Estabilidade numérica de fórmulas baricêntricas para interpolação

André Pierro de Camargo

Tese apresentada
ao
Instituto de Matemática e Estatística
da
Universidade de São Paulo
para
obtenção do título
de
Doutor em Ciências

Programa: Doutorado em Matemática Aplicada
Orientador: Walter Figueiredo Mascarenhas

Durante o desenvolvimento deste trabalho o autor recebeu auxílio financeiro do CNPq

São Paulo, dezembro de 2015

Estabilidade numérica de fórmulas baricêntricas para interpolação

Esta versão da tese contém as correções e alterações sugeridas pela Comissão Julgadora durante a defesa da versão original do trabalho, realizada em 15/12/2015. Uma cópia da versão original está disponível no Instituto de Matemática e Estatística da Universidade de São Paulo.

Comissão Julgadora:

- Prof. Dr. Walter Figueiredo Mascarenhas (orientador) - IME-USP
- Prof. Dr. Saulo Rabello Maciel de Barros - IME-USP
- Prof. Dr. Álvaro Rodolfo De Pierro - UNICAMP
- Prof. Dr. José Alberto Cuminato - ICMC-USP (São Carlos)
- Alagacone Sri Ranga - UNESP (São José do Rio Preto)

Agradecimentos

Agradeço aos meu pais, que sempre me apoiaram e me deram suporte em todas as minhas escolhas pessoais e profissionais. Agradeço à minha esposa Roberta e à minha filha Marina por todo o apoio e carinho recebidos nesse período. Agradeço a todos os meus mentores que tornaram o processo de aprendizagem mais interessante e me estimularam, direta ou indiretamente, a prosseguir nessa jornada superando os obstáculos. Agradeço, em particular, aos meus ex-orientadores de iniciação científica: Eduardo do Nascimento Marcos e Paulo Agozzini Martin, ao professor Antônio de Padua Franco Filho e ao meu orientador de doutorado Walter Figueiredo Mascarenhas. Agradeço a todos os meus amigos pelas diversas formas de apoio. Em especial agradeço ao amigo Leandro Cândido Batista e aos amigos Pedro da Silva Peixoto e Tiago De Morais Montanher pela ajuda com linux e C++, dentre outras coisas. Agradeço, também, ao CNPq pelo auxílio financeiro. Por fim, agradeço a todos os que não foram citados explicitamente mas que contribuíram, de alguma forma, para que eu pudesse completar mais essa etapa da minha carreira.

A todos vocês, muito obrigado!

Resumo

Camargo, A. P. **Estabilidade numérica de fórmulas baricêntricas para interpolação**. 2015. 120 f. Tese (Doutorado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2015.

O problema de reconstruir uma função $f : [a, b] \longrightarrow \mathbb{R}$ a partir de um número finito de valores conhecidos $f(x_0), f(x_1), \ldots, f(x_n)$ aparece com frequência em modelagem matemática. Em geral, não é possível determinar f completamente a partir de $f(x_0), f(x_1), \ldots, f(x_n)$, mas, em muitos casos de interesse, podemos encontrar aproximações razoáveis para f usando interpolação, que consiste em determinar uma função (um polinômio, ou uma função racional ou trigonométrica, etc) g que satisfaça

$$g(x_i) = f(x_i), \ i = 0, 1, \ldots, n.$$

Na prática, a função interpoladora g é avaliada em precisão finita e o valor final computado $\widehat{g(x)}$ pode diferir do valor exato $g(x)$ devido a erros de arredondamento. Essa diferença pode, inclusive, ultrapassar o erro de interpolação $E(x) = f(x) - g(x)$ em várias ordens de magnitude, comprometendo todo o processo de aproximação. A estabilidade numérica de um algoritmo reflete sua sensibilidade em relação a erros de arredondamento. Neste trabalho apresentamos uma análise detalhada da estabilidade numérica de alguns algoritmos utilizados no cálculo de interpoladores polinomiais ou racionais que podem ser postos na forma baricêntrica.

Os principais resultados deste trabalho também estão disponíveis em lingua inglesa nos artigos

- Mascarenhas, W e **Camargo, A. P.**, On the backward stability of the second barycentric formula for interpolation, *Dolomites research notes on approximation* v. 7 (2014) pp. 1–12.
- **Camargo, A. P.**, On the numerical stability of Floater-Hormann's rational interpolant, *Numerical Algorithms*, DOI 10.1007/s11075-015-0037-z.
- **Camargo, A. P.**, Erratum: "On the numerical stability of Floater-Hormann's rational interpolant", *Numerical Algorithms*, DOI 10.1007/s11075-015-0071-x.
- **Camargo, A. P.** e Mascarenhas, W., The stability of extended Floater-Hormann interpolants, *Numerische Mathematik*, submetido. arXiv:1409.2808v5

Palavras-chave: Interpolação, Fórmula baricêntrica, Estabilidade numérica.

Abstract

Camargo, A. P. **Numerical stability of barycentric formulae for interpolation**. 2015. 120 f. Tese (Doutorado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2015.

The problem of reconstructing a function $f : [a, b] \longrightarrow \mathbb{R}$ from a finite set of known values $f(x_0), f(x_1), \ldots, f(x_n)$ appears frequently in mathematical modeling. It is not possible, in general, to completely determine f from $f(x_0), f(x_1), \ldots, f(x_n)$ but, in several cases of interest, it is possible to find reasonable approximations for f by interpolation, which consists in finding a suitable function (a polynomial function, a rational or trigonometric function, etc.) g such that

$$g(x_i) = f(x_i), i = 0, 1, \ldots, n.$$

In practice, the interpolating function g is evaluated in finite precision and the final computed value $\widehat{g(x)}$ may differ from the exact value $g(x)$ due to rounding. In fact, such difference can even exceed the interpolation error $E(x) = f(x) - g(x)$ in several orders of magnitude, compromising the entire approximation process. The numerical stability of an algorithm reflect is sensibility with respect to rounding. In this work we present a detailed analysis of the numerical stability of some algorithms used to evaluate polynomial or rational interpolants which can be put in the barycentric format.

The main results of this work are also available in english in the papers

- Mascarenhas, W e **Camargo, A. P.**, On the backward stability of the second barycentric formula for interpolation, *Dolomites research notes on approximation* v. 7 (2014) pp. 1–12.
- **Camargo, A. P.**, On the numerical stability of Floater-Hormann's rational interpolant, *Numerical Algorithms*, DOI 10.1007/s11075-015-0037-z.
- **Camargo, A. P.**, Erratum: "On the numerical stability of Floater-Hormann's rational interpolant", *Numerical Algorithms*, DOI 10.1007/s11075-015-0071-x.
- **Camargo, A. P.** e Mascarenhas, W., The stability of extended Floater-Hormann interpolants, *Numerische Mathematik*, submited. arXiv:1409.2808v5

Keywords: Interpolation, Barycentric formulae, Numerical stability.

Sumário

Lista de Símbolos

$\mathbf{x}$	nós de interpolação
$\mathbf{y}$	valores interpolados
$\mathbf{w}$	pesos de interpolação
$n+1$	cardinalidade do conjunto dos pontos para interpolação
$f(\mathbf{x})$	imagem do conjunto $\mathbf{x}$ sob a função f
$\mathbf{x_{cheb1}}$	nós de Chebyshev do primeiro tipo
$\mathbf{x_{cheb2}}$	nós de Chebyshev do segundo tipo
$\mathbf{x_{eq}}$	nós igualmente espaçados
$\gamma(.)$	pesos de interpolação para o interpolador de Lagrange
d	parâmetro que define a ordem de aproximação dos interpoladores de Floater-Hormann
$\mu_d(.)$	pesos de interpolação para o interpolador de Floater-Hormann
$r_d(x, \mathbf{x}, \mathbf{y})$	interpolador de Floater-Hormann
$\tilde{r}_{d,\tilde{n},\tilde{d},\kappa}(x, \mathbf{x}, \mathbf{y})$	interpolador de Floater-Hormann estendido
$p(x, \mathbf{x}, \mathbf{y}, \mathbf{w})$	primeira Fórmula baricêntrica
$q(x, \mathbf{x}, \mathbf{y}, \mathbf{w})$	fórmula baricêntrica/segunda fórmula baricêntrica
$Leb(x, \mathbf{x}, \mathbf{y}, \mathbf{w}))$	função de Lebesgue
$\Lambda(\mathbf{x}, \mathbf{w})$	constante de Lebesgue
$C[a, b]$	espaço das funções contínuas no intervalo $[a, b]$
$\|\|.\|\|_\infty$	norma do Supremo
$\lfloor\ .\ \rfloor$	parte inteira
χ	homeomorfismo afim por partes
$\boldsymbol{\zeta}$	vetor de erros relativos nos pesos
$fl(.)$	operador de arredondamento
$cond(x, \mathbf{x}, \widehat{\mathbf{y}})$	número de condição
$\langle . \rangle$	contador de Stewart
ϵ	precisão da máquina

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

O problema de reconstruir uma função $f : [a, b] \longrightarrow \mathbb{R}$ a partir de um número finito de valores conhecidos $f(x_0), f(x_1), \ldots, f(x_n)$ aparece com frequência em modelagem matemática. Em geral, não é possível determinar f completamente a partir de $f(x_0), f(x_1), \ldots, f(x_n)$ mas, em muitos casos de interesse, é possível encontrar aproximações razoáveis para f usando interpolação, que consiste em determinar uma função (um polinômio, ou uma função racional ou trigonométrica, etc) g que satisfaça

$$g(x_i) = f(x_i), \ i = 0, 1, \ldots, n.$$

Na prática, a função interpoladora g é avaliada em precisão finita e o valor final computado $\widehat{g(x)}$ pode diferir do valor exato $g(x)$ devido a erros de arredondamento. Essa diferença pode, inclusive, ultrapassar o erro de interpolação $E(x) = f(x) - g(x)$ em várias ordens de magnitude, comprometendo todo o processo de aproximação. A estabilidade numérica de um algoritmo reflete a sua sensibilidade em relação a erros de arredondamento. Neste trabalho apresentamos uma análise detalhada da estabilidade numérica de alguns algoritmos utilizados no cálculo de interpoladores polinomiais e racionais que podem ser postos na forma baricêntrica

$$g(x) \quad = \quad \sum_{i=0}^{n} \tfrac{w_i}{x-x_i} f(x_i) \Big/ \sum_{i=0}^{n} \tfrac{w_i}{x-x_i}, \quad x \ \notin \ \{x_0, x_1, \ldots, x_n\},$$

como os interpoladores de Lagrange e de Floater-Hormann.

No Capítulo 2 faremos uma breve revisão sobre os principais conceitos relacionados à interpolação baricêntrica e também sobre os fundamentos da aritmética de ponto flutuante e no Capítulo 3 reunimos os principais resultados conhecidos sobre a convergência e estabilidade numérica dos Interpoladores de Lagrange (polinomial) e de Floater-Hormann (racional). As nossas contribuições são apresentadas nos Capítulos 4, 5 e 6.

No Capítulo 4 apresentamos uma análise da sensibilidade da fórmula baricêntrica genérica com relação à perturbações dos seus parâmetros: nós de interpolação, valores interpolados e pesos de interpolação. Mostramos, também, que a fórmula baricêntrica possui a propriedade de estabilidade backward quando a constante de Lebesgue associada aos nós de interpolação é pequena, isto é: que o valor computado da fórmula baricêntrica (em aritmética de precisão finita) corresponde ao valor exato da fórmula baricêntrica com parâmetros (nós, valores e pesos) perturbados. Esse resultado generaliza os resultados obtidos em [Mas14] para interpolação nos nós de Chebyshev do segundo tipo.

No Capítulo 5 apresentamos um algoritmo para calcular o interpolador de Floater-Hormann que possui a propriedade de estabilidade backward sobre toda a reta real. Esse algoritmo é baseado em uma generalização, para o interpolador de Floater-Hormann, da primeira fórmula baricêntrica para o interpolador polinomial de Lagrange. A nossa análise generaliza os resultados de [Hig04] sobre a estabilidade da fórmula baricêntrica para o interpolador polinomial de Lagrange. Também utilizamos os resultados do Capítulo 4 para fazer uma análise detalhada da estabilidade numérica

da fórmula baricêntrica para o interpolador de Floater-Hormann. Diversos experimentos numéricos são apresentados para consolidar os resultados teóricos.

No Capítulo 6 apresentamos uma análise da estabilidade numérica dos interpoladores de Floater-Hormann estendidos, os quais visam diminuir os efeitos causados pelos erros de arredondamento na avaliação numérica do interpolador de Floater-Hormann para nós igualmente espaçados. A nossa análise identifica diversos argumentos inconsistentes presentes na literatura corrente acerca da estabilidade numérica de tais interpoladores e mostra que esses interpoladores são bem menos estáveis do que se acreditava até então. O nosso resultado principal mostra que, em alguns casos, a constante de Lebesgue para o interpolador de Floater-Hormann estendido possui ordem de crescimento exponencial com relação ao parâmetro que define a sua ordem de aproximação (acreditava-se que a ordem de crescimento dessa constante de Lebesgue era logaritmica com relação a esse mesmo parâmetro.)

Capítulo 2

Preliminares

2.1 A fórmula baricêntrica

A fómula baricêntrica para interpolação do vetor real $\mathbf{y} = (y_0, y_1, \ldots, y_n)$, associada aos nós de interpolação $\mathbf{x} = (x_0, x_1, \ldots, x_n)$, $a \leq x_0 < x_1 < \ldots x_n \leq b$, e aos pesos $\mathbf{w} = (w_0, w_1, \ldots, w_n)$ é dada por

$$q(x, \mathbf{x}, \mathbf{y}, \mathbf{w}) \;=\; \begin{cases} \sum\limits_{i=0}^{n} \frac{w_i y_i}{x - x_i} \Big/ \sum\limits_{i=0}^{n} \frac{w_i}{x - x_i} \;, & \text{para } x \in [a,b]/\{x_0, x_1, \ldots, x_n\}. \\ y_i, & \text{para } x = x_i. \end{cases} \tag{2.1}$$

Eventualmente também escreveremos $\mathbf{x} = \mathbf{x}^n = (x_0^n, x_1^n, \ldots, x_n^n)$ para denotar que $\mathbf{x}$ depende de n. Durante o texto, assumiremos que os pesos de interpolação são não nulos, isto é

$$w_i \neq 0, \;\; i = 0, 1, \ldots, n.$$

Assumiremos, também, que

$$\sum_{i=0}^{n} \frac{w_i}{x - x_i} \;\neq\; 0, \;\; \forall\, x \in [a,b]/\{x_0, x_1, \ldots, x_n\}. \tag{2.2}$$

Dessa forma, a expressão (2.1) define uma função contínua em $[a,b]$ que interpola $\mathbf{y}$ em $\mathbf{x}$ e o operador linear

$$y \overset{\Phi}{\longmapsto} q(., \mathbf{x}, \mathbf{y}, \mathbf{w}) \in C[a,b], \tag{2.3}$$

entre os espaços normados $(\mathbb{R}^{n+1}, ||.||_\infty)$ e $(C[a,b], ||.||_\infty)$, está bem definido.

Quando os valores interpolados $\mathbf{y}$ provém de uma função contínua $f \in C[a,b]$, isto é,

$$y_i = f(x_i), \;\; i = 0, 1, \ldots, n,$$

a transformação Φ também pode ser interpretada como o operador linear

$$f \longmapsto q(., \mathbf{x}, f(\mathbf{x}), \mathbf{w}) \in C[a,b], \qquad f(\mathbf{x}) := (f(x_0), f(x_1), \ldots, f(x_n)). \tag{2.4}$$

É fácil ver que ambos os operadores definidos por (2.3) e (2.4) possuem a mesma norma

$$\Lambda(\mathbf{x}, \mathbf{w}) \;:=\; \max_{||\mathbf{y}||_\infty \leq 1} ||q(., \mathbf{x}, \mathbf{y}, \mathbf{w})||_\infty \;=\; \max_{||f||_\infty \leq 1} ||q(., \mathbf{x}, f(\mathbf{x}), \mathbf{w})||_\infty \tag{2.5}$$

com relação à norma do supremo $||.||_\infty$ definida em $C[a,b]$ e em $\mathbb{R}^{n+1}$. O número $\Lambda(\mathbf{x}, \mathbf{w})$ é historicamente denominado a constante de Lebesgue do interpolador (2.1) [BMHK12], [BMHS13], [MdC14]. Eventualmente também escreveremos $\Lambda(\mathbf{x}, \mathbf{w})|_a^b$ para explicitar o intervalo $[a,b]$ de referência em

questão. Note que a constante de Lebesgue também corresponde ao valor máximo da função de Lebesgue $Leb(x, \mathbf{x}, \mathbf{y}, \mathbf{w}) \;=$

$$\max_{||\mathbf{y}||_\infty \leq 1} |q(x, \mathbf{x}, \mathbf{y}, \mathbf{w})| \;=\; \begin{cases} 1, & \text{se } x \in \{x_0, x_1, \ldots, x_n\}, \\ \sum\limits_{i=0}^{n} \left|\frac{w_i}{x-x_i}\right| \Big/ \left|\sum\limits_{i=0}^{n} \frac{w_i}{x-x_i}\right|, & \text{caso contrário,} \end{cases} \tag{2.6}$$

pois

$$\max_{||\mathbf{y}||_\infty \leq 1} \left(\max_{x \in [a,b]} |q(x, \mathbf{x}, \mathbf{y}, \mathbf{w})| \right) \;=\; \max_{x \in [a,b]} \left(\max_{||\mathbf{y}||_\infty \leq 1} |q(x, \mathbf{x}, \mathbf{y}, \mathbf{w})| \right).$$

A constante de Lebesgue possui um papel fundamental no estudo da estabilidade numérica da fórmula baricêntrica (2.1), pois ela mede a sensibilidade do interpolador q com relação à perturbações nos valores interpolados $\mathbf{y}$. De fato, para $\mathbf{y}_1, \mathbf{y_2} \in \mathbb{R}^{n+1}$, temos

$$||q(., \mathbf{x}, \mathbf{y}_2, \mathbf{w}) - q(., \mathbf{x}, \mathbf{y_1}, \mathbf{w})||_\infty \leq \Lambda(\mathbf{x}, \mathbf{w}) \; ||\mathbf{y}_2 - \mathbf{y_1}||_\infty. \tag{2.7}$$

A constante de Lebesgue também é útil para medir a taxa de aproximação de $q(., \mathbf{x}, f(\mathbf{x}), \mathbf{w})$, para uma função particular f, em relação à melhor approximação $q(., \mathbf{x}, \mathbf{v}^*, \mathbf{w})$ para f dentro do subespaço $\Phi(\mathbb{R}^{n+1})$. De fato, segue direto de (2.7) que

$$\begin{aligned} ||f - q(., \mathbf{x}, f(\mathbf{x}), \mathbf{w})||_\infty &\leq ||f - q(., \mathbf{x}, \mathbf{v}^*, \mathbf{w})||_\infty \;+\; ||q(., \mathbf{x}, \mathbf{v}^*, \mathbf{w}) - q(., \mathbf{x}, f(\mathbf{x}), \mathbf{w})||_\infty \\ &\leq ||f - q(., \mathbf{x}, \mathbf{v}^*, \mathbf{w})||_\infty \;+\; \Lambda(\mathbf{x}, \mathbf{w})||\mathbf{v}^* - f(\mathbf{x})||_\infty \\ &\leq (1 + \Lambda(\mathbf{x}, \mathbf{w})) \; ||f - q(., \mathbf{x}, \mathbf{v}^*, \mathbf{w})||_\infty. \end{aligned} \tag{2.8}$$

Observação 1. *A desigualdade (2.8) vale, de fato, para qualquer aproximação* $q(., \mathbf{x}, \mathbf{v}, \mathbf{w}) \in \Phi(\mathbb{R}^{n+1})$.

