troca dos nós x_0 e x_k não altera o valor da diferença dividida. Este procedimento pode ser estendido para mostrar que a troca de quaisquer dois nós adjacentes (e, conseqüentemente, quaisquer dois nós) não altera o resultado.

□

A.4 Teorema de Gerschgorin

O resultado a seguir, demonstrado por S. Gerschgorin em 1931, estabelece limitantes no plano complexo para a localização dos autovalores de uma matriz a partir de seus elementos diagonais e da soma de seus elementos fora da diagonal.

Teorema A.4.1 (Gerschgorin). Seja $A = (a_{ij}) \in M_n(\mathbb{C})$. Para cada $i \in \{1, \dots, n\}$, definimos o disco de Gerschgorin D_i como o subconjunto do plano complexo dado por:

$$D_i = \{z \in \mathbb{C} : |z - a_{ii}| \leq R_i\}$$

onde $R_i = \sum_{j \neq i} |a_{ij}|$ é o raio do disco, correspondente à soma dos módulos dos elementos da i -ésima linha, excluindo o elemento da diagonal. Então, todo autovalor λ de A pertence à união dos discos:

$$\sigma(A) \subseteq \bigcup_{i=1}^n D_i.$$

Demonstração. Seja $\lambda \in \sigma(A)$ um autovalor de A e $v = (v_1, v_2, \dots, v_n)^T \in \mathbb{C}^n \setminus \{0\}$ o vetor próprio associado, tal que $Av = \lambda v$. Consideremos a componente de v com magnitude máxima [cite: 12]. Seja $i \in \{1, \dots, n\}$ o índice tal que:

$$|v_i| = \max_{1 \leq j \leq n} |v_j|$$

Como $v \neq 0$, temos necessariamente que $|v_i| > 0$. Olhando para a i -ésima linha da equação matricial $Av = \lambda v$, obtemos:

$$\sum_{j=1}^n a_{ij} v_j = \lambda v_i.$$

Isolando o termo diagonal $a_{ii} v_i$ e reorganizando a soma, temos:

$$(\lambda - a_{ii}) v_i = \sum_{j \neq i} a_{ij} v_j.$$

Tomando o módulo em ambos os lados e aplicando a desigualdade triangular:

$$|\lambda - a_{ii}| \cdot |v_i| = \left| \sum_{j \neq i} a_{ij} v_j \right| \leq \sum_{j \neq i} |a_{ij}| \cdot |v_j|.$$

Dividindo ambos os membros por $|v_i|$ (que é estritamente positivo):

$$|\lambda - a_{ii}| \leq \sum_{j \neq i} |a_{ij}| \frac{|v_j|}{|v_i|}$$

Pela escolha do índice i , sabemos que $\frac{|v_j|}{|v_i|} \leq 1$ para todo j . Logo:

$$|\lambda - a_{ii}| \leq \sum_{j \neq i} |a_{ij}| = R_i$$

O que prova que $\lambda \in D_i$ para algum índice i e, conseqüentemente, $\lambda \in \bigcup_{i=1}^n D_i$.

□

A.5 Teorema de Lévy-Desplanques

Teorema A.5.1. [19] Se uma matriz é estritamente diagonal dominante, então ela é não-singular.

Demonstração. Para demonstrar que A é não-singular, basta provar que 0 não é autovalor de A . Como A é estritamente diagonal dominante, temos que, para todo i :

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| = R_i.$$

Ou seja, o ponto 0 não pertence a nenhum dos discos de Gerschgorin. Assim, como todos os autovalores de A pertencem à união desses discos. Logo, 0 não pode ser autovalor de A , o que implica que A é não-singular.

□

Apêndice B — Principais Algoritmos Implementados

B.1 Polinômio interpolador por sistema linear (Exemplo [1.3.2](#))

```
1  # Universidade Federal Rural de Pernambuco
2  # Curso de Licenciatura em Matemática
3  # Data : 08\11\2025
4  # Autores : Jefferson Matheus de L. Souza e Maria Ângela C. Didier
5  # Polinômio Interpolador por Sistema Linear
6  # IMPLEMENTACAO EM PYTHON
7
8  import numpy as np
9  import sympy as sp
10 import matplotlib.pyplot as plt
11
12 # --- 1. DADOS DE ENTRADA ---
13 x_dados = np.array([0, 1, 3])
14 y_dados = np.array([1, 3, 1])
15 n = len(x_dados)
16
17 # --- 2. MONTAGEM E SOLUÇÃO DO SISTEMA LINEAR ---
18 # Matriz A (Vandermonde):
19 # np.vander(x_dados, n) cria A na ordem [..., x^2, x^1, x^0].
20 # np.fliplr inverte as colunas para a ordem canônica: [x^0, x^1, x^2, ...]
21 A = np.fliplr(np.vander(x_dados, n))
22 B = y_dados
23
24 print("---- SISTEMA LINEAR ----")
25 print("Matriz A (Vandermonde, [1, x_i, x_i^2...]):")
26 print(A)
27 print("\nVetor B (Valores de y):")
28 print(B)
```

```

29
30 # --- 3. RESOLVER SISTEMA E OBTER COEFICIENTES ---
31 coef = np.linalg.solve(A, B)
32
33 # Converter coeficientes para frações SymPy
34 coef_frac = [sp.Rational(c).limit_denominator() for c in coef]
35
36 print("\nCoeficientes do polinômio (a0, a1, a2):")
37 for i, c in enumerate(coef_frac):
38     print(f"a{i} = {c}")
39
40 # --- 4. CONSTRUÇÃO E PLOTAGEM DO POLINÔMIO ---
41 x = sp.symbols('x')
42 # Polinômio:  $P(x) = a_0*x^0 + a_1*x^1 + a_2*x^2 + \dots$ 
43 polinomio = sum(c * x**i for i, c in enumerate(coef_frac))
44 polinomio_simples = sp.simplify(polinomio)
45
46 print("\n--- POLINÔMIO RESULTANTE ---")
47 print("Polinômio Interpolador:")
48 sp.pprint(polinomio_simples)
49
50 # Converter para função numérica para plotagem
51 p_func = sp.lambdify(x, polinomio_simples, "numpy")
52
53 # PLOTAGEM
54 x_range = np.linspace(min(x_dados) - 0.5, max(x_dados) + 0.5, 400)
55 y_range = p_func(x_range)
56
57 plt.figure(figsize=(8, 5))
58 plt.plot(x_range, y_range,
59         label=fr"$P(x) = {sp.latex(polinomio_simples)}$",
60         linewidth=2)
61 plt.scatter(x_dados, y_dados, color="black", label=
62 "Pontos amostrados", s=80, zorder=5)
63
64 plt.title("Gráfico do Polinômio Interpolador (Método Direto)", fontsize=14)
65 plt.xlabel("$x$")
66 plt.ylabel("$y$")
67 plt.grid(True, linestyle='--', alpha=0.6)
68 plt.legend(loc='upper right')
69 plt.show()

```

B.2 Calibração de termômetro pelo método de Lagrange

```

1  # Universidade Federal Rural de Pernambuco
2  # Curso de Licenciatura em Matemática
3  # Data : 08\11\2025
4  # Autores : Jefferson Matheus de L. Souza e Maria Ângela C. Didier
5  # Polinômio interpolador pelo método de Lagrange: Calibragem de termômetro
6  # IMPLEMENTACAO EM PYTHON
7
8  import numpy as np
9  import matplotlib.pyplot as plt
10 from sympy import symbols, simplify, prod, fraction
11
12 # --- DADOS DA TABELA DE CALIBRAÇÃO ---
13 # x: Leitura do instrumento (Nós)
14 # y: Temperatura padrão (Valor real de referência)
15 x_data = np.array([0, 10, 20, 30, 40], float)
16 y_data = np.array([17, 32, 1, 54, 245], float)
17
18 # 1. Cálculo dos polinômios de base e do polinômio interpolador (Simbólico)
19 def lagrange_interpolacao(x, y):
20     x_sym = symbols("x")
21     n = len(x)
22
23     L = [] # Lista dos polinômios de base  $L_i(x)$ 
24
25     print("\n--- POLINÔMIOS DE BASE  $L_i(x)$  ---")
26     for i in range(n):
27         # Numerador: produto de  $(x - x_j)$ 
28         numer = prod([x_sym - x[j] for j in range(n) if j != i])
29         # Denominador: produto de  $(x_i - x_j)$ 
30         denom = prod([x[i] - x[j] for j in range(n) if j != i])
31         L_i = numer / denom
32         L.append(L_i)
33         print(f" $L_{i}(x) = \frac{{numer}}{denom}$ ")
34
35     # Polinômio interpolador  $P(x) = \text{somatório de } y_i * L_i(x)$ 
36     P = simplify(sum(y[i] * L[i] for i in range(n)))
37
38     print("\n--- P(x) NA FORMA CANÔNICA (COEFICIENTES EM FRAÇÃO) ---")
39
40     # Extração dos coeficientes para exibição limpa
41     poly_p = P.as_poly(x_sym)