Para os principais interpoladores considerados nesse trabalho (interpoladores de Lagrange e de Floater-Hormann), os pesos de interpolação $\mathbf{w}(\mathbf{x})$ são funções homogêneas de $\mathbf{x}$, no sentido de que

$$\mathbf{w}(\varphi(\mathbf{x})) = \alpha^k \mathbf{w}(\mathbf{x}), \qquad \varphi(\mathbf{x}) := (\varphi(x_0), \varphi(x_1), \ldots, \varphi(x_n)), \tag{2.9}$$

para toda função afim invertível $\varphi : [a, b] \to [c, d], \;\; x \overset{\varphi}{\longmapsto} \alpha x + \beta,$ onde k é um número inteiro não negativo que depende de $\mathbf{w}$, mas não depende de φ. Sob essa condição, para $g : [c, d] \longrightarrow \mathbb{R}$ e $\mathbf{x}' = \varphi(\mathbf{x})$, temos

$$\begin{aligned} q(u, \mathbf{x}', g(\mathbf{x}'), \mathbf{w}(\mathbf{x}')) &= \sum_{i=0}^{n} \frac{w_i' g(x_i')}{u - x_i'} \Big/ \sum_{i=0}^{n} \frac{w_i'}{u - x_i'} \;=\; \sum_{i=0}^{n} \frac{\alpha^k w_i g(x_i')}{\varphi(\varphi^{-1}(u)) - \varphi(x_i)} \Big/ \sum_{i=0}^{n} \frac{\alpha^k w_i}{\varphi(\varphi^{-1}(u)) - \varphi(x_i)} \\ &= \sum_{i=0}^{n} \frac{\alpha^k w_i g(x_i')}{\alpha(\varphi^{-1}(u) - x_i)} \Big/ \sum_{i=0}^{n} \frac{\alpha^k w_i}{\alpha(\varphi^{-1}(u) - x_i)} \;=\; \sum_{i=0}^{n} \frac{w_i g(x_i')}{\varphi^{-1}(u) - x_i} \Big/ \sum_{i=0}^{n} \frac{w_i}{\varphi^{-1}(u) - x_i} \\ &= q(\varphi^{-1}(u), \mathbf{x}, g(\varphi(\mathbf{x})), \mathbf{w}(\mathbf{x})). \end{aligned} \tag{2.10}$$

Portanto, sob (2.9), a fórmula baricêntrica (2.1) para $\mathbf{x}'$ e g é facilmente obtida compondo-se a fórmula baricêntrica para $\mathbf{x} = \varphi^{-1}(\mathbf{x}')$ e $(g \circ \varphi)$ com a transformação afim φ^{-1}. Por esse motivo, muitas vezes os resultados referentes à fórmula baricêntrica (2.1) são enunciados para os intervalos de interpolação padrão $[-1, 1]$ ou $[0, 1]$. Por exemplo, segue de (2.10) que $\Lambda(\varphi(\mathbf{x}), \mathbf{w}(\varphi(\mathbf{x})))|_c^d \;=$

$$\max_{||g||_\infty \leq 1} ||q(., \mathbf{x}', g(\mathbf{x}'), \mathbf{w}(\mathbf{x}'))||_\infty \;=\; \max_{||g \circ \varphi||_\infty \leq 1} ||q(., \mathbf{x}, (g \circ \varphi)(\mathbf{x}), \mathbf{w}(\mathbf{x}))||_\infty \;=\; \Lambda(\mathbf{x}, \mathbf{w}(\mathbf{x}))|_a^b,$$

pois $||g||_\infty \leq 1 \Longleftrightarrow ||g \circ \varphi||_\infty \leq 1$.

2.2 Derivadas de interpoladores na forma baricêntrica

Sob a hipótese (2.2), as singularidades $x_0, x_1, \ldots, x_n$ do interpolador (2.1) são removíveis e temos uma função analítica em um entorno da reta real[1]. As derivadas de $q(x, \mathbf{x}, \mathbf{y}, \mathbf{w})$ são lineares em $\mathbf{y}$:

$$\frac{\partial^k q(x, \mathbf{x}, \mathbf{y}, \mathbf{w})}{\partial x^k} = \sum_{i=0}^{n} \left(\frac{\partial^k q\left(x, \mathbf{x}, \mathbf{e^{(i)}}, \mathbf{w}\right)}{\partial x^k} \right) y_i, \tag{2.11}$$

com $\mathbf{e^{(i)}} = (0, \ldots, 0, \overbrace{1}^{\text{posição i}}, 0 \ldots, 0)$, $i = 0, 1, \ldots, n$.

Neste trabalho não estudamos as propriedades dessas derivadas, porém precisamos de seus valores nos nós de interpolação para definir os interpoladores de Floater-Horamnn estendidos no Capítulo 6. Baseados na Proposição 12 de [SW86], Klein e Berrut [KB12] apresentaram uma fórmula de recorrência simples[2] para calcular os coeficientes $D_{j,i}^{(k)} := \frac{\partial^k q\left(x_j, \mathbf{x}, \mathbf{e^{(i)}}, \mathbf{w}\right)}{\partial x^k}$ em (2.11):

$$D_{j,i}^{(0)} = \begin{cases} 1, & \text{se } i = j \\ 0, & \text{se } i \neq j \end{cases} \quad \text{e, para } \; k \geq 1, \quad D_{j,i}^{(k)} = \begin{cases} \frac{w_i}{w_j} D_{j,j}^{(k-1)} - D_{j,i}^{(k-1)}, & \text{se } i \neq j \\ -\sum\limits_{\ell \neq j} D_{j,\ell}^{(k)}, & \text{se } i = j. \end{cases} \tag{2.12}$$

Temos, portanto, que

$$\frac{\partial^k q(x_j, \mathbf{x}, \mathbf{y}, \mathbf{w})}{\partial x^k} = \sum_{i=0}^{n} D_{j,i}^{(k)} \; y_i. \tag{2.13}$$

2.3 Fundamentos da aritmética de ponto flutuante

Para compreender os efeitos dos erros de arredondamento na avaliação numérica de expressões matemáticas, é necessário conhecer as propriedades fundamentais do sistema de números de ponto flutuante. O nosso trabalho baseia-se no modelo padrão apresentado em [Hig02], sob o qual estão alicerçadas as aritméticas de ponto flutuante utilizadas por diversas linguagens de progração como C, C++, Java e Fortran, e também a maioria dos microprocessadores atuais. Nesse modelo, o sistema de números de ponto flutuante $\mathbb{F}$ é definido como um subconjunto da reta real cujos elementos

$$y = \pm m \times \beta^{e-t}$$

são caracterizados por 4 parâmetros inteiros

- a base β,
- a precisão t
- e o intervalo $e_{min} \leq e \leq e_{max}$ para o expoente.

A mantissa m é um número inteiro tal que $0 \leq m < \beta^t$ e $m \geq \beta^{t-1}$ para $y \neq 0$ e $e > e_{min}$. Essa última condição garante a unicidade da representação de cada número de ponto flutuante em $\mathbb{F}$. O tipo *double* (IEEE 754) definido em C,C++ e Java corresponde a $\beta = 2$, $t = 53$, $e_{min} = -1021$ e $e_{max} = 1024$ ([Hig02], p. 37.)

Para uma expressão matemática $expr = expr(\alpha_1, \alpha_2, \ldots, \alpha_k)$, função dos parâmetros reais $\alpha_1, \alpha_2, \ldots, \alpha_k$, denotaremos por $fl(expr)$, ou simplesmente $\widehat{expr}$, o elemento de $\mathbb{F}$ obtido pela avaliação numérica de $expr$ segundo as regras de arredondamento que definem a aritmética de ponto flutuante em questão. Por exemplo,

[1] Como a fórmula baricêntrica representa sempre um interpolador racional, então ela possui um número finito de pólos.

[2] Aqui estendemos a definição desses coeficientes também para $k = 0$.

$$fl(\alpha_1 + (\alpha_2 + \alpha_3)), \quad fl(fl(\alpha_1) + fl(\alpha_2 + \alpha_3)), \quad \text{e} \quad fl(fl(\alpha_1) + fl(fl(\alpha_2) + fl(\alpha_3)))$$

denotam o mesmo elemento de $\mathbb{F}$. Na terceira expressão acima, todos os arredondamentos na avaliação numérica de $\alpha_1 + (\alpha_2 + \alpha_3)$ estão explícitos e, nas duas primeiras, alguns estão implícitos.

Na nossa análise iremos considerar o modelo de aritmética de ponto flutuante padrão[3] descrito na página 40 de [Hig02]

$$fl(w \; op \; z) = (w \; op \; z)(1 + \delta), \quad |\delta| \leq \epsilon, \quad op = +, -, *, /, \quad \forall w, z \in \mathbb{F}, \tag{2.14}$$

onde $\epsilon = \beta^{1-t}$ denota a precisão do sistema $\mathbb{F}$, ou precisão da máquina (não confundir com o parâmetro precisão t.) Para a aritmética de precisão dupla (tipo *double*), por exemplo, temos $\epsilon = 2^{-52} \approx 2.22 \times 10^{-16}$.

A principal ferramenta para analisar a propagação de erros de arredondamento é o contador de erro de Stewart [Hig02]

$$\langle k \rangle \quad = \quad \prod_{i=1}^{k} (1 + \xi_i)^{\rho_i}, \quad \text{com } \rho_i = \pm 1 \quad \text{e } |\xi_i| \leq \epsilon. \tag{2.15}$$

Em vista de (2.15), podemos reescrever (2.14) como

$$fl(w \; op \; z) = (w \; op \; z)\langle 1 \rangle, \quad op = +, -, *, /. \tag{2.16}$$

É conveniente, também, escrever $\langle k \rangle_u$ quando desejamos indexar por u, k erros de arredondamento específicos. As propriedades básicas do contador de erro de Stewart podem ser resumidas por

$$\frac{1}{\langle k \rangle_u} = \langle k \rangle_w, \qquad \langle k_1 \rangle_{u_1} \langle k_2 \rangle_{u_2} = \langle k_1 + k_2 \rangle_v, \tag{2.17}$$

$$\text{se} \quad k_1 \leq k_2 \quad \text{então} \quad \langle k_1 \rangle_u = \langle k_2 \rangle_v \tag{2.18}$$

e (lema 3.1 de [Hig02])

$$\text{se} \quad k\epsilon < 0.001 \quad \text{então} \quad |\langle k \rangle - 1| \leq \frac{k\epsilon}{1 - k\epsilon} \leq 1.01 k\epsilon. \tag{2.19}$$

Por exemplo, suponha que desejamos avaliar o produto $(w-z)(r-s)$ numericamente, com $w, z, r, s \in \mathbb{F}$. Por (2.16), temos que

$$\begin{aligned} fl(w - z) &= (w - z)\langle 1 \rangle_1 \\ fl(r - s) &= (r - s)\langle 1 \rangle_2 \end{aligned}$$

e, portanto,

$$\begin{aligned} fl((w - z)(r - s)) &= fl((w - z)(r - s)\langle 1 \rangle_1 \langle 1 \rangle_2) \\ &\overset{(2.16)}{=} (w - z)(r - s)\langle 1 \rangle_1 \langle 1 \rangle_2 \langle 1 \rangle_3 \\ &\overset{(2.17)}{=} (w - z)(r - s)\langle 3 \rangle_1, \end{aligned}$$

com $|\langle 3 \rangle_1 - 1| \leq 3.03\epsilon$, se $3\epsilon < 0.001$. Em outras palavras, o erro relativo $\left(\frac{fl((w-z)(r-s))}{(w-z)(r-s)} - 1 \right)$ entre o valor computado e o valor exato de $(w - z)(r - s)$ possui valor absoluto menor ou igual a 3.03ϵ. No caso geral, é possível mostrar que, se $w_i, z_i, r_j, s_j \in \mathbb{F}, i \leq p, j \leq q$, então

[3] A equação (2.4) da página 40 de [Hig02] utiliza $u = \frac{1}{2}\epsilon$ ao invés de ϵ.

$$fl\left(\frac{\prod_{i=1}^{p}(w_i \pm z_i)}{\prod_{j=1}^{q}(r_j \pm s_j)}\right) = \left(\frac{\prod_{i=1}^{p}(w_i \pm z_i)}{\prod_{j=1}^{q}(r_j \pm s_j)}\right)\alpha, \text{ com } \alpha = \begin{cases} \langle 2p+2q-1\rangle & \text{se } p \geq 1 \text{ e } q \geq 1, \\ \langle 2p-1\rangle & \text{se } p \geq 1 \text{ e } q = 0, \\ \langle 2q\rangle & \text{se } p = 0 \text{ e } q \geq 1. \end{cases} \tag{2.20}$$

Por (2.17) e (2.18), também é fácil constatar que: se uma soma de $m+1$ números de ponto flutuante $a_0, a_1, \ldots, a_m$ é avaliada de forma recursiva $\left(\sum_{i=0}^{r+1} a_i = \left[\sum_{i=0}^{r} a_i\right] + a_{r+1}\right)$, então o valor final computado da soma satisfaz

$$fl\left(\sum_{i=0}^{m} a_i\right) = \sum_{i=0}^{m} a_i\langle m\rangle_i. \tag{2.21}$$

Por fim, vale que

$$fl\left(\sum_{i=0}^{m} a_i\langle k\rangle_i\right) = \left(\sum_{i=0}^{m} a_i\right)\langle m+k\rangle \tag{2.22}$$

sempre que $a_0\langle k\rangle_0, a_1\langle k\rangle_1, \ldots, a_m\langle k\rangle_m \in \mathbb{F}$ possuírem o mesmo sinal ($k \geq 0$). Nesse caso, (2.22) mostra que a soma $\sum_{i=0}^{m} a_i$ pode ser computada com erro relativo pequeno. A equação (2.22) segue de (2.21) e

$$\frac{1}{(1+\epsilon)^{m+k}} \leq \left(\min_i \langle m+k\rangle_i\right) \leq \frac{\sum_{i=0}^{m} a_i\langle m+k\rangle_i}{\sum_{i=0}^{m} a_i} \leq \left(\max_i \langle m+k\rangle_i\right) \leq (1+\epsilon)^{m+k}.$$

Capítulo 3

Interpoladores e a fórmula baricêntrica: exemplos

3.1 Interpolador polinomial de Lagrange

Dados $\mathbf{x}$ e $\mathbf{y}$, o interpolador polinomial de Lagrange é definido como o único polinômio $p_n(x)$, de grau menor ou igual a n, que satisfaz

$$p_n(x_i) = y_i, \ i = 0, 1, \ldots, n. \tag{3.1}$$

Explicitamente, temos

$$p_n(x) = \sum_{i=0}^{n} y_i \ell_i(x, \mathbf{x}), \qquad \ell_i(x, \mathbf{x}) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}, \qquad i = 0, 1, \ldots, n. \tag{3.2}$$

Os polinômios $\ell_i(x, \mathbf{x})$ são chamados os polinômios de Lagrange associados aos nós $\mathbf{x}$ e são caracterizados por

$$\ell_i(x_k, \mathbf{x}) = \delta_{i,k} \qquad \text{(delta de Kronecker.)} \tag{3.3}$$

Segue direto de (3.3) que os polinômios de Lagrange formam um conjunto linearmente independente, e, portanto, uma base do espaço vetorial $\mathbb{P}_n$ dos polinômios de grau menor ou igual a n. Isso destaca a unicidade do polinômio caracterizado por (3.1), pois $\mathbf{y}$ fornece os coeficientes do polinômio $p_n(x)$ com relação à base $\{\ell_0(x, \mathbf{x}), \ell_1(x, \mathbf{x}), \ldots, \ell_n(x, \mathbf{x})\}$.