```

```

42     coefs = poly_p.all_coeffs()
43     termos_str = []
44
45     for i, c in enumerate(coefs):
46         grau = len(coefs) - i - 1
47         num, den = fraction(c)
48
49         # Formatação de sinal e fração
50         sinal = " + " if c > 0 and i > 0 else ""
51         if c < 0: sinal = " - " if i > 0 else "-"
52
53         termo = f"{abs(num)}/{den}" if den != 1 else f"{abs(num)}"
54         if grau > 0:
55             termo += f"x^{grau}" if grau > 1 else "x"
56
57         termos_str.append(f"{sinal}{termo}")
58
59     print("p_4(x) = " + "".join(termos_str))
60
61     # Retorna as bases, o objeto simbólico e uma função numérica
62     return L, P, lambda val: float(P.subs(x_sym, val))
63
64 # Executa o cálculo
65 L_i, P_sym, P_num_func = lagrange_interpolacao(x_data, y_data)
66
67 # 2. Cálculo das bases usando NumPy para plotagem fluida
68 def lagrange_numpy(x, x_plot):
69     n = len(x)
70     L_vals = []
71     for i in range(n):
72         # Vetorização para o numpy processar todo o x_plot de uma vez
73         termos = [ (x_plot - x[j]) / (x[i] - x[j]) for j in range(n) if
74                   j != i]
75         L_vals.append(np.prod(termos, axis=0))
76     return L_vals
77
78 # 3. Preparação para Plotagem
79 x_plot = np.linspace(x_data.min(), x_data.max(), 500)
80 # Aplica a função numérica P_num_func em cada ponto do gráfico
81 y_interp = np.array([P_num_func(val) for val in x_plot])
82
83 # --- CRIAÇÃO DO GRÁFICO ---

```

```

84 plt.figure(figsize=(12, 7))
85
86 # Plotagem da curva interpoladora
87 plt.plot(x_plot, y_interp, color="blue", linewidth=2.5, label=
88 "Polinômio Interpolador  $p_4(x)$ ")
89
90 # Plotagem dos pontos de calibração (nós)
91 plt.scatter(x_data, y_data, color="red", s=100, zorder=5, label=
92 "Pontos de Calibração ( $x_i, f(x_i)$ ")
93
94 # Títulos e Eixos
95 plt.title("Interpolação de Lagrange: Calibração do Instrumento",
96 fontsize=20, pad=20)
97 plt.xlabel("Leitura do instrumento  $x$  (°C)", fontsize=16)
98 plt.ylabel("Temperatura real padrão  $f(x)$  (°C)", fontsize=16)
99
100 # Estética do Gráfico
101 plt.grid(True, which='both', linestyle='--', alpha=0.5)
102 plt.axhline(0, color='black', linewidth=1)
103 plt.axvline(0, color='black', linewidth=1)
104 plt.xticks(np.arange(0, 41, 5), fontsize=12)
105 plt.yticks(fontsize=12)
106 plt.legend(fontsize=14, loc='upper left')
107
108 plt.tight_layout()
109 plt.show()