3.1.1 Erro de interpolação

Para funções suaves, o erro de interpolação pode ser estimado com base no seguinte resultado clássico ([Hen82], p. 231 ou [IK94], p. 190):

Teorema 1. *Sejam $f : [a, b] \longrightarrow \mathbb{R}$ uma função de classe C^{n+1} e $p_n(x)$ o polinômio interpolador de Lagrange para f de modo que*

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

Então, dado $x \in [a, b]$ existe um número real ξ (com $\min\{x_0, x\} < \xi < \max\{x, x_n\}$) tal que

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega(x), \qquad \omega(x) := \prod_{i=0}^{n} (x - x_i). \tag{3.4}$$

Como corolário direto, segue que, se f é de classe C^∞ e possui todas as suas derivadas limitadas em $[a, b]$ por uma mesma constante (como as funções trigonométricas seno e cosseno, por exemplo),

então para qualquer sequência de malhas de interpolação $\mathbf{x}^1, \mathbf{x}^2 \ldots, \mathbf{x}^k, \ldots$ ($\mathbf{x}^k$ com $k+1$ pontos), a sequência de interpoladores de Lagrange associados $p_1(x), p_2(x), \ldots, p_k(x), \ldots$, converge uniformemente para f. Porém, para funções contínuas sem nenhum grau de diferenciabilidade, esse resultado é falso. De fato, Faber mostrou, em 1914, que, para qualquer sequência de malhas $\mathbf{x}^1, \mathbf{x}^2 \ldots, \mathbf{x}^k, \ldots$ contidas em $[a, b]$, existe uma função contínua $f^* : [a, b] \longrightarrow \mathbb{R}$ tal que a sequência de interpoladores de Lagrange associados $p_1(x), p_2(x), \ldots, p_k(x), \ldots$ não converge uniformemente[1] para f^* [Smi06].

3.1.2 A fórmula baricêntrica para o interpolador de Lagrange

Se $x \notin \{x_0, x_1, \ldots, x_n\}$, segue, por (3.2), que

$$p_n(x) = \sum_{i=0}^{n} y_i \left(\prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{x-x_j}{x_i-x_j} \right) \frac{x-x_i}{x-x_i} = \omega(x) \sum_{i=0}^{n} \frac{y_i \gamma(\mathbf{x})_i}{x-x_i} = p(x, \mathbf{x}, \mathbf{y}, \gamma(\mathbf{x})), \tag{3.5}$$

para $\omega(x)$ definido por (3.4) e

$$\gamma(\mathbf{x})_i := \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{1}{x_i - x_j} = \frac{1}{\omega'(x_i)}, \; i = 0, 1, \ldots, n. \tag{3.6}$$

A expressão (3.5) para o interpolador de Lagrange é comumente chamada de fórmula de Lagrange modificada, ou primeira fórmula baricêntrica. Aplicando (3.5) para o polinômio constante "1", obtemos

$$1 = \omega(x) \sum_{i=0}^{n} \frac{\gamma(\mathbf{x})_i}{x-x_i}. \tag{3.7}$$

Logo, por (3.5) e (3.7), segue que

$$p_n(x) = \sum_{i=0}^{n} \frac{\gamma(\mathbf{x})_i y_i}{x-x_i} \Bigg/ \sum_{i=0}^{n} \frac{\gamma(\mathbf{x})_i}{x-x_i} = q(x, \mathbf{x}, \mathbf{y}, \gamma(\mathbf{x})). \tag{3.8}$$

Temos, então, que o interpolador de Lagrange pode ser expresso em função da fórmula baricêntrica (2.1), para os pesos definidos por (3.6). A expressão (3.8) para o interpolador de Lagrange é também chamada de segunda fórmula baricêntrica ([Tre12], cap. 5, p. 34.)

3.1.3 Famílias especiais de nós para interpolação polinomial

Nós igualmente espaçados

Os conjunto de $(n+1)$ nós igualmente espaçados em $[a, b]$ é definido por

$$x_i = (x_{eq})_i = a + ih, \; i = 0, 1, \ldots, n, \qquad h := \frac{b-a}{n}. \tag{3.9}$$

Por meio de manipulações algébricas simples ([Hen82], p. 239), pode-se mostrar que, para essa familia de nós, os pesos de interpolação (3.6) satisfazem

$$\gamma(\mathbf{x_{eq}})_i = \frac{(-1)^{n-i}}{n! h^n} \binom{n}{i}. \tag{3.10}$$

Por um lado, nós igualmente espaçados são convenientes, pois possibilitam a discretização de operadores de altas ordens para a resolução de equações diferenciais ordinárias por diferenças finitas, por exemplo ([Lev07], cap. 1.) Por outro lado, aos nós igualmente espaçados está associado o chamado fenômeno de Runge, no qual a sequência de polinômios que interpolam a função

[1] De fato, vale que $\lim_{k \to} ||f^* - p_k(x)|| = \infty$.

$$f(x) = \frac{1}{1+x^2} \tag{3.11}$$

em nós igualmente espaçados em $[-5, 5]$ não converge uniformemente para f [Epp87], [IK94] p.275.

Nós de Chebyshev do primeiro tipo

Em vista da fórmula do erro de interpolação (3.4), uma estratégia para evitar esse fenômeno de divergência consiste em escolher nós de interpolação para os quais a norma do respectivo polinômio $\omega(x)$ seja mínima, isto é, encontrar a solução do seguinte problema de otimização

$$\min_{x_0,x_1,\ldots,x_n} \left(\max_{x \in [a,b]} |\omega(x)| \right) \quad s.a. \quad a \leq x_0 < x_1 < \ldots < x_n \leq b. \tag{3.12}$$

A solução do problema (3.12), para $[a, b] = [-1, 1]$, é dada pelas raízes do polinômio de Chebyshev do primeiro tipo $T_{n+1}(x)$:

$$\begin{cases} T_0(x) & = & 1, \quad T_1(x) \ = \ x, \\ T_{k+1}(x) & = & 2xT_k(x) - T_{k-1}(x), \quad k \geq 1. \end{cases}$$

Utilizando-se as identidades trigonométricas para a soma e subtração de arcos, é fácil mostrar (por indução) que

$$T_{n+1}(x) = \cos((n+1) * \arccos(x)) \ \forall \ x \ \in \ [-1, 1]. \tag{3.13}$$

Temos, então, que o polinômio mônico $\frac{1}{2^n}T_{n+1}(x)$ atinge seu módulo máximo, em $[-1, 1]$ (com sinais alternados), nos $n + 2$ pontos distintos $\cos(k\pi/(n + 1)), k = 0, 1, \ldots, n + 1$ e essa é uma condição suficiente (e também necessária) para ser solução do problema (3.12) ([IK94], p. 228 ou [Hen64], p. 194.) Logo, por (3.13), obtemos a solução de (3.12) analiticamente:

$$x_i = (x_{cheb1})_i = -\cos\left(\frac{(2i+1)\pi}{2(n+1)}\right), \ i = 0, 1, \ldots, n. \tag{3.14}$$

Os nós (3.14) são denominados os nós de Chebyshev do primeiro tipo. A solução geral de (3.12) é dada pela imagem dos nós de Chebyshev (3.14) pela transformação afim $\varphi : [-1, 1] \longrightarrow [a, b]$, $\varphi(x) = \frac{b-a}{2}x + \frac{b+a}{2}$.

Para os nós (3.14), os pesos de interpolação (3.6) também possuem uma forma analítica fechada ([Hen82], p. 249)[2], pois, nesse caso, temos $\omega(x) = \frac{1}{2^n}T_{n+1}(x) = \frac{1}{2^n}\cos((n + 1) * \text{arccoss}(x))$. Assim,

$$\omega'(x) \ = \ \tfrac{1}{2^n}T'_{n+1}(x) \ = \ -\tfrac{n+1}{2^n}\sin((n+1) * arccos(x))\tfrac{1}{\sqrt{(1-x^2)}}$$

e, portanto,

$$\gamma(\mathbf{x_{cheb1}})_i \overset{(3.6)}{=} \frac{1}{\omega'(x_i)} \ = \ (-1)^{n-i}\frac{2^n}{(n+1)}\sin\left(\frac{(2i+1)\pi}{2(n+1)}\right) \ i = 0, 1, \ldots, n. \tag{3.15}$$

Uma vantagem da fórmula baricêntrica (3.8) em relação à primeira fórmula baricêntrica (3.5) é a sua invariância sob dilatações dos pesos de interpolação por um mesmo fator. Logo, podemos utilizar os pesos simplificados

$$\gamma_i^* = (-1)^i \sin\left(\tfrac{(2i+1)\pi}{2(n+1)}\right), \ i \ = \ 0, 1, \ldots, n$$

em (3.8) ao invés dos pesos dados por (3.15).

[2] Na expressão para w_k^{-1} da página 249 de [Hen82] há um erro tipográfico. No lugar de 2^n, deveria estar escrito 2^{-n}.

Nós de Chebyshev do segundo tipo

Um outro conjunto interessante de nós são as raízes do polinômio $(1-x^2)U_{n-1}(x)$, onde $U_{n-1}(x)$ denota o polinômio de Chebyshev do segundo tipo:

$$\begin{cases} U_0(x) & := & 1, \quad U_1(x) \ := \ 2x, \\ U_{k+1}(x) & := & 2xU_k(x) - U_{k-1}(x), \quad k \geq 1. \end{cases}$$

Esses são os nós de Chebyshev do segundo tipo e são dados, explicitamente, por

$$x_i = (x_{cheb2})_i = -\cos\left(\frac{i\pi}{n}\right), \ i = 0, 1, \ldots, n. \tag{3.16}$$

A expressão trigonométrica para os polinômios de Chebyshev do segundo tipo é

$$U_{n-1}(x) = \frac{1}{\sqrt{(1-x^2)}} \sin(n * arccos(x)) \ \forall \ x \ \in \ [-1, 1]$$

e sua relação com os polinômios de Chebyshev do primeiro tipo se dá por

$$U_{n-1}(x) = \frac{1}{n} \, T_n'(x),$$

ou seja, os nós de Chebyshev do segundo tipo correspondem aos pontos de máximos e mínimos do polinômio $T_n(x)$ em $[-1, 1]$.

Em 1972, Salzer [Sal72] observou algumas boas propriedades dessa família de nós para interpolação. Por exemplo, o polinômio $\omega(x)$ (3.4) associado aos nós de Chebyshev do segundo tipo é quase ótimo, no sentido de que

$$\max_{x \ \in \ [-1,1]} \left(\prod_{i=0}^{n} (x - (x_{cheb2})_i) \right) \ \leq \ \frac{1}{2^{n-1}} \ = \ 2 \max_{x \ \in \ [-1,1]} \left(\prod_{i=0}^{n} (x - (x_{cheb1})_i) \right).$$

Além disso, para essa família de nós, os pesos (3.6) assumem uma forma extremamente simples:

$$\gamma(\mathbf{x_{cheb2}})_i = \frac{1}{n} \tau_i (-1)^{n-i} \, 2^{n-2}, \ i = 0, 1, \ldots, n,$$

com $\tau_0 = \tau_n = 1$ e $\tau_i = 2$, para $0 < i < n$. Novamente, podemos utilizar os pesos simplificados

$$\gamma_i^* = \tau_i (-1)^{n-i}, \ i = 0, 1, \ldots, n \tag{3.17}$$

em (3.8).

3.1.4 Constantes de Lebesgue

O famoso teorema de aproximação de Weierstrass [PQ08] afirma que, dada uma função contínua $f : [a, b] \longrightarrow \mathbb{R}$, existe uma sequência de polinômios $p_1^*(x), p_2^*(x), \ldots, p_k^*(x), \ldots$ (grau de $p_k^*(x) = k$) que converge uniformemente para f. Em particular, o Teorema de Jackson ([Pow81], p.196) afirma que, se f é Lipschitziana com constante de Lipschitz L, então para cada inteiro positivo k, existe um polinômio $\tilde{p}_k(x)$, de grau menor ou igual a k, tal que

$$\max_{x \ \in \ [a,b]} |f(x) - \tilde{p}_k(x)| \ = \ ||f - \tilde{p}_k||_\infty \ \leq \ \frac{L\pi}{4(k+1)}. \tag{3.18}$$

Como vale

$$||f - q(., \mathbf{x}, \mathbf{y}, \gamma(\mathbf{x}))||_\infty \ = \ ||f - p_n||_\infty \ \overset{(2.8)}{\leq} \ (1 + \Lambda(\mathbf{x}, \gamma(\mathbf{w}))) \, ||f - \tilde{p}_n||_\infty,$$

temos que, para uma sequência fixa de malhas de interpolação $\mathbf{x}^1, \mathbf{x}^2 \ldots, \mathbf{x}^k, \ldots$ ($\mathbf{x}^k$ com $k+1$ pontos), a sequência de polinômios interpoladores de Lagrange associados $p_1(x), p_2(x), \ldots, p_k(x), \ldots$ converge uniformemente para a função Lipschitziana f se

$$\lim_{k \longrightarrow \infty} \frac{\Lambda\left(\mathbf{x}^k, \gamma\left(\mathbf{x}^k\right)\right)}{k} = 0. \tag{3.19}$$

Observe que a fórmula (3.2) para o interpolador polinomial de Lagrange fornece a seguinte expressão para a constante de Lebesgue (2.5)

$$\Lambda(\mathbf{x}, \gamma(\mathbf{x})) = \max_{||\mathbf{y}||_\infty \leq 1} ||q(., \mathbf{x}, \mathbf{y}, \gamma(\mathbf{x}))||_\infty = \max_{x \in [a,b]} \left(\sum_{i=0}^{n} |\ell_i(x, \mathbf{x})| \right). \tag{3.20}$$

A divergência dos polinômios interpoladores associado ao fenômeno de Runge descrito na Seção 3.1.3 mostra que a condição (3.19) é falsa para a família de nós igualmente espaçados (3.9). De fato, temos que a sequência de constantes de Lebesgue associada aos nós igualmente espaçados possui comportamento assintótico exponencial. Mais precisamente, vale que [TW91]

$$\frac{2^{n-2}}{n^2} < \Lambda\left(\mathbf{x_{eq}^n}, \gamma\left(\mathbf{x_{eq}^n}\right)\right) < \frac{2^{n+3}}{n}. \tag{3.21}$$

O limitante inferior para (3.21) pode ser obtido calculando-se a função de Lebesgue correspondente no ponto $\breve{x} = a + h/2 = x_0 + h/2$. De fato, para $n \geq 2$, temos

$$\begin{aligned}
\Lambda\left(\mathbf{x_{eq}^n}, \gamma\left(\mathbf{x_{eq}^n}\right)\right) &\overset{(3.20)}{\geq} \sum_{i=0}^{n} \left|\ell_i\left(\breve{x}, \mathbf{x_{eq}^n}\right)\right| = \sum_{i=0}^{n} \left|\gamma\left(\mathbf{x_{eq}^n}\right)_i\right| \left| \prod_{\substack{j=0 \\ j \neq i}}^{n} \left(\breve{x} - (x_{eq}^n)_j\right) \right| \\
&\overset{(3.9),(3.10)}{=} \sum_{i=0}^{n} \tfrac{1}{n! h^n} \binom{n}{i} \left| \prod_{\substack{j=0 \\ j \neq i}}^{n} \left([a + h/2] - [a + jh]\right) \right| \\
&= \sum_{i=0}^{n} \binom{n}{i} \tfrac{1}{n!} \prod_{\substack{j=0 \\ j \neq i}}^{n} \left|j - \tfrac{1}{2}\right| \geq \sum_{i=0}^{n} \tfrac{1}{n!} \binom{n}{i} \tfrac{1}{4} \prod_{\substack{j=2 \\ j \neq i}}^{n} |j - 1| \\
&= \tfrac{1}{4n} \sum_{i=0}^{n} \binom{n}{i} \tfrac{1}{\max\{1, i-1\}} > \tfrac{1}{4n^2} \sum_{i=0}^{n} \binom{n}{i} = \tfrac{2^{n-2}}{n^2}.
\end{aligned}$$

O limitante superior em (3.21) pode ser obtido de forma análoga.

Para os nós de Chebyshev (do primeiro e segundo tipo), o cenário é oposto e a constante de Lebesgue cresce logaritmicamente em função do número de nós. Mais precisamente, para $n \geq 2$, vale que (ver [Gün80] e [MP73])

$$\Lambda\left(\mathbf{x_{cheb2}^n}, \gamma\left(\mathbf{x_{cheb2}^n}\right)\right) \leq \Lambda\left(\mathbf{x_{cheb1}^n}, \gamma\left(\mathbf{x_{cheb1}^n}\right)\right) < \frac{2}{\pi} \log(n) + 1.01. \tag{3.22}$$

Assim, por (3.19), a sequências de interpoladores de Lagrange associadas aos nós de Chebyshev convergem uniformemente para f, contanto que f seja Lipschitziana.

Erdös [Erd64] mostrou que, dada uma sequência de malhas $\mathbf{x}^1, \mathbf{x}^2 \ldots, \mathbf{x}^k, \ldots$ ($\mathbf{x}^k$ com $k+1$ pontos), existe uma constante positiva c tal que

$$\Lambda\left(\mathbf{x^k}, \gamma\left(\mathbf{x^k}\right)\right) \geq \tfrac{2}{\pi} \log(k) - c, \ \forall\ k \geq 1.$$

Logo, em vista de (3.22), temos que os nós de Chebyshev são quase ótimos em termos da constante de Lebesgue. Porém, o problema de encontrar, para cada inteiro positivo k, a malha $(\mathbf{x^k})^*$ que minimiza a constante de Lebesgue (3.20) para o interpolador de Lagrange, segue sem solução analítica. Esse

problema possui solução única se $x_0 = a$ e $x_n = b$ (extremos fixos), ou se assumirmos que a função de Lebesgue (2.6) correspondente assume o seu valor máximo (a constante de Lebesgue) também nos extremos do intervalo $[a, b]$. No primeiro caso, Bernstein e Erdös conjecturaram que a solução ótima deveria ser tal que todos os máximos locais da função de Lebesgue correspondente deveriam possuir o mesmo valor, isto é,

$$\lambda_1^* = \lambda_2^* = \ldots = \lambda_k^*, \tag{3.23}$$

$$\text{com} \quad x_{i-1} < \lambda_i^* < x_i, \quad \lambda_i^* \;=\; Leb\left(\tau_i^*, \mathbf{x^k}, \gamma(\mathbf{x^k})\right) \quad \text{e} \quad \frac{\partial Leb\left(\tau_i^*, \mathbf{x^k}, \gamma(\mathbf{x^k})\right)}{\partial x} \;=\; 0. \tag{3.24}$$

Essa conjectura foi posteriormente provada por Kilgore [Kil77], [Kil78] e por De Boor e Pinkus [BP78].

Para fins práticos, a quase otimalidade dos nós de Chebyshev (do primeiro e segundo tipos) já é suficiente e não há necessidade de se determinar $(\mathbf{x^k})^*$. Porém, para fins teóricos, aproximações de $(\mathbf{x^k})^*$, obtidas numericamente, poderiam ser de algum auxílio para indicar uma possível forma analítica (se é que há) para $(\mathbf{x^k})^*$. Nesse contexto, os resultados de Kilgore e de De Boor e Pinkus fornecem um método prático para determinar $(\mathbf{x^k})^*$ numericamente, pois, em vista de (3.23), o problema de encontrar os nós os que minimizam a constante de Lebesgue pode ser reescrito como

$$\min_{a=x_0<x_1\ldots<x_k=b} \Psi(x_0, x_1, \ldots, x_k), \qquad \Psi(x_0, x_1, \ldots, x_k) \;:=\; \sum_{i=1}^{k-1} \left(\lambda_i^* - \lambda_{i+1}^*\right)^2$$

e o gradiente de Ψ é facilmente obtido em função das derivadas parciais de $\lambda_1^*, \lambda_2^*, \ldots \lambda_k^*$:

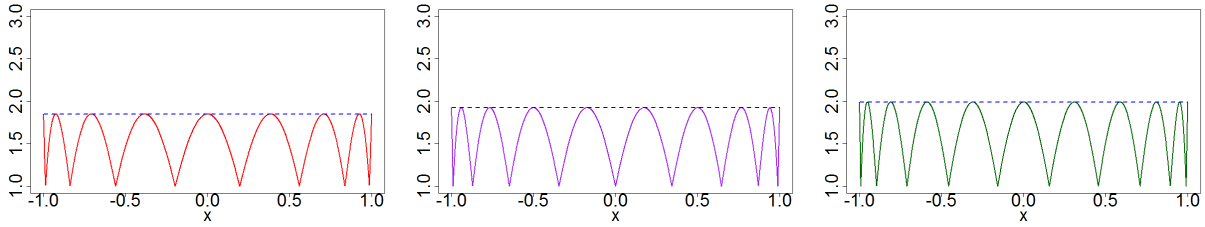
$$\frac{\partial \lambda_i^*}{\partial x_j} \overset{(3.24)}{=} \cancelto{0}{\frac{\partial Leb(\tau_i^*, \mathbf{x^k}, \gamma(\mathbf{x^k}))}{\partial x}} \frac{\partial \tau_i^*}{\partial x_j} + \frac{\partial Leb(\tau_i^*, \mathbf{x^k}, \gamma(\mathbf{x^k}))}{\partial x_j}.$$

Assim, $(\mathbf{x^k})^*$ pode ser obtido numericamente pela aplicação de algum método de otimização baseado em busca sobre linhas (*line search methods*) [NW99], como o método de máxima descida ou métodos de Newton ou quasi-Newton, etc.

Na Tabela 3.1 exibimos os valores de $\widehat{(\mathbf{x^7})^*}$, $\widehat{(\mathbf{x^8})^*}$ e $\widehat{(\mathbf{x^9})^*}$ para $[a, b] = [-1, 1]$ (sem extremos fixos), obtidos numericamente com o auxílio da rotina *constrOptim* (método = BFGS) implementada no pacote stats para o programa R para computação estatística. Na Figura 3.1 vemos a confirmação visual da Conjectura de Bernstein/Erdös para $\widehat{(\mathbf{x^7})^*}$, $\widehat{(\mathbf{x^8})^*}$ e $\widehat{(\mathbf{x^9})^*}$. Em contrapartida, a Figura 3.2 mostra a função de Lebesgue para os nós de Chebyshev (primeiro e segundo tipos) e igualmente espaçados), para $n = 9$.

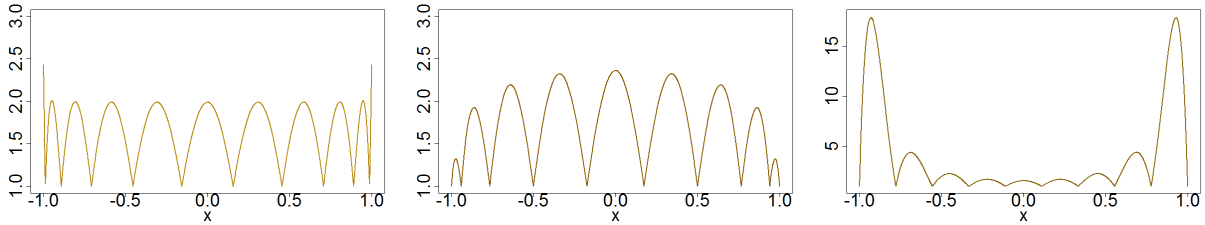
$(\mathbf{x^7})^*$	$(\mathbf{x^8})^*$	$(\mathbf{x^9})^*$
-0.9865454955855157	-0.9891036618765031	-0.9910000713804621
-0.8374508094897234	-0.8706395597205336	-0.8946461009573271
-0.5597961051979238	-0.6464123573168064	-0.7101643870420273
-0.1966064122640150	-0.3439983562775071	-0.4560085103542909
0.1966064122640154	0.0000000000003631	-0.1571382123669736
0.5597961051979242	0.3439983562781299	0.1571382124291026
0.8374508094897233	0.6464123573172054	0.4560085104135425
0.9865454955855156	0.8706395597207011	0.7101643870778138
	0.9891036618765271	0.8946461009711987
		0.9910000713819509

Tabela 3.1: *Malhas otimizadas em* $[-1, 1]$ *sem extremos fixos*



(a) $k = 7$ (b) $k = 8$ (c) $k = 9$

Figura 3.1: *Funções de Lebesgue associadas aos nós* $\widehat{(\mathbf{x}^7)^*}, \widehat{(\mathbf{x}^8)^*}$ *e* $\widehat{(\mathbf{x}^9)^*}$.



(a) $\mathbf{x}^{\mathbf{9}}_{\mathbf{cheb1}}$ (b) $\mathbf{x}^{\mathbf{9}}_{\mathbf{cheb2}}$ (c) $\mathbf{x}^{\mathbf{9}}_{\mathbf{eq}}$

Figura 3.2: *Funções de Lebesgue ($n = 9$) para: nós de Chebyshev do primeiro tipo (a), nós de Chebyshev do segundo tipo (b) e nós igualmente espaçados (c).*

3.1.5 A influência da constante de Lebesgue na etapa numérica de aproximação

Na seção anterior discutimos diversas propriedades teóricas da constante de Lebesgue para o interpolador de Lagrange, porém a sua relevância na etapa numérica de aproximação ainda não foi devidamente apresentada. Para esse fim, faremos a seguir uma discussão informal[3] sobre a estabilidade numérica da primeira fórmula baricêntrica (3.5) para o interpolador de Lagrange.

Para compreender melhor o processo de aproximação na prática, é conveniente decompor o erro total de interpolação $E_T(x) = E_A(x) + E_N(x)$ como a soma do erro de aproximação $E_A(x) = f(x) - p(x, \mathbf{x}, \mathbf{y}, \gamma(\mathbf{x}))$, $\mathbf{y} = f(\mathbf{x})$, e do o erro numérico $E_N(x) = p(x, \mathbf{x}, \mathbf{y}, \gamma(\mathbf{x})) - fl(p(x, \mathbf{x}, \mathbf{y}, \gamma(\mathbf{x})))$ oriundo da avaliação de $p(x, \mathbf{x}, \mathbf{y}, \gamma(\mathbf{x}))$ em precisão finita. Por outro lado, conforme explicaremos com mais detalhe no Capítulo 4, também é conveniente decompor o processo de avaliação numérica em diversas etapas. Uma possível decomposição em duas etapas é apresentada a seguir

- Na primeira etapa: $\mathbf{x}, \mathbf{y} \longmapsto \hat{\mathbf{x}}, \hat{\mathbf{y}}$, os nós e valores interpolados são avaliados em precisão finita.
- Na segunda etapa: $p\left(x, \hat{\mathbf{x}}, \hat{\mathbf{y}}, \gamma(\hat{\mathbf{x}})\right) \longmapsto fl\left(p\left(x, \hat{\mathbf{x}}, \hat{\mathbf{y}}, \gamma(\hat{\mathbf{x}})\right)\right)$, os valores arredondados $\hat{\mathbf{x}}$ e $\hat{\mathbf{y}}$ são utilizados por algum algoritmo para produzir o valor final computado de $p(x, \mathbf{x}, \mathbf{y}, \gamma(\mathbf{x}))$.

É possível mostrar que, para alguns algoritmos (por exemplo os algoritmos do Tipo I que serão apresentados no Capítulo 5), o erro introduzido na segunda etapa é, no máximo, proporcional ao produto $n\epsilon\Lambda\left(\hat{\mathbf{x}}, \gamma\left(\hat{\mathbf{x}}\right)\right)||\mathbf{y}||_\infty$. Além disso, se os nós de interpolação já são arredondados, isto é $\hat{\mathbf{x}} = \mathbf{x}$, então o erro introduzido na primeira etapa satisfaz

$$\begin{aligned} |p(x, \mathbf{x}, \hat{\mathbf{y}}, \gamma(\mathbf{x})) - p(x, \mathbf{x}, \mathbf{y}, \gamma(\mathbf{x}))| &= |q(x, \mathbf{x}, \hat{\mathbf{y}}, \gamma(\mathbf{x})) - q(x, \mathbf{x}, \mathbf{y}, \gamma(\mathbf{x}))| \\ &\overset{(2.7)}{\leq} \Lambda\left(\mathbf{x}, \gamma\left(\mathbf{x}\right)\right)||\hat{\mathbf{y}} - \mathbf{y}||_\infty. \end{aligned}$$

[3] A análise detalhada da estabilidade numérica da primeira fórmula baricêntrica (3.5) segue como o caso particular $d = n$ da análise dos algoritmos do Tipo I que será apresentada no Capítulo 5.

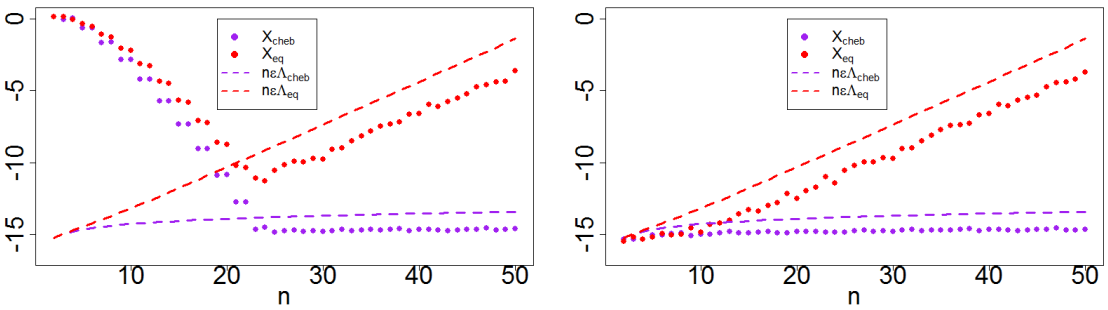
Dessa forma, temos que o erro numérico total para primeira fórmula baricêntrica para o interpolador de Lagrange está intimamente relacionado à constante de Lebesgue (o mesmo ocorre com a fórmula baricêntrica (2.1), como será mostrado no Capítulo 4.)

Para verificar como isso ocorre na prática, realizamos um experimento numérico relativo à interpolação da função $f(x) = \sin(x)$ no intervalo $[-5, 5]$, para as familias de nós igualmente espaçados $\mathbf{x_{eq}}$ em $[-5, 5]$ e nós de Chebyshev do segundo tipo $\mathbf{x_{cheb2,[-5,5]}}$ definidos em $[-5, 5]$, ou seja $\mathbf{x_{cheb2,[-5,5]}}$ é a imagem dos nós de Chebyshev do segundo tipo (3.16) sob a transformação afim $\varphi : [-1, 1] \longrightarrow [-5, 5]$ tal que $\varphi(-1) = -5$ e $\varphi(1) = 5$.

No gráfico (a) da Figura (3.3) temos o erro $\max\limits_{x \in [-5,5]} \log_{10} |f(x) - fl(p(x, \hat{\mathbf{x}}, \mathbf{y}, \gamma(\hat{\mathbf{x}})))|$ entre o valores[4] de $f(x)$ e os valores computados $fl(p(x, \hat{\mathbf{x}}, \mathbf{y}, \gamma(\hat{\mathbf{x}})))$ de $p(x, \hat{\mathbf{x}}, \mathbf{y}, \gamma(\hat{\mathbf{x}}))$ em precisão dupla ($\epsilon = 2^{-52} \approx 2.2 \times 10^{-16}$.) Como as derivadas da função seno são todas menores ou iguais a 1, em magnitude, segue de (3.4) que o erro de aproximação $E_A(x)$ converge uniformemente para zero conforme $n \longrightarrow \infty$. Em particular, para $n \geq 30$, vale que $|E_A(x)| < 10^{-16} < \epsilon$, para ambas as famílias de nós. Assim, os valores plotados para $n \geq 30$ correspondem, essencialmente, ao erro $E_N(x)$ oriundo da avaliação de $p(x, \mathbf{x}, \mathbf{y}, \gamma(\mathbf{x}))$ em precisão finita.

No gráfico (b) da Figura (3.3), temos o erro $\log_{10} \max\limits_{x \in [-5,5]} |p(x, \hat{\mathbf{x}}, \hat{\mathbf{y}}, \gamma(\hat{\mathbf{x}})) - fl\,(p(x, \hat{\mathbf{x}}, \hat{\mathbf{y}}, \gamma(\hat{\mathbf{x}})))\,|$ relativo à segunda etapa da decomposição do erro numérico descrita acima (os valores relativos à primeira etapa são similares). Os valores $p(x, \hat{\mathbf{x}}, \hat{\mathbf{y}}, \gamma(\hat{\mathbf{x}}))$ foram obtidos avaliando-se a fórmula (3.5) com precisão de 50 casas decimais, com o auxílio da biblioteca de precisão múltipla MPRF, e depois arredondando o resultado obtido para precisão dupla.

Em ambos os casos, podemos observar que o valor $n\epsilon\Lambda\,(\mathbf{x}, \gamma\,(\mathbf{x}))\,||f||_\infty$ fornece a ordem correta para a magnitude do erro numérico. Em particular, vemos que o erro numérico associado aos nós igualmente espaçados possui crescimento exponencial em função de n. Além do fenômeno de Runge descrito na seção 3.1.3, esse e outros fenômenos similares de instabilidade numérica contribuem para a má reputação dos nós igualmente espaçados para interpolação de Lagrange.



(a) (b)

Figura 3.3: *Erro total de interpolação* $\log_{10} \max\limits_{x \in [-5,5]} |f(x) - fl\,(p(x, \mathbf{x}, \mathbf{y}, \gamma(\mathbf{x})))\,|$ *(a) e erro na segunda etapa* $\log_{10} \max\limits_{x \in [-5,5]} |p(x, \hat{\mathbf{x}}, \hat{\mathbf{y}}, \gamma(\hat{\mathbf{x}})) - fl\,(p(x, \hat{\mathbf{x}}, \hat{\mathbf{y}}, \gamma(\hat{\mathbf{x}})))\,|$ *(b) para a função* $f(x) = \sin(x)$*, para os nós de Chebyshev do segundo tipo* $\left(\mathbf{x} = \mathbf{x}^{(\mathbf{n})}_{\mathbf{cheb2}}\right)$ *e nós igualmente espaçados* $\left(\mathbf{x} = \mathbf{x}^{(\mathbf{n})}_{\mathbf{eq}}\right)$*, ambos definidos em* $[-5, 5]$*. As linhas tracejadas indicam os valores do produto* $n\epsilon\Lambda\,(\hat{\mathbf{x}}, \gamma\,(\hat{\mathbf{x}}))$*, em cada caso.*

3.2 Interpolador de Floater-Hormann

Vimos nas Seções 3.1.3 e 3.1.5 que a interpolação de Lagrange para nós igualmente espaçados é problemática, tanto do ponto de vista de aproximação, quanto do ponto de vista da estabilidade

[4] Na verdade, os valores de $f(x)$ plotados também são arredondados, mas desprezamos essa diferença pois o erro de arredondamento é da ordem da precisão da máquina para a função considerada.

numérica. Uma alternativa para contornar esse inconveniente é utilizar intepoladores que possuam ordem de convergência baixa, mas que garantam convergência uniforme conforme o espaçamento entre dois nós de interpolação consecutivos tenda a zero, como Splines ([Hen82], cap. 5). Uma outra alternativa seria o uso direto de interpoladores polinomiais locais de grau baixo.