```

B.3 Média móvel dos casos de COVID-19 pelo método de Newton.

```

1 # Universidade Federal Rural de Pernambuco
2 # Curso de Licenciatura em Matemática
3 # Data : 08\11\2025
4 # Autores : Jefferson Matheus de L. Souza e Maria Ângela C. Didier
5 # Polinômio interpolador pelo método de Newton: Média móvel de COVID-19
6 # IMPLEMENTACAO EM PYTHON
7
8 import numpy as np
9 import matplotlib.pyplot as plt
10 from sympy import symbols, simplify, Rational, lambdify
11 from tabulate import tabulate
12

```

```
13 # 1. POLINÔMIO INTERPOLADOR DE NEWTON + DIFERENÇAS DIVIDIDAS
14 def polinomio_interpolador_newton(x_vals, y_vals):
15     """
16     Constrói o polinômio interpolador de Newton e gera a tabela de
17     diferenças divididas usando frações exatas (Rational).
18     """
19     n = len(x_vals)
20     tabela = [[None] * n for _ in range(n)]
21
22     # Primeira coluna: valores de y convertidos para fração
23     for i in range(n):
24         tabela[i][0] = Rational(y_vals[i])
25
26     # Construção da tabela de diferenças divididas
27     for j in range(1, n):
28         for i in range(n - j):
29             tabela[i][j] = Rational(
30                 tabela[i + 1][j - 1] - tabela[i][j - 1]
31                 ) / (x_vals[i + j] - x_vals[i])
32
33     # Construção simbólica do polinômio
34     x = symbols("x")
35     polinomio = 0
36     termo_prod = 1
37
38     for i in range(n):
39         polinomio += tabela[0][i] * termo_prod
40         termo_prod *= (x - x_vals[i])
41
42     return simplify(polinomio), tabela
43
44
45 # 2. IMPRIMIR TABELA FORMATADA
46 def imprimir_tabela_diferencas(tabela, x_vals, ano):
47     headers = ["x"] + [f"Ordem {i}" for i in range(len(tabela))]
48
49     tabela_formatada = [
50         [x_vals[i]] + [str(val) if val is not None else "" for
51         val in tabela[i]]
52         for i in range(len(x_vals))
53     ]
54
```

```

55     print(f"\n--- Tabela de Diferenças Divididas de Newton ({ano}) ---")
56     print(tabulate(tabela_formatada, headers=headers, tablefmt="grid"))
57
58 # 3. PLOTAGEM DO POLINÔMIO
59 def plotar_polinomio(polinomio, x_vals, y_vals, ano):
60     x = symbols("x")
61     f = lambdify(x, polinomio, "numpy")
62
63     x_plot = np.linspace(min(x_vals) - 1, max(x_vals) + 1, 800)
64     y_plot = f(x_plot)
65
66     plt.figure(figsize=(12, 7))
67
68     # Curva interpolada
69     plt.plot(
70         x_plot, y_plot,
71         label=f"Polinômio interpolador  $P_6(x)$  ({ano})",
72         linewidth=2
73     )
74     # Pontos amostrados
75     plt.scatter(
76         x_vals, y_vals,
77         color="red", s=60,
78         label=f"Dados amostrados ({ano})",
79         zorder=5
80     )
81
82     plt.xlabel("Semana epidemiológica", fontsize=12)
83     plt.ylabel("Número de casos", fontsize=12)
84
85     plt.grid(True, linestyle="--", alpha=0.6)
86     plt.legend()
87     plt.tight_layout()
88
89     plt.show()
90
91 # 4. EXECUÇÃO COMPLETA
92 x_vals_2021 = [6, 10, 14, 18, 23, 27, 31]
93 y_vals_2021 = [9700, 8440, 13718, 14926, 20515, 11647, 4734]
94
95 # Calcula polinômio e tabela
96 polinomio_sym, tabela_2021 =

```

```
97 polinomio_interpolador_newton(x_vals_2021, y_vals_2021)
98
99 # Exibe tabela
100 imprimir_tabela_diferencas(tabela_2021, x_vals_2021, 2021)
101
102 # Exibe polinômio final
103 print("\n--- Polinômio Interpolador Simplificado (2021) ---")
104 print(polinomio_sym)
105
106 # Plotagem
107 plotar_polinomio(polinomio_sym, x_vals_2021, y_vals_2021, 2021
```