Em 2007, Floater e Hormann [FH07] apresentaram uma maneira interessante de misturar esses interpoladores polinomiais locais (de grau, no máximo, d) para formar um interpolador racional global:

$$r_d(x,\mathbf{x},\mathbf{y}) := \frac{\sum_{i=0}^{n-d} \lambda_{i,d}(x,\mathbf{x}) p_{i,d}(x,\mathbf{x},\mathbf{y})}{\sum_{i=0}^{n-d} \lambda_{i,d}(x,\mathbf{x})}, \tag{3.25}$$

onde $p_{i,d}(x,\mathbf{x},\mathbf{y})$ denota o único polinômio de grau menor ou igual a d que interpola $(x_i, y_i), \ldots, (x_{i+d}, y_{i+d})$ e as funções ponderadoras $\lambda_{i,d}(x,\mathbf{x})$ são definidos por

$$\lambda_{i,d}(x,\mathbf{x}) := \frac{(-1)^i}{(x-x_i)\ldots(x-x_{i+d})}, \quad i = 0,1,\ldots,n-d. \tag{3.26}$$

Para $d = n$, o interpolador de Floater-Hormann (3.25) se reduz ao interpolador polinomial de Lagrange (3.2) e, para $d = 0$, (3.25) se reduz à fórmula comumente chamada de interpolador de Berrut [Ber98].

3.2.1 A fórmula baricêntrica para o interpolador de Floater-Hormann

Floater e Hormann mostraram que o interpolador (3.25) pode ser descrito pela fórmula baricêntrica (2.1) para os pesos de interpolação dados por

$$w_i = \mu_d(\mathbf{x})_i := \sum_{j=\max\{0,i-d\}}^{\min\{n-d,i\}} (-1)^j \prod_{\substack{\tau=j \\ \tau\neq i}}^{j+d} \frac{1}{x_i - x_\tau} = (-1)^{i-d} \sum_{j=\max\{0,i-d\}}^{\min\{n-d,i\}} \left| \prod_{\substack{\tau=j \\ \tau\neq i}}^{j+d} \frac{1}{x_i - x_\tau} \right|, \tag{3.27}$$

isto é,

$$r_d(x,\mathbf{x},\mathbf{y}) = q(x,\mathbf{x},\mathbf{y},\mu_d(\mathbf{x})). \tag{3.28}$$

Essa propriedade segue das identidades

$$\sum_{i=0}^{n} \frac{\mu_d(\mathbf{x})_i}{x-x_i} = \sum_{i=0}^{n-d} \lambda_{i,d}(x,\mathbf{x}), \tag{3.29}$$

e

$$\sum_{i=0}^{n} \frac{y_i \mu_d(\mathbf{x})_i}{x-x_i} = \sum_{i=0}^{n-d} \lambda_{i,d}(x,\mathbf{x}) p_{i,d}(x,\mathbf{x},\mathbf{y}), \tag{3.30}$$

as quais são obtidas ao exprimir cada $p_{i,d}(x,\mathbf{x},\mathbf{y})$ em função da primeira fórmula baricêntrica (3.5).

Na próxima seção veremos que

$$\sum_{i=0}^{n} \frac{\mu_d(\mathbf{x})_i}{x-x_i} \neq 0, \quad \forall\, x \in \mathbb{R}/\{x_0, x_1, \ldots, x_n\} \tag{3.31}$$

e, portanto, (3.28) define uma função analítica (em uma faixa aberta contendo reta real) que interpola $\mathbf{y}$ em $\mathbf{x}$.

3.2.2 Ausência de pólos reais

Floater e Hormann mostraram que, para cada valor real x, o lado direito de (3.29) pode ser escrito como uma soma de termos não nulos de mesmo sinal (o que prova (3.31).) No Capítulo 5 utilizaremos essa propriedade para definir os algoritmos do Tipo I e, devido a isso, a seguir discutimos essa propriedade com algum detalhe.

Sejam

$$\begin{cases} \xi_{-1}(x,\mathbf{x}) &= \lambda_{0,d}(x,\mathbf{x}), \\ \xi_i(x,\mathbf{x}) &= (-1)^i \left(\prod_{j=i+1}^{i+d} \frac{1}{x-x_j} \right) \frac{x_i - x_{i+d+1}}{(x-x_i)(x-x_{i+d+1})}, \quad 0 \le i \le n-d-1, \\ \xi_{n-d}(x,\mathbf{x}) &= \lambda_{n-d,d}(x,\mathbf{x}) \end{cases}$$

e $x_{-1} := -\infty$ e $x_{n+1} := \infty$. Observe que, para $0 \le i \le n-d-1$, $\xi_i(x,\mathbf{x})$ é apenas uma outra forma de escrever $\lambda_{i,d}(x,\mathbf{x}) + \lambda_{i+1,d}(x,\mathbf{x})$. Assim, para um dado valor real $x \notin \{x_0, x_1, \ldots, x_n\}$ tal que $x_k < x < x_{k+1}$, para algum $k \in \{-1, 0, \ldots, n\}$, escrevemos

$$\begin{aligned} \sum_{i=0}^{n-d} \lambda_i(x,\mathbf{x}) &= \delta_{k-d}\, \xi_{-1}(x,\mathbf{x}) + \sum_{0 \le k-d-2j-1 \le k-d} \xi_{k-d-2j-1}(x,\mathbf{x}) \\ &+ \sum_{\substack{k-d < i \le k \\ 0 \le i \le n-d}} \lambda_i(x,\mathbf{x}) + \sum_{k < k+1+2j \le n-d} \xi_{k+1+2j}(x,\mathbf{x}), \end{aligned} \tag{3.32}$$

com

$$\delta_j := \begin{cases} 1, & \text{se } j \text{ é par}, \\ 0, & \text{se } j \text{ é ímpar}, \end{cases}$$

e com a convenção de que somas sobre conjuntos vazios valem zero. Mostrar que todas as parcelas em (3.32) possuem o mesmo sinal é um exercício simples, porém um pouco tedioso. Por exemplo, tomando-se duas parcelas consecutivas na útilma somatória em (3.32), temos

$$\begin{aligned} \xi_{k+1+2j}(x,\mathbf{x}) &= \lambda_{k+1+2j}(x,\mathbf{x}) + \lambda_{k+1+2j+1}(x,\mathbf{x}) \\ &\overset{(3.26)}{=} (-1)^{k+1+2j+d+1} \left(|\lambda_{k+1+2j}(x,\mathbf{x})| - |\lambda_{k+1+2j+1}(x,\mathbf{x})| \right) \end{aligned} \tag{3.33}$$

e $\xi_{k+1+2(j+1)}(x,\mathbf{x}) =$

$$(-1)^{k+1+2(j+1)+d+1} \left(\left|\lambda_{k+1+2(j+1)}(x,\mathbf{x})\right| - \left|\lambda_{k+1+2(j+1)+1}(x,\mathbf{x})\right| \right). \tag{3.34}$$

Como as diferenças entre parenteses em (3.33) e (3.34) são positivas, segue que $\xi_{k+1+2j}(x,\mathbf{x})$ e $\xi_{k+1+2(j+1)}(x,\mathbf{x})$ possuem o mesmo sinal.

3.2.3 Erro de interpolação

Se a função interpolada f é suficientemente suave, então, para $d \ge 1$, a taxa de convergência de $r_d(x,\mathbf{x},\mathbf{y})$ para $f(x)$ é $O(h^{d+1})$, onde

$$h := h(\mathbf{x}) = \max_{0 \le i \le n-1} x_{i+1} - x_i.$$

Mais precisamente, vale que

Teorema 2 ([FH07]). *Suponha que $d \ge 1$ e que $f : [a,b] \longrightarrow \mathbb{R}$ é de classe $C^{d+2}([a,b])$. Se $n-d$ é ímpar, então*

$$||r_d(.,\mathbf{x},f(\mathbf{x})) - f||_\infty \leq h^{d+1}(b-a)\frac{||f^{(d+2)}||_\infty}{d+2}. \tag{3.35}$$

Se $n-d$ é par, então

$$||r_d(.,\mathbf{x},f(\mathbf{x})) - f||_\infty \leq h^{d+1}\left((b-a)\tfrac{||f^{(d+2)}||_\infty}{d+2} + \tfrac{||f^{(d+1)}||_\infty}{d+1}\right).$$

Portanto, se f é ao menos de classe $C^{d+2}([a,b])$, o Teorema 2 garante que, para $d \geq 1$ fixado, a sequência de interpoladores de Floater-Hormann associados a uma sequência de malhas $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^k, \ldots$ ($\mathbf{x}^k$ com $k+1$ pontos), converge uniformemente para f (com ordem de convergência $d+1$) contanto que

$$\lim_{k\to\infty} h\left(\mathbf{x}^k\right) = 0.$$

Em particular, vale a convergência para nós igualmente espaçados.

Para $d = 0$, a taxa de convergência é de ordem $O(h)$, contanto que a razão de espaçamento local

$$\beta(\mathbf{x}) := \max_{1\leq i\leq n-2} \min\left\{\tfrac{x_{i+1}-x_i}{x_i-x_{i-1}}, \tfrac{x_{i+1}-x_i}{x_{i+2}-x_{i+1}}\right\}$$

permaneça limitada quando $h \longrightarrow 0$.

Teorema 3 ([FH07]). *Suponha que $d = 0$ e que $f : [a,b] \longrightarrow \mathbb{R}$ é de classe $C^2([a,b])$. Se $n-d$ é ímpar, então*

$$||r_d(.,\mathbf{x},f(\mathbf{x})) - f||_\infty \leq h(1+\beta(\mathbf{x}))(b-a)\tfrac{||f''||_\infty}{2}.$$

Se $n-d$ é par, então

$$||r_d(.,\mathbf{x},f(\mathbf{x})) - f||_\infty \leq h(1+\beta(\mathbf{x}))\left((b-a)\tfrac{||f''||_\infty}{2} + ||f'||_\infty\right).$$

Manter d fixo é uma condição suficiente, porém não necessária, para garantir a convergência da sequência dos interpoladores de Floater-Hormann para uma função particular f. Para os nós igualmente espaçados, por exemplo, se f é de classe $C^\infty([a,b])$ e possui todas as suas derivadas limitadas por uma mesma constante, então a sequência dos interpoladores de Floater-Hormann $r_1(.,\mathbf{x_{eq}}^1, f(\mathbf{x_{eq}}^1)), r_2(.,\mathbf{x_{eq}}^2, f(\mathbf{x_{eq}}^2)), \ldots, r_k(.,\mathbf{x_{eq}}^k, f(\mathbf{x_{eq}}^k)), \ldots$ (mantendo-se $d = n$) converge uniformemente para f com erro da ordem de h^n, o qual é muito menor do que h^{d+1} para $d \ll n$. Não é simples, porém, determinar, para uma dada função f, qual relação entre d e n deve existir para garantir a convergência dos interpoladores quando $n \to \infty$. Um estudo mais detalhado sobre essa questão foi apresentado em [GK12] para os casos nos quais f é analítica.

3.2.4 Constantes de Lebesgue

Os resultados conhecidos sobre o comportamento assintótico da constante de Lebesgue (2.5) para o interpolador de Floater-Hormann restringem-se, essencialmente, à familia de nós igualmente espaçados. Bos, De Marchi, Hormann e Klein [BMHK12] mostraram que, para a família de nós igualmente espaçados (3.9), a constante de Lebesgue $\Lambda(\mathbf{x^n_{eq}}, \mu_d(\mathbf{x^n_{eq}}))$ associada ao interpolador de Floater-Hormann com parâmetro d possui crescimento logarítmico com relação à n e crescimento exponencial com relação à d:

$$\frac{1}{2^{d+2}}\begin{pmatrix} 2d+1 \\ d \end{pmatrix} \ln\left(\frac{n}{d}-1\right) \quad\leq\quad \Lambda(\mathbf{x^n_{eq}}, \mu_d(\mathbf{x^n_{eq}})) \quad\leq\quad 2^{d-1}(2+\ln(n)). \tag{3.36}$$

Posteriormente, Hormann, Klein and De Marchi [HKM12] mostraram que esse comportamento não é afetado quando os nós de interpolação são obtidos a partir de pequenas perturbações dos nós igualmente espaçados:

$$\frac{1}{2^{d+2}M^{d+1}}\begin{pmatrix} 2d+1 \\ d \end{pmatrix}\ln\left(\frac{n}{d}-1\right) \quad \leq \quad \Lambda(\mathbf{x^n}, \mu_d(\mathbf{x^n})) \quad \leq \quad 2^{d-1}M^d(2+M\ln(n)),$$

para

$$M := \frac{\max\limits_{0\leq i<n} \; x_{i+1}^n - x_i^n}{\min\limits_{0\leq i<n} \; x_{i+1}^n - x_i^n}.$$

Para outros tipos de nós, a análise da constante de Lebesgue em função do parâmetro d parece ser bastante intricada, pois a distribuição dos $d+1$ nós dos subconjuntos utilizados para calcular os interpoladores locais em (3.25) pode ser bastante sensível com relação ao parâmetro d. Por exemplo, para os nós de Chebyshev do segundo tipo (3.16), quando $d \ll n$, os $d+1$ pontos das malhas locais são quase igualmente espaçados e é razoável esperar que o comportamento da constante de Lebesgue $\Lambda\left(\mathbf{x^n_{cheb2}}, \mu_d(\mathbf{x^n_{cheb2}})\right)$ fosse similar ao comportamento da constante de Lebesgue polinomial para $d+1$ nós igualmente espaçados (que é da ordem de 2^d.) Por outro lado, para $d \approx n$, o comportamento da contante de Lebesgue $\Lambda\left(\mathbf{x^n_{cheb2}}, \mu_d(\mathbf{x^n_{cheb2}})\right)$ deve ser similar ao comportamento da constante de Lebesgue para o interpolador de Lagrange ($d = n$), a qual possui crescimento logaritmico em n, conforme visto em (3.22). A Figura 3.4 mostra que esse é, de fato, o comportamento esperado para essa constante de Lebesgue.

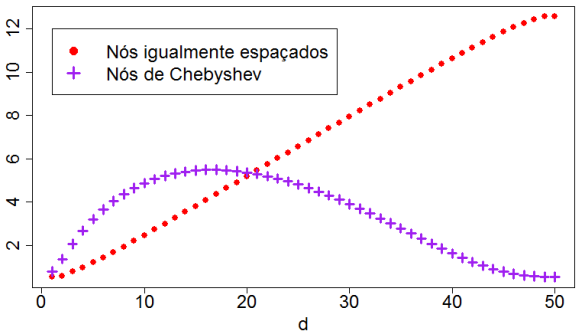


Figura 3.4: *As constantes de Lebesgue* $\Lambda\left(\mathbf{x^n_{eq}}, \mu_d(\mathbf{x^n_{eq}})\right)$ *e* $\Lambda\left(\mathbf{x^n_{cheb2}}, \mu_d(\mathbf{x^n_{cheb2}})\right)$*, em escala logaritmica (base* $= 10$*) para* $n = 50$ *e* $1 \leq d \leq 50$*.*

3.2.5 Sobre a magnitude da função de Lebesgue para o interpolador de Floater-Hormann no interior do intervalo de interpolação

Nesta seção, o conjunto de nós igualmente espaçados $\mathbf{x^n_{eq}}$ será denotado, simplesmente, por $\mathbf{x}$. Em 2013, Klein [Kle13] observou que a função de Lebesgue (2.6)

$$Leb(x, \mathbf{x}, \mu_d(\mathbf{x}))$$

associada ao interpolador de Floater-Hormann com nós igualmente espaçados e $d < n/2$ possui magnitude da ordem de $\log(n)$ no intervalo $[x_d, x_{n-d}]$. Mais precisamente, vale que

$$|Leb(x, \mathbf{x}, \mu_d(\mathbf{x}))| \leq 0.65(2+\log(n+2d)), \quad x \in [x_d, x_{n-d}], d \geq 5,$$

ou, em termos da constante de Lebesgue (2.5) com intervalo de referência[5] $[x_d, x_{n-d}]$,

$$\Lambda(\mathbf{x}, \mu_d(\mathbf{x})|_{x_d}^{x_{n-d}} \leq 0.65(2+\log(n+2d)), \quad d \geq 5. \tag{3.37}$$

Essa é a motivação principal para a definição dos interpoladores de Floater-Hormann estendidos, os quais estudaremos, em detalhe, no Capítulo 6. A desigualdade (3.37) corresponde ao Teorema

[5]Note que o intervalo de referência não precisa necessariamente conter todos os nós de interpolação.

3.1 de [Kle13]. Porém, o leitor eventualmente poderá ter um pouco de dificuldade em fazer essa associação, devido à falta de rigor com que essa desigualdade é apresentada em [Kle13].

A seguir apresentamos uma abordagem alternativa para (3.37), a qual é baseada no caso particular $d = 1$ em (3.36)

$$\Lambda(\mathbf{x}, \mu_d(\mathbf{x})|_{x_0}^{x_n} \quad \leq \quad 2 + \ln(n) \tag{3.38}$$

e na seguinte relação de recorrência (isso será provado adiante) para $d \leq s$:

$$r_{d+1}(x, \mathbf{x}[k, k+s+1], \mathbf{y}[k, k+s+1]) \quad = \\ \frac{\sum_{j=0}^{s} \frac{\mu_d(\mathbf{x}[k,k+s])_j \;\; y_{k+j}}{x - x_{k+j}} \quad - \quad \sum_{j=0}^{s} \frac{\mu_d(\mathbf{x}[k+1,k+1+s])_j \;\; y_{k+1+j}}{x - x_{k+1+j}}}{\sum_{j=0}^{s} \frac{\mu_d(\mathbf{x}[k,k+s])_j}{x - x_{k+j}} \quad - \quad \sum_{j=0}^{s} \frac{\mu_d(\mathbf{x}[k+1,k+1+s])_j}{x - x_{k+1+j}}}, \tag{3.39}$$

onde

$$\mathbf{x}[k, k+s] := (x_k, x_{k+1}, \ldots, x_{k+s}), \qquad \mathbf{y}[k, k+s] := (y_k, y_{k+1}, \ldots, y_{k+s}) \tag{3.40}$$

e

$$r_d(x, \mathbf{x}[k, k+s], \mathbf{y}[k, k+s]) \quad = \quad \sum_{j=0}^{s} \frac{\mu_d(\mathbf{x}[k,k+s])_j \;\; y_{k+j}}{x - x_{k+j}} \Bigg/ \sum_{j=0}^{s} \frac{\mu_d(\mathbf{x}[k,k+s])_j}{x - x_{k+j}}.$$

Note que a relação (3.39) fornece um método simples para calcular os pesos de interpolação para o interpolador de Floater-Hormann para nós igualmente espaçados: $\mu_{d+1}(\mathbf{x}[k, k+1+s])_j \quad =$

$$\begin{cases} \mu_d(\mathbf{x}[k, k+s])_0, & \text{para} \;\; j = 0. \\ \mu_d(\mathbf{x}[k, k+s])_j - \mu_d(\mathbf{x}[k+1, k+1+s])_{j-1}, & \text{para} \;\; 1 \leq j \leq s. \\ -\mu_d(\mathbf{x}[k+1, k+1+s])_s, & \text{para} \;\; j = s+1. \end{cases} \tag{3.41}$$

Assim, para determinar os pesos (simplificados) para o interpolador de Floater-Hormann para $n = 9$ e $d = 3$, por exemplo, podemos partir dos pesos simplificados

$$\tilde{\mu}_0(\mathbf{x}[0, 6]) \quad = \quad (1, -1, 1, -1, 1, -1, 1) \tag{3.42}$$

para o interpolador de Berrut ($d = 0$) e iterar (3.41) para $k = 0$ e $s = 6, 7$ e 8:

$$\begin{aligned} \tilde{\mu}_1(\mathbf{x}[0, 7]) \quad &= \quad \begin{matrix} (1, -1, 1, -1, 1, -1, 1, 0) \\ -(0, 1, -1, 1, -1, 1, -1, 1) \end{matrix} \quad &= \quad (1, -2, 2, -2, 2, -2, 2, -1), \\ \tilde{\mu}_2(\mathbf{x}[0, 8]) \quad &= \quad \begin{matrix} (1, -2, 2, -2, 2, -2, 2, -1, 0) \\ -(0, 1, -2, 2, -2, 2, -2, 2, -1) \end{matrix} \quad &= \quad (1, -3, 4, -4, 4, -4, 4, -3, 1), \\ \tilde{\mu}_3(\mathbf{x}[0, 9]) \quad &= \quad \begin{matrix} (1, -3, 4, -4, 4, -4, 4, -3, 1, 0) \\ -(0, 1, -3, 4, -4, 4, -4, 4, -3, 1) \end{matrix} \quad &= \quad (1, -4, 7, -8, 8, -8, 8, -7, 4, -1). \end{aligned}$$

Essas relações podem ser encontradas na página 8 de [FH07].

A prova de (3.39) segue das identidades (3.29) e (3.30). De fato, temos

$$\frac{\sum_{j=0}^{s} \frac{\mu_d(\mathbf{x}[k,k+s])_j \;\; y_{k+j}}{x-x_{k+j}} \;-\; \sum_{j=0}^{s} \frac{\mu_d(\mathbf{x}[k+1,k+1+s])_j \;\; y_{k+1+j}}{x-x_{k+1+j}}}{\sum_{j=0}^{s} \frac{\mu_d(\mathbf{x}[k,k+s])_j}{x-x_{k+j}} \;-\; \sum_{j=0}^{s} \frac{\mu_d(\mathbf{x}[k+1,k+1+s])_j}{x-x_{k+1+j}}} =$$

$$\frac{\sum_{i=k}^{k+s-d} \lambda_{i,d}(x,\mathbf{x})p_{i,d}(x,\mathbf{x},\mathbf{y}) \;-\; \sum_{i=k+1}^{k+s+1-d} \lambda_{i,d}(x,\mathbf{x})p_{i,d}(x,\mathbf{x},\mathbf{y})}{\sum_{i=k}^{k+s-d} \lambda_{i,d}(x,\mathbf{x}) \;-\; \sum_{i=k+1}^{k+s+1-d} \lambda_{i,d}(x,\mathbf{x})} =$$

$$\frac{\sum_{i=k}^{k+s-d} \Big[\lambda_{i,d}(x,\mathbf{x})p_{i,d}(x,\mathbf{x},\mathbf{y}) \;-\; \lambda_{i+1,d}(x,\mathbf{x})p_{i+1,d}(x,\mathbf{x},\mathbf{y})\Big]}{\sum_{i=k}^{k+s-d} \Big[\lambda_{i,d}(x,\mathbf{x}) \;-\; \lambda_{i+1,d}(x,\mathbf{x})\Big]} =$$

$$\frac{\sum_{i=k}^{k+s-d} \lambda_{i,d+1}(x,\mathbf{x})\Big[p_{i,d}(x,\mathbf{x},\mathbf{y})(x-x_{i+d+1}) \;+\; p_{i+1,d}(x,\mathbf{x},\mathbf{y})(x_i-x)\Big]}{\sum_{i=k}^{k+s-d} (x_i-x_{i+d+1})\Big[\lambda_{i,d+1}(x,\mathbf{x})\Big]} \overset{(3.9)}{=}$$

$$\frac{\sum_{i=k}^{k+s-d} \lambda_{i,d+1}(x,\mathbf{x}) \frac{\Big[p_{i,d}(x,\mathbf{x},\mathbf{y})(x-x_{i+d+1}) \;+\; p_{i+1,d}(x,\mathbf{x},\mathbf{y})(x_i-x)\Big]}{(x_i-x_{i+d+1})}}{\sum_{i=k}^{k+s-d} \lambda_{i,d+1}(x,\mathbf{x})} =$$

$$\frac{\sum_{i=k}^{k+s-d} \lambda_{i,d+1}(x,\mathbf{x})p_{i,d+1}(x,\mathbf{x},\mathbf{y})}{\sum_{i=k}^{k+s-d} \lambda_{i,d+1}(x,\mathbf{x})}.$$

A nossa estimativa para o lado esquerdo de (3.37) é dada por

Teorema 4. *Se* $\mathbf{x}$ *é formado por nós igualmente espaçados e* $1 \le d \le \lfloor n/2 \rfloor$*, então*

$$\Lambda(\mathbf{x}, \mu_d(\mathbf{x}))|_{x_d}^{x_{n-d}} \;\le\; 2 + \ln(n).$$

Esse resultado segue do seguinte Lema, para $u = 0$ e $s = n$

Lema 1. *Se* $\mathbf{x}$ *é formado por nós igualmente espaçados,* $1 \le d \le \lfloor n/2 \rfloor$*,* $s \ge 2d$*,* $u \ge 0$ *e* $u+s \le n$*, então*

$$\Lambda(\mathbf{x}[u,u+s], \mu_d(\mathbf{x}[u,u+s]))|_{x_{u+d}}^{x_{u+s-d}} \;\le\; 2 + \ln(n). \tag{3.43}$$

Demonstração. A prova é feita por indução em d. Para $d = 1$, o resultado segue de (3.38). Suponha, então, que $d \ge 1$, $u + s \le n$, $s \ge 2(d+1)$ e que o resultado vale para d.

Como $u + s - 1 < u + s \le n$ e $s - 1 \ge 2d$, dados $\mathbf{y} \in \mathbb{R}^{n+1}$, com $||\mathbf{y}||_\infty \le 1$ e $x \in [x_{u+(d+1)}, x_{u+s-(d+1)}]/\{x_{u+d+1}, x_{u+d+2}, \ldots, x_{u+s-(d+1)}\}$, segue, pela hipótese de indução, que

$$\left|\sum_{j=0}^{s-1} \frac{\mu_d(\mathbf{x}[u,u+s-1])_j \;\; y_{u+j}}{x-x_{u+j}}\right| \overset{(2.5),(3.43)}{\leq} (2+\ln(n)) \left|\sum_{j=0}^{s-1} \frac{\mu_d(\mathbf{x}[u,u+s-1])_j}{x-x_{u+j}}\right|$$

e

$$\left|\sum_{j=0}^{s-1} \frac{\mu_d(\mathbf{x}[u+1,u+1+s-1])_j \;\; y_{u+1+j}}{x-x_{u+1+j}}\right| \overset{(2.5),(3.43)}{\leq} (2+\ln(n)) \left|\sum_{j=0}^{s-1} \frac{\mu_d(\mathbf{x}[u+1,u+1+s-1])_j}{x-x_{u+1+j}}\right|.$$

Logo, obtemos $|r_{d+1}(x, \mathbf{x}[u, u+s], \mathbf{y}[u, u+s])| \overset{(3.39)}{\leq}$

$$(2+\ln(n)) \frac{\left|\sum_{j=0}^{s-1} \frac{\mu_d(\mathbf{x}[u,u+s-1])_j}{x-x_{u+j}}\right| + \left|\sum_{j=0}^{s-1} \frac{\mu_d(\mathbf{x}[u+1,u+1+s-1])_j}{x-x_{u+1+j}}\right|}{\left|\sum_{j=0}^{s-1} \frac{\mu_d(\mathbf{x}[u,u+s-1])_j}{x-x_{u+j}} - \sum_{j=0}^{s-1} \frac{\mu_d(\mathbf{x}[u+1,u+1+s-1])_j}{x-x_{u+1+j}}\right|}. \tag{3.44}$$

Portanto, basta mostrar que as duas somatórias do denominador em (3.44) possuem sinais opostos. De fato, como os interpoladores de Floater-Hormann não possuem pólos, então, para $0 \leq m \leq n-1$, a função

$$\varphi_m(t) \;\; := \;\; \sum_{j=0}^{s-1} \frac{\mu_d(\mathbf{x}[u+1,u+1+s-1])_j}{t-x_{u+1+j}} \Bigg/ \sum_{j=0}^{s-1} \frac{\mu_d(\mathbf{x}[u,u+s-1])_j}{t-x_{u+j}}$$

não muda de sinal em $]x_m, x_{m+1}[$ e o sinal de $\varphi_m(x)$ pode ser obtido analisando o sinal do limite

$$\lim_{t \longrightarrow x_m^+} \varphi(t) \;=\; \frac{\mu_d(\mathbf{x}[u+1, u+1+s-1])_{m-u-1}}{\mu_d(\mathbf{x}[u, u+s-1])_{m-u}}. \tag{3.45}$$

Por (3.27), temos que os pesos para o interpolador de Floater-Hormann com nós igualmente espaçados depedem apenas dos parâmetros d e n e do espaçamento entre os nós. Logo, temos $\mu_d(\mathbf{x}[u+1, u+1+s-1])_{m-u-1} = \mu_d(\mathbf{x}[u, u+s-1])_{m-u-1}$ e isso e (3.27) mostram que (3.45) é negativo.

□

Capítulo 4

A estabilidade backward da fórmula baricêntrica para interpolação

Neste capítulo apresentamos um estudo sobre a estabilidade numérica da fórmula baricêntrica (2.1) para interpolação, o qual consiste na análise dos Erros II e III definidos na Figura 4.1. Esse estudo teve como motivação principal a análise da estabilidade numérica da fórmula baricêntrica para o interpolador de Lagrange (3.2). A principal referência sobre o tópico até então [Hig04] considerava apenas os efeitos dos erros de arredondamento no Passo 3 da Figura 4.1, ou seja, assumia-se que $\mathbf{x}$ e $\mathbf{y}$ eram formados por números de ponto flutuante (no contexto da interpolação de Lagrange, os pesos de interpolação (3.6) não eram interpretados como parâmetros do problema original de interpolação, por serem definidos em função de $\mathbf{x}$.)

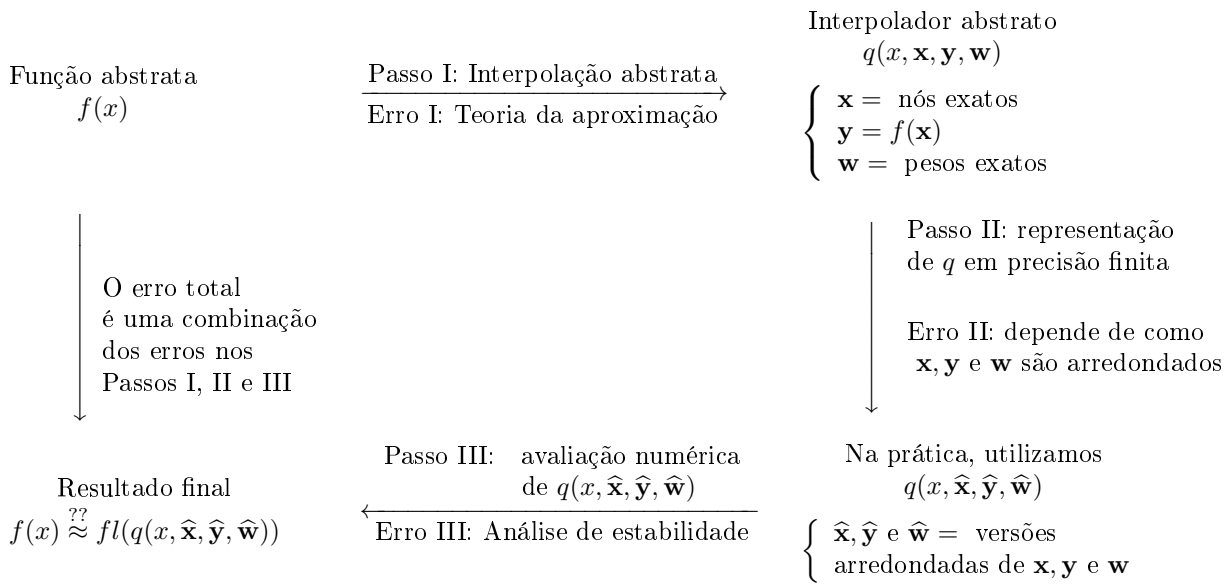


Figura 4.1: *Decomposição do erro de interpolação.*

O trabalho de Higham [Hig04] mostra que, sob o modelo de aritmética de ponto flutuante definido na Seção 2.3, a primeira fórmula baricêntrica (3.5) é backward estável, no sentido de que, para cada número de ponto flutuante x, o valor computado $fl(p(x, \widehat{\mathbf{x}}, \widehat{\mathbf{y}}, \gamma(\widehat{\mathbf{x}})))$ de $p(x, \widehat{\mathbf{x}}, \widehat{\mathbf{y}}, \gamma(\widehat{\mathbf{x}}))$ é igual a $p(x, \widehat{\mathbf{x}}, \widetilde{\mathbf{y}}, \gamma(\widehat{\mathbf{x}}))$, para $\widetilde{\mathbf{y}}$ relativamente próximo a $\widehat{\mathbf{y}}$. Por outro lado, Higham afirma que a segunda fórmula baricêntrica (3.8) para interpolação de Lagrange não possui a propriedade de estabilidade backward em geral.

A propriedade de estabilidade backward depende do significado preciso do termo *relativamente próximo* empregado acima, o qual pode variar de acordo com o contexto, devido às diferenças de rigor exigido em cada situação. A seguir apresentamos uma possível formalização desse conceito, porém não iremos insistir doravante no formalismo com relação a essa questão.

Definição 1. *Seja $\phi(n,\epsilon)$ uma função positiva. Um algoritmo para avaliar uma expressão numérica $u(\alpha_0,\ldots,\alpha_k,\ldots,\alpha_r)$ é dito ϕ-backward estável com relação ao parâmetro $n-$dimensional $\alpha_k = (\alpha_{k,1},\alpha_{k,2},\ldots,\alpha_{k,n})$ se para cada conjunto de parâmetros arredondados $\widehat{\alpha_0},\ldots,\widehat{\alpha_k},\ldots,\widehat{\alpha_r}$ existe um vetor $n-$dimensional $\widetilde{\alpha_k}$ tal que o valor computado $fl(u(\widehat{\alpha_0},\ldots,\widehat{\alpha_k},\ldots,\widehat{\alpha_r}))$ de $u(\alpha_0,\ldots,\alpha_k,\ldots,\alpha_r)$ satisfaz*

$$fl(u(\widehat{\alpha_0},\ldots\widehat{\alpha_k},\ldots,\widehat{\alpha_r})) \;=\; u(\widehat{\alpha_0},\ldots,\widetilde{\alpha_k},\ldots,\widehat{\alpha_r}), \quad com \quad \max_{1\le j\le n}\left|\frac{\widetilde{\alpha_{k,j}}-\widehat{\alpha_{k,j}}}{\widehat{\alpha_{k,j}}}\right| \;\le\; \phi(n,\epsilon). \tag{4.1}$$

No trabalho de Higham, por exemplo, temos $\phi(n,\epsilon) = O(n\epsilon)$.

O primeiro resultado concreto referente à estabilidade backward da segunda fórmula baricêntrica para interpolação foi obtido por Mascarenhas [Mas14], o qual desenvolveu um algoritmo backward estável (conforme a noção de estabilidade backward definida acima) para calcular

$$q(x,\widehat{\mathbf{x}},\widehat{\mathbf{y}},\gamma^*) \tag{4.2}$$

para todo conjunto de nós formados por números de ponto flutuante $\widehat{\mathbf{x}}$, onde γ^* são os pesos simplificados dados por (3.17). Esse interpolador foi inspirado no interpolador de Lagrange para os nós de Chebyshev do segundo tipo $q(x,\mathbf{x_{cheb2}},\widehat{\mathbf{y}},\gamma^*)$. Por coincidência, o interpolador (4.2) também corresponde ao interpolador de Floater-Hormann para nós igualmente espaçados, com $d=1$ (ver equações após (3.42).)

O nosso trabalho generaliza os resultados obtidos em [Mas14] ao estabelecer a propriedade de estabilidade backward para a fórmula baricêntrica (2.1) para famílias mais gerais de nós e de pesos para interpolação, a saber aqueles para os quais a constante de Lebesgue (2.5) é pequena[1]. Além disso, incorporamos à analise da estabilidade backward os efeitos causados pelos erros de arredondamento no Passo II da Figura 4.1.