B.4 Reconstrução digital do contorno de uma folha por *spline* cúbico.

```
1 # Universidade Federal Rural de Pernambuco
2 # Curso de Licenciatura em Matemática
3 # Data : 08\11\2025
4 # Autores : Jefferson Matheus de L. Souza e Maria Ângela C. Didier
5 # Polinômio interpolador por spline cúbico: Reconstrução digital
6 # IMPLEMENTACAO EM PYTHON
7 # Algoritmo baseado em :
8 # Burden et . al , Analise Numerica , 10 a Ed . (2015)
9 # Pagina 164 , Algoritmo 3.4
10
11 import numpy as np
12 import matplotlib.pyplot as plt
13 from scipy.interpolate import CubicSpline
14
15 # 1. PONTOS DO CONTORNO DA FOLHA
16 pontos = np.array([
17     (1.01644, 1.36468), (1.04119, 1.37809), (1.07677, 1.37293),
18     (1.11545, 1.36416), (1.14484, 1.36726), (1.17114, 1.37499),
19     (1.21858, 1.39510), (1.26950, 1.40738), (1.31766, 1.40738),
20     (1.36998, 1.39243), (1.40818, 1.38247), (1.43974, 1.36503),
21     (1.47379, 1.34012), (1.50784, 1.30025), (1.52030, 1.25956),
22     (1.52860, 1.21056)
23 ])
24
25 x, y = pontos[:, 0], pontos[:, 1]
26
```

```

27 # Verificação da monotonicidade (requisito da spline não-paramétrica)
28 if not np.all(np.diff(x) > 0):
29     raise ValueError("x não é estritamente crescente. Use parametrização.")
30
31 # 2. CRIAÇÃO DA SPLINE CÚBICA NATURAL
32 cs = CubicSpline(x, y, bc_type='natural')
33
34 # 3. COEFICIENTES DOS POLINÔMIOS
35 # cs.c → matriz 4 × (n1)
36 # linhas: d, c, b, a (ordem decrescente)
37 d, c, b, a = cs.c
38 n = len(x) - 1
39
40 print("\nCoeficientes da Spline Cúbica Natural (a_i, b_i, c_i, d_i):\n")
41
42 # 4. TABELA FORMATADA
43 tabela = "Intervalo | [x_i, x_{i+1}] | a_i | b_i | c_i | d_i\n"
44 tabela += "---|---|---|---|---|---\n"
45
46 for i in range(n):
47     tabela += (
48         f"{i+1} | [{x[i]:.5f}, {x[i+1]:.5f}] | "
49         f"{a[i]:.6f} | {b[i]:.6f} | {c[i]:.6f} | {d[i]:.6f}\n"
50     )
51
52 print(tabela)
53
54 # 5. EXEMPLO DO 1º POLINÔMIO S_1(x)
55 x0 = x[0]
56 ex1 = (
57     f"S_1(x) = {a[0]:.6f}"
58     f" + {b[0]:.6f}(x - {x0:.5f})"
59     f" + {c[0]:.6f}(x - {x0:.5f})^2"
60     f" + {d[0]:.6f}(x - {x0:.5f})^3"
61 )
62
63 print("\nExemplo do polinômio no 1º intervalo:\n")
64 print(ex1)
65
66 # 6. PLOTAGEM DA SPLINE
67 x_plot = np.linspace(x.min(), x.max(), 1000)
68 y_plot = cs(x_plot)

```

```
69
70 plt.figure(figsize=(9, 6))
71 plt.plot(x_plot, y_plot, label="Spline Cúbica Natural", linewidth=2)
72 plt.scatter(x, y, color="red", s=40, label="Pontos Originais")
73
74 plt.xlabel("x")
75 plt.ylabel("y")
76 plt.legend()
77 plt.grid(True)
78 plt.axis("equal")
79 plt.xlim(1.00, 1.60)
80 plt.ylim(1.20, 1.50)
81 plt.show()
```