A necessidade de fragmentar o processo de avaliação numérica da fórmula baricêntrica em duas etapas (Passos II e III) é motivada pelos resultados apresentados na Tabela 5 de [Mas14]. Nela são comparadas as performances das fórmulas

$$p(x,\mathbf{x},fl(f(\mathbf{x})),\gamma(\mathbf{x})) \;=\; \tfrac{(-1)^n2^{n-1}}{n}p(x,\mathbf{x},fl(f(\mathbf{x})),\tfrac{n}{(-1)^n2^{n-1}}\gamma(\mathbf{x})), \tag{4.3}$$

$$p(x,\mathbf{x},fl(f(\mathbf{x})),\gamma(\mathbf{x_{cheb2}})) \;=\; \tfrac{(-1)^n2^{n-1}}{n}p(x,\mathbf{x},fl(f(\mathbf{x})),\gamma^*), \tag{4.4}$$

e

$$q(x,\mathbf{x},fl(f(\mathbf{x})),\gamma(\mathbf{x_{cheb2}})) \;=\; q(x,\mathbf{x},fl(f(\mathbf{x})),\gamma^*), \quad \mathbf{x}=\widehat{\mathbf{x_{cheb2}}} \tag{4.5}$$

para interpolação da função $f(x)=\sin(x)$ próximo aos nós de interpolação e para uma quantidade grande de nós ($n\ge 1000$), de modo a termos um Erro I desprezível (ver equação (3.4)) quando comparado à precisão da máquina utilizada $\epsilon\approx 2.2\times 10^{-16}$. Os valores exibidos naquela tabela correspondem aos valores obtidos pela avaliação numérica das expressões no lado direito das equações (4.3), (4.4) e (4.5) (observe que, nas duas últimas, os pesos simplificados são utilizados.)

O erro para o interpolador (4.3) na Tabela 5 de [Mas14] é $O(n\epsilon)$ e está de acordo com os resultados estabelecidos por Higham [Hig04]. Porém, para o interpolador (4.4), os dados apresentados mostram um erro da ordem de $n^2\epsilon$ e isso justifica a necessidade de uma análise mais apurada sobre a sensibilidade das fórmulas baricêntricas (3.5) e (3.8) com relação aos erros oriundos do arredondamento dos parâmetros $\mathbf{x}$ e $\mathbf{w}$.

Essa discrepância $n\epsilon$ vs $n^2\epsilon$ motivou o estudo apresentado em [MdC16] sobre a estabilidade numérica de interpoladores do tipo (4.4) e (4.5) para interpolação de Lagrange, para os quais os

[1] Para o interpolador de Lagrange associado aos nós de Chebyshev do segundo tipo considerado em [Mas14], por exemplo, a constante de Lebesgue é $O(\log(n))$ (ver equação (3.22).)

pesos de interpolação $\mathbf{w}$ não estão em exata concordância com os pesos $\gamma(\mathbf{x})$ para interpolação polinomial com nós arredondados $\mathbf{x} = \widehat{\mathbf{x}}$. O caso de interesse

$$\mathbf{x} \;=\; \widehat{\mathbf{x}_{\mathbf{cheb2}}}, \qquad \mathbf{w} \;=\; \widehat{\mathbf{w}} = \gamma^*$$

contemplado em (4.4) e (4.5), onde os nós de Chebyshev arredondados são combinados com os pesos simplificados (3.17), é denominado em [MdC16] de o *caso de Salzer*, em homenagem à fórmula (3.17) para os pesos de interpolação polinomial primeiramente deduzida por Salzer [Sal72].

A vantagem na utilização das fórmulas (4.4) e (4.5) com os pesos simplificados ao invés da fórmula (4.3) é a ordem de complexidade computacional. Enquanto são necessárias $O(n^2)$ operações aritméticas para calcular todos os pesos, e portanto, toda a expressão, em (4.3), as expressões (4.4) e (4.5) podem ser avaliadas com apenas $O(n)$ operações aritméticas.

4.1 A estabilidade backward da fórmula baricêntrica para interpolação

Nesta seção apresentamos a análise da estabilidade backward da fórmula baricêntrica (2.1) para interpolação. Para tal fim, consideramos nós $\mathbf{x} \subset [a, b]$ e pesos de interpolação teóricos de referência $\mathbf{w}$ e os nós $\widehat{\mathbf{x}} \subset [\widehat{a}, \widehat{b}]$ e pesos $\widehat{\mathbf{w}}$ (ambos arredondados) que serão efetivamente utilizados na avaliação numérica de (2.1). Assumiremos, também, que os valores observados $\mathbf{y}$ são formados por números de ponto flutuante[2], isto é $\mathbf{y} = \widehat{\mathbf{y}}$. Quando $x_0 = a$, por exemplo, podemos ter $\hat{x}_0 < a$ e isso justifica considerar um novo intervalo de referência $[\widehat{a}, \widehat{b}]$ para interpolação. Em todo caso, definimos $x_{-1} := a$, $\widehat{x}_{-1} := \widehat{a}$, $x_{n+1} := b$ e $\widehat{x}_{n+1} := \widehat{b}$. A relação entre $\mathbf{x}$ e $\widehat{\mathbf{x}}$ e entre os intervalos de referência $[a, b]$ e $[\widehat{a}, \widehat{b}]$ é mensurada por meio dos números

$$\delta_{j,k} \;:=\; \delta(\mathbf{x}, \widehat{\mathbf{x}})_{j,k} \;=\; \tfrac{x_j - x_k}{\widehat{x}_j - \widehat{x}_k} - 1, \quad -1 \le j < k \le n+1$$

$$\delta := \delta(\mathbf{x}, \widehat{\mathbf{x}}) = \max_{j \neq k} |\delta_{j,k}| \tag{4.6}$$

e do homeomorfismo (afim por partes) $\chi : [\widehat{a}, \widehat{b}] \longrightarrow [a, b]$

$$\chi(\widehat{x}) = x_i + \frac{x_{i+1} - x_i}{\widehat{x}_{i+1} - \widehat{x}_i}(\widehat{x} - \widehat{x}_i), \quad \widehat{x}_i \;\le\; \widehat{x} \;<\; \widehat{x}_{i+1}, \quad -1 \;\le\; i \;\le\; n, \tag{4.7}$$

convencionando-se que

$$\widehat{x}_i < \widehat{x}_{i+1} \text{ para } 0 \le i < n, \tag{4.8}$$

$$x_0 = a \quad \Longleftrightarrow \quad \widehat{x}_0 = \widehat{a}, \tag{4.9}$$

$$x_n = b \quad \Longleftrightarrow \quad \widehat{x}_n = \widehat{b} \tag{4.10}$$

e que a relação (4.7) está definida para $i = -1$ e $i = n$ apenas quando $\hat{a} \neq \hat{x}_0$ e $\hat{x}_n \neq \hat{b}$. Analogamente, definimos $\delta_{j,k} = 0$ quando $\widehat{x}_j = \widehat{x}_k$.

Para evitar casos patológicos, iremos assumir, também, que

$$w_i \;\neq\; 0, \qquad \widehat{w}_i \;\neq\; 0, \quad 0 \;\le\; i \le\; n. \tag{4.11}$$

A relação entre $\mathbf{w}$ e $\widehat{\mathbf{w}}$ é quantizada pelo vetor de erros relativos $\boldsymbol{\zeta}$

$$\zeta_i \;:=\; \zeta(\mathbf{w}, \widehat{\mathbf{w}})_i \;=\; \frac{w_i}{\widehat{w}_i} - 1, \quad i = 0, 1, \ldots, n. \tag{4.12}$$

[2] Os efeitos do arredondamento de $\mathbf{y}$ podem ser mensurados facilmente em termos da desigualdade (2.7)

Exemplo 1. *Em [MdC16], mostramos que a diferença entre as performances dos interpoladores (4.3) e (4.4) relatada na seção anterior se deve, essencialmente*[3]*, à diferença na magnitude do vetor de erros relativos (4.12) em cada caso. Em ambas as fórmulas (4.3) e (4.4), os nós de referência* $\mathbf{x}$ *são tomados como os nós de Chebyshev do segundo tipo arredondados, isto é:*

$$\mathbf{x} = \widehat{\mathbf{x}} = \widehat{\mathbf{x_{cheb2}}}$$

e os pesos de referência são dados por

$$\mathbf{w} = \frac{n}{(-1)^n 2^{n-1}} \gamma(\mathbf{x}). \tag{4.13}$$

Porém, no caso do interpolador (4.3), os pesos arredondados $\widehat{\mathbf{w}}^{num}$ *são obtidos pela avaliação numérica de* $\frac{n}{(-1)^n 2^{n-1}} \gamma(\mathbf{x})$ *segundo a fórmula (3.6)*

$$\widehat{\mathbf{w}}^{num} = fl\left(\frac{n}{(-1)^n 2^{n-1}} \gamma(\mathbf{x})\right), \tag{4.14}$$

equanto que os pesos arredondados $\widehat{\mathbf{w}}^{sal}$ *para o interpolador (4.4) são os pesos simplicados* γ^* *dados por (3.17).*

De acordo com as regras de arredondamento descritas na Seção 2.3, vale que

$$\begin{aligned}
\widehat{w}_i^{num} &\overset{(*)}{=} \tfrac{1}{(-1)^n 2^{n-1}} fl\left(n\gamma(\mathbf{x})\right) \\
&\overset{(2.16)}{=} \tfrac{n}{(-1)^n 2^{n-1}} \langle 1\rangle_i fl\left(\gamma(\mathbf{x})\right) \\
&= \tfrac{n}{(-1)^n 2^{n-1}} \langle 1\rangle_i fl\left(\prod_{\substack{j=0\\ j\neq i}}^{n} \tfrac{1}{(\widehat{x_{cheb2}})_i - (\widehat{x_{cheb2}})_j}\right) \\
&\overset{(2.20)}{=} \tfrac{n}{(-1)^n 2^{n-1}} \langle 1\rangle_i \langle 2n\rangle_i \prod_{\substack{j=0\\ j\neq i}}^{n} \tfrac{1}{(\widehat{x_{cheb2}})_i - (\widehat{x_{cheb2}})_j} \\
&\overset{(2.17)}{=} \langle 2n+1\rangle_i \tfrac{n}{(-1)^n 2^{n-1}} \gamma(\mathbf{x})_i \\
&= \langle 2n+1\rangle_i w_i,
\end{aligned}$$

ou seja,

$$||\boldsymbol{\zeta}^{num}||_\infty = \max_{0\le i\le n}\left|\frac{w_i}{\widehat{w}_i^{num}} - 1\right| = \max_{0\le i\le n}\left|\frac{1}{\langle 2n+1\rangle_i} - 1\right| \overset{(2.17),(2.19)}{\le} 1.01(2n+1)\epsilon, \tag{4.15}$$

para $(2n+1)\epsilon < 0.001$*. Em contraste, o Lema 3 de [MdC16] fornece o seguinte limitante superior para* $||\boldsymbol{\zeta}^{sal}||_\infty$

$$||\boldsymbol{\zeta}^{sal}||_\infty \le 2.4624||\widehat{\mathbf{x_{cheb2}}} - \mathbf{x_{cheb2}}||_\infty n^2 < 9n^2\epsilon, \tag{4.16}$$

para $||\widehat{\mathbf{x_{cheb2}}} - \mathbf{x_{cheb2}}||_\infty \le 3\epsilon$.

[3] As quantidades mais apropriadas para descrever esse fenômeno são $z_i = \frac{\widehat{w}_i}{w_i} - 1, i = 0, 1, \ldots, n$ [MdC16]. Porém, se z_i é pequeno, como, por exemplo, $|z_i| < 0.5$, então z_i e $\zeta_i = \frac{-z_i}{1+z_i}$ possuem a mesma ordem de magnitude.

(*) Não há erros de arrendamento ao multiplicar/ dividir um número de ponto flutuante por uma potência de 2.

A Figura 4.2 mostra os valores de $||\boldsymbol{\zeta}^{num}||_\infty$ *e* $||\boldsymbol{\zeta}^{sal}||_\infty$ *calculados numericamente para diversos valores de* n *e as retas de quadrados mínimos obtidas mostram que os limitantes superiores em* (4.15) *e* (4.16) *fornecem a ordem correta para magnitude dos vetores* $\boldsymbol{\zeta}^{num}$ *e* $\boldsymbol{\zeta}^{sal}$*. Os dados representados na Figura 4.2 foram obtidos com o auxílio da biblioteca MPFR. Os nós* $\widehat{\mathbf{x}_{\mathbf{cheb2}}}$ *foram calculados com precisão dupla, utilizando-se a constante* M_PI *da biblioteca C++ padrão e a expressão* $x_i = \sin\left(\pi\frac{2i-n}{2n}\right)$ *para os nós* (3.16)*. Os pesos de referência* $\mathbf{w}$ *foram obtidos pela avaliação numérica da expressão* (4.13) *utilizando-se aritmética de alta precisão, com 50 casas decimais, e os pesos arredondados* $\widehat{\mathbf{w}}^{num}$ *foram obtidos pela avaliação numérica da expressão* (4.14) *em aritmética de precisão dupla.*

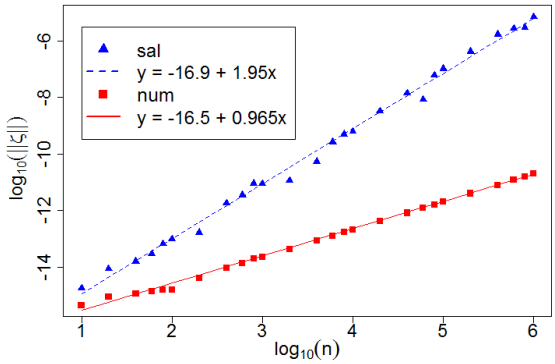


Figura 4.2: *Erro relativo nos pesos de interpolação.*

Antes de prosseguir com o enunciado dos teoremas sobre a estabilidade backward da fórmula baricêntrica, precisamos do seguinte resultado

Lema 2. *A transformação* $\chi : [\widehat{x}_{-1}, \widehat{x}_{n+1}] \longrightarrow [x_{-1}, x_{n+1}]$ *definida em* (4.7) *é estritamente crescente e satisfaz*

$$|\chi(\hat{x}) - \hat{x}| \;\leq\; \max_{-1\leq i\leq n+1} |\hat{x}_i - x_i| \tag{4.17}$$

e

$$\left|\frac{\chi(\hat{x})-x_i}{\hat{x}-\hat{x}_i} - 1\right| \;\leq\; \max\{\delta_{k,i},\ \delta_{k+1,i}\}, \quad \textit{para} \;\; \hat{x}_k \leq \hat{x} \leq \hat{x}_{k+1}.$$

Demonstração. Para mostrar (4.17), observe que $g(\widehat{x}) := \chi(\hat{x}) - \hat{x}$ é uma função afim por partes e, portanto, assume seus máximos e mínimos locais nos pontos $\widehat{x}_{-1}, \widehat{x}_0, \ldots, \widehat{x}_n, \widehat{x}_{n+1}$. Analogamente, a função

$$\begin{aligned} h(\hat{x}) &= \frac{\chi(\hat{x})-x_i}{\hat{x}-\hat{x}_i} - 1 \\ &= \frac{x_k + \frac{x_{k+1}-x_k}{\widehat{x}_{k+1}-\widehat{x}_k}(\widehat{x}-\hat{x}_i+\hat{x}_i-\widehat{x}_k)-x_i}{\hat{x}-\hat{x}_i} - 1 \\ &= \left(\frac{x_{k+1}-x_k}{\widehat{x}_{k+1}-\widehat{x}_k} - 1\right) + \left(\frac{x_{k+1}-x_k}{\widehat{x}_{k+1}-\widehat{x}_k}(\widehat{x}_i - \widehat{x}_k) + (x_k - x_i)\right)\frac{1}{\hat{x}-\hat{x}_i} \end{aligned}$$

ou é constante (para $i \in \{k, k+1\}$), ou é monótona (para $i \notin \{k, k+1\}$) no intervalo $[\hat{x}_k, \widehat{x}_{k+1}]$ e, portanto, assume seus valores de máximo e mínimo nos extremos desse intervalo. Logo,

$$|h(\hat{x})| \;\leq\; \max\{|h(\hat{x}_k)|, |h(\hat{x}_{k+1})|\} \;=\; \max\{\delta_{k,i}, \delta_{k+1,i}\} \;\; \forall\, \hat{x} \;\; [\hat{x}_k, \hat{x}_{k+1}].$$

□

4.1.1 Resultados teóricos

Teorema 5. *Sob as hipóteses* (4.8), (4.9), (4.10) *e* (4.11), *se*

$$\sum_{i=0}^{n} \tfrac{w_i}{x-x_i} \neq 0 \ \ \forall\, x \in [x_{-1}, x_{n+1}]/\{x_0, x_1, \ldots, x_n\}$$

e δ *em* (4.6), $\boldsymbol{\zeta}$ *em* (4.12) *e a constante de Lebesgue* (2.5) *satisfazem*

$$\delta < \frac{1-||\boldsymbol{\zeta}||_\infty}{\Lambda(\mathbf{x},\mathbf{w})} - ||\boldsymbol{\zeta}||_\infty, \tag{4.18}$$

então, dado $\hat{x} \in [\hat{x}_{-1}, \hat{x}_{n+1}]/\{\hat{x}_0, \hat{x}_1, \ldots, \hat{x}_n\}$, *vale que*

$$\sum_{i=0}^{n} \frac{\widehat{w}_i}{\hat{x}-\hat{x}_i} \neq 0 \tag{4.19}$$

e existe $\boldsymbol{\beta} \in \mathbb{R}^{n+1}$ *tal que*

$$q(\hat{x}, \widehat{\mathbf{x}}, \mathbf{y}, \widehat{\mathbf{w}}) = q(\chi(\hat{x}), \mathbf{x}, \widetilde{\mathbf{y}}, \mathbf{w}), \qquad \textit{para} \quad \tilde{y}_i = (1+\beta_i)y_i, \ i = 0, 1, \ldots, n, \tag{4.20}$$

com

$$||\boldsymbol{\beta}||_\infty \leq \frac{(\delta + ||\boldsymbol{\zeta}||_\infty)(1+\Lambda(\mathbf{x},\mathbf{w}))}{1-||\boldsymbol{\zeta}||_\infty - [\delta + ||\boldsymbol{\zeta}||_\infty]\Lambda(\mathbf{x},\mathbf{w})}. \tag{4.21}$$

Demonstração. Pelo Lema 2, Temos que

$$\nu_i := \tfrac{\chi(\hat{x})-x_i}{\hat{x}-\hat{x}_i} - 1$$

satisfaz $|\nu_i| \leq \delta$ e (4.18) mostra que $||\boldsymbol{\zeta}||_\infty < 1$. Logo, podemos escrever $\hat{w}_i = w_i/(1+\zeta_i)$ e segue que

$$\begin{aligned} \sum_{i=0}^{n} \tfrac{\widehat{w}_i}{\hat{x}-\hat{x}_i} \Big/ \sum_{i=0}^{n} \tfrac{w_i}{\chi(\hat{x})-x_i} &= \sum_{i=0}^{n} \tfrac{w_i/(1+\zeta_i)}{(\chi(\hat{x})-x_i)/(1+\nu_i)} \Big/ \sum_{i=0}^{n} \tfrac{w_i}{\chi(\hat{x})-x_i} \\ &= 1 + \sum_{i=0}^{n} \frac{w_i\left(\frac{1+\nu_i}{1+\zeta_i}-1\right)}{\chi(\hat{x})-x_i} \Big/ \sum_{i=0}^{n} \tfrac{w_i}{\chi(\hat{x})-x_i} = 1 + q(\chi(\hat{x}), \mathbf{x}, \sigma, \mathbf{w}), \end{aligned} \tag{4.22}$$

para

$$\sigma_i := \tfrac{1+\nu_i}{1+\zeta_i} - 1, i = 0, 1, \ldots, n.$$

Como

$$||\sigma||_\infty \leq \frac{||\nu||_\infty + ||\boldsymbol{\zeta}||_\infty}{1-||\boldsymbol{\zeta}||_\infty} \leq \frac{\delta + ||\boldsymbol{\zeta}||_\infty}{1-||\boldsymbol{\zeta}||_\infty}, \tag{4.23}$$

então

$$|q(\chi(\hat{x}), \mathbf{x}, \sigma, \mathbf{w})| \overset{(2.5)}{\leq} \Lambda(\mathbf{x},\mathbf{w})||\sigma||_\infty \leq \Lambda(\mathbf{x},\mathbf{w})\frac{\delta + ||\boldsymbol{\zeta}||_\infty}{1-||\boldsymbol{\zeta}||_\infty} \overset{(4.18)}{<} 1 \tag{4.24}$$

e isso prova (4.19).

Por (4.22), segue, também, que

$$
\begin{aligned}
q(\hat{x}, \widehat{\mathbf{x}}, \mathbf{y}, \widehat{\mathbf{w}}) &= \sum_{i=0}^{n} \frac{y_i \widehat{w}_i}{\hat{x}-\hat{x}_i} \Bigg/ \sum_{i=0}^{n} \frac{\widehat{w}_i}{\hat{x}-\hat{x}_i} \\
&= \sum_{i=0}^{n} \frac{y_i w_i \left(\frac{1+\nu_i}{1+\zeta_i}\right)}{\chi(\hat{x})-x_i} \Bigg/ \left(\sum_{i=0}^{n} \frac{w_i}{\chi(\hat{x})-x_i} \right) (1 + q(\chi(\hat{x}), \mathbf{x}, \sigma, \mathbf{w})) \\
&= q(\chi(\hat{x}), \mathbf{x}, \tilde{\mathbf{y}}, \mathbf{w}),
\end{aligned}
$$

para

$$
\tilde{y}_i := y_i \left(\frac{1+\nu_i}{1+\zeta_i} \right) \frac{1}{1 + q(\chi(\hat{x}), \mathbf{x}, \sigma, \mathbf{w})}. \tag{4.25}
$$

Logo, $\tilde{y}_i = y_i(1+\beta_i)$, com

$$
\begin{aligned}
|\beta_i| &= \left| \left(\frac{1+\nu_i}{1+\zeta_i}\right) \frac{1}{1+q(\chi(\hat{x}),\mathbf{x},\sigma,\mathbf{w})} - 1 \right| \\
&= \left| \frac{\left(\frac{1+\nu_i}{1+\zeta_i}-1\right) - q(\chi(\hat{x}),\mathbf{x},\sigma,\mathbf{w})}{1+q(\chi(\hat{x}),\mathbf{x},\sigma,\mathbf{w})} \right| \leq \frac{\left|\frac{1+\nu_i}{1+\zeta_i}-1\right| + |q(\chi(\hat{x}),\mathbf{x},\sigma,\mathbf{w})|}{1-|q(\chi(\hat{x}),\mathbf{x},\sigma,\mathbf{w})|} \\
&\overset{(4.24)}{\leq} \frac{(1+\Lambda(\mathbf{x},\mathbf{w}))||\sigma||_\infty}{1-\Lambda(\mathbf{x},\mathbf{w})||\sigma||_\infty} \\
&\overset{(4.23)}{\leq} \frac{(\delta+||\zeta||_\infty)[1+\Lambda(\mathbf{x},\mathbf{w})]}{1-||\boldsymbol{\zeta}||_\infty - \Lambda(\mathbf{x},\mathbf{w})(\delta+||\boldsymbol{\zeta}||_\infty)}, \quad i = 0, 1, \ldots, n.
\end{aligned}
$$

□

Corolário 1. *Sob as hipóteses do Teorema 5, vale que*

$$
\Lambda(\widehat{\mathbf{x}}, \widehat{\mathbf{w}})|_{\hat{a}}^{\hat{b}} \leq \Lambda(\mathbf{x}, \mathbf{w})|_a^b \frac{1+\delta}{1-||\boldsymbol{\zeta}||_\infty - \Lambda(\mathbf{x},\mathbf{w})|_a^b(\delta+||\boldsymbol{\zeta}||_\infty)}.
$$

Demonstração. Para $\hat{x} \in [\hat{a}, \hat{b}]$, denote por $\tilde{\mathbf{y}}(\hat{x}, \mathbf{y})$ e $\beta(\hat{x}, \mathbf{y})$ os vetores especificados em (4.25). Então:

$$
\begin{aligned}
\Lambda(\widehat{\mathbf{x}}, \widehat{\mathbf{w}})|_{\hat{a}}^{\hat{b}} &= \max_{||\mathbf{y}|| \leq 1} \left(\max_{\hat{x} \in [\hat{a},\hat{b}]} |q(\hat{x}, \widehat{\mathbf{x}}, \mathbf{y}, \widehat{\mathbf{w}})| \right) \\
&\overset{(4.20)}{=} \max_{||\mathbf{y}|| \leq 1} \left(\max_{\hat{x} \in [\hat{a},\hat{b}]} |q(\chi(\hat{x}), \mathbf{x}, \tilde{\mathbf{y}}(\hat{x}, \mathbf{y}), \mathbf{w})| \right) \\
&\overset{(2.5)}{\leq} \Lambda(\mathbf{x}, \mathbf{w})|_a^b \left(\max_{||\mathbf{y}|| \leq 1} \max_{\hat{x} \in [\hat{a},\hat{b}]} ||\tilde{\mathbf{y}}(\hat{x}, \mathbf{y})||_\infty \right) \\
&\leq \Lambda(\mathbf{x}, \mathbf{w})|_a^b \left(1 + \max_{||\mathbf{y}|| \leq 1} \max_{\hat{x} \in [\hat{a},\hat{b}]} ||\boldsymbol{\beta}(\hat{x}, \mathbf{y})||_\infty \right) \\
&\overset{(4.21)}{\leq} \Lambda(\mathbf{x}, \mathbf{w})|_a^b \left(1 + \frac{(\delta+||\zeta||_\infty)(1+\Lambda(\mathbf{x},\mathbf{w})|_a^b)}{1-||\zeta||_\infty - [\delta+||\zeta||_\infty]\,\Lambda(\mathbf{x},\mathbf{w})|_a^b} \right) \\
&= \Lambda(\mathbf{x}, \mathbf{w})|_a^b \left(\frac{1+\delta}{1-||\zeta||_\infty - [\delta+||\zeta||_\infty]\,\Lambda(\mathbf{x},\mathbf{w})|_a^b} \right)
\end{aligned}
$$

□

Teorema 6. *Assuma que a precisão da máquina ϵ definida em (2.14) seja tal que $(2n+6)\epsilon < 0.001$ e defina*

$$
Z = \frac{||\boldsymbol{\zeta}||_\infty + (n+2)\epsilon}{1-(n+2)\epsilon}. \tag{4.26}
$$

Sob as hipóteses do Teorema 5, se δ, Z e $\Lambda(\mathbf{x},\mathbf{w})$ satisfazem

$$(\delta+Z)\Lambda(\mathbf{x},\mathbf{w})+Z<1$$

e $\hat{x}\in[\hat{a},\hat{b}]$ é um número de ponto flutuante, então o valor computado $fl(q(\hat{x},\widehat{\mathbf{x}},\mathbf{y},\widehat{\mathbf{w}}))$ é igual a $q(x,\mathbf{x},\tilde{\mathbf{y}},\mathbf{w}))$ para algum $x\in[a,b]$ e $\mathbf{y}\in\mathbb{R}^{n+1}$ tais que

$$|\hat{x}-x|\leq\max_{-1\leq i\leq n+1}|\hat{x}_i-x_i|$$

e

$$\tilde{y}_i=y_i(1+\alpha_i)(1+\nu_i),\ \ i=0,1,\ldots,n,$$

com

$$||\boldsymbol{\nu}||_\infty\ \leq\ 1.01(2n+6)\epsilon\ \ \ e\ \ ||\boldsymbol{\alpha}||_\infty\ \leq\ \frac{(1+\Lambda(\mathbf{x},\mathbf{w}))\,(\delta+Z)}{1-Z-(\delta+Z)\Lambda(\mathbf{x},\mathbf{w})}. \tag{4.27}$$

Demonstração. Se $\hat{x}=\hat{x}_k$, para $k\in\{0,1,\ldots,n\}$, basta tomar $x=x_k$ e $\tilde{\mathbf{y}}=\mathbf{y}$. Vamos assumir, portanto, que $\hat{x}\in[\hat{a},\hat{b}]/\{\hat{x}_0,\hat{x}_1,\ldots,\hat{x}_n\}$.

Pelas regras de arredondamento descritas na Seção (2.3), temos que

$$fl\left(\sum_{i=0}^{n}\tfrac{\widehat{w}_i}{\hat{x}-\hat{x}_i}\right)\ =\ \sum_{i=0}^{n}\tfrac{\widehat{w}_i\langle n+2\rangle_i}{\hat{x}-\hat{x}_i}\quad\text{e}\quad fl\left(\sum_{i=0}^{n}\tfrac{y_i\widehat{w}_i}{\hat{x}-\hat{x}_i}\right)\ =\ \sum_{i=0}^{n}\tfrac{y_i\widehat{w}_i\langle n+3\rangle_i}{\hat{x}-\hat{x}_i}$$

e, portanto,

$$\begin{aligned}fl(q(\hat{x},\widehat{\mathbf{x}},\mathbf{y},\widehat{\mathbf{w}}))\ &=\ fl\left(\sum_{i=0}^{n}\tfrac{y_i\widehat{w}_i\langle n+3\rangle_i}{\hat{x}-\hat{x}_i}\Big/\sum_{i=0}^{n}\tfrac{\widehat{w}_i\langle n+2\rangle_i}{\hat{x}-\hat{x}_i}\right)\\ &\overset{(2.17)}{=}\ \sum_{i=0}^{n}\tfrac{y_iw_i'\frac{\langle n+4\rangle_i}{\langle n+2\rangle_i}}{\hat{x}-\hat{x}_i}\Big/\sum_{i=0}^{n}\tfrac{w_i'}{\hat{x}-\hat{x}_i}\\ &=\ q(\hat{x},\widehat{\mathbf{x}},\mathbf{y}',\mathbf{w}'),\end{aligned}$$

para

$$w_i'\ =\ \hat{w}_i\langle n+2\rangle_i\quad\text{e}\quad y_i'\ =\ y_i\frac{\langle n+4\rangle_i}{\langle n+2\rangle_i}\overset{(2.17)}{=}\ y_i\langle 2n+6\rangle_i,\ \ i=0,1,\ldots,n. \tag{4.28}$$

Como, também, $w_i=\hat{w}_i(1+\zeta_i), i=0,1,\ldots,n$, então $\zeta_i':=\frac{w_i}{w_i'}-1$ satisfaz

$$\begin{aligned}|\zeta_i'|\ &=\ \left|\tfrac{w_i}{\hat{w}_i\langle n+2\rangle_i}-1\right|\ =\ \left|\tfrac{(1+\zeta_i)}{\langle n+2\rangle_i}-1\right|\overset{(2.17)}{=}\ |\langle n+2\rangle_{i'}(1+\zeta_i)-1|\\ &\leq\ |\langle n+2\rangle_{i'}\zeta_i|+|\langle n+2\rangle_{i'}-1|\overset{(2.19)}{\leq}\ |\zeta_i|\left(1+\tfrac{(n+2)\epsilon}{1-(n+2)\epsilon}\right)+\tfrac{(n+2)\epsilon}{1-(n+2)\epsilon}\\ &\leq\ \tfrac{|\zeta_i|+(n+2)\epsilon}{1-(n+2)\epsilon}\ \leq\ Z.\end{aligned}$$

Logo, aplicando o Teorema 5 para $(\mathbf{x},\mathbf{w})$, $(\hat{\mathbf{x}},\mathbf{w}')$ e $\mathbf{y}=\mathbf{y}'$, obtemos $fl(q(\hat{x},\widehat{\mathbf{x}},\mathbf{y},\widehat{\mathbf{w}}))=q(\hat{x},\widehat{\mathbf{x}},\mathbf{y}',\mathbf{w}')=q(\chi(\hat{x}),\mathbf{x},\tilde{\mathbf{y}},\mathbf{w})$, para

$$\begin{aligned}\tilde{y}_i\ =\ y_i'(1+\alpha_i)\ &=\ y_i(1+\alpha_i)\langle 2n+6\rangle_i\\ &=\ y_i(1+\alpha_i)(1+\nu_i), i=0,1,\ldots,n,\end{aligned}$$

com $|\chi(\hat{x})-x|\leq||\hat{\mathbf{x}}-\mathbf{x}||_\infty$ (ver Lema 2) e

$$|\nu_i|=|\langle 2n+6\rangle_i-1|\leq 1.01(2n+6)\epsilon\ \ \text{e}\ \ |\alpha_i|\ \leq\ \tfrac{(\delta+Z)(1+\Lambda(\mathbf{x},\mathbf{w}))}{1-Z-(\delta+Z)\Lambda(\mathbf{x},\mathbf{w})}.$$

□

Observação 2. *Os resultados dessa seção foram publicados no artigo [MdC14]. No enunciado do Teorema (2.1) em [MdC14], resultado correspondente ao Teorema 6, está escrito "$(2n+5)\epsilon$"ao invés de "$(2n+6)\epsilon$". Esse erro é proveniente da omissão de uma passagem no desenvolvimento da equação analoga à equação (4.28) em [MdC14].*

4.1.2 Experimentos numéricos

O Teorema 6 afirma que, sob certas hipóteses,

$$fl(q(\hat{x}, \widehat{\mathbf{x}}, \mathbf{y}, \widehat{\mathbf{w}})) = q(x, \mathbf{x}, \tilde{\mathbf{y}}, \mathbf{w})), \tag{4.29}$$

para algum vetor $\boldsymbol{\beta} \in \mathbb{R}^{n+1}$ tal que

$$\tilde{y}_i = (1+\beta_i)y_i, \ i = 0, 1, \ldots, n,$$

e fornece uma estimativa superior para $||\boldsymbol{\beta}||_\infty$, a qual depende dos parâmetros $n, \delta, \boldsymbol{\zeta}$ e Λ.

Se os denominadores em (4.26) e (4.27) são relativamente grandes, digamos

$$1 - Z - (\delta + Z)\Lambda(\mathbf{x}, \mathbf{w}) \ \geq \ 0.5 \quad \text{e} \quad 1 - (n+2)\epsilon \ \geq \ 0.5,$$

então a ordem de grandeza dos limitantes superiores para $\boldsymbol{\alpha}$ e $\boldsymbol{\nu}$ em (4.27) são

$$\max\{||\boldsymbol{\zeta}||_\infty, \ n\epsilon, \ \delta\}\Lambda(\mathbf{x}, \mathbf{w}) \quad \text{e} \quad n\epsilon,$$

respectivamente, e isso nos leva a um limitante superior para $\boldsymbol{\beta} = \boldsymbol{\alpha} + \boldsymbol{\nu} + \boldsymbol{\alpha\nu}$ ($\boldsymbol{\alpha\nu}$ denota o produto coordenada a coordenada de $\boldsymbol{\alpha}$ e $\boldsymbol{\nu}$) da ordem de

$$\max\{||\boldsymbol{\zeta}||_\infty, \ n\epsilon, \ \delta\}\Lambda(\mathbf{x}, \mathbf{w}). \tag{4.30}$$

Nessa seção apresentamos alguns experimentos numéricos para mostrar que a estimativa (4.30) fornece a ordem correta da dependência do erro backward em função desses parâmetros.

O vetor de erros relativos $\boldsymbol{\beta}$ é, em geral, difícil de mensurar, pois há apenas uma equação, a saber a equação (4.29), para determinar as $n+1$ incógnitas $\beta_0, \beta_1, \ldots, \beta_n$. Porém, quando $\mathbf{y} = \mathbf{e^{(j)}}$ para algum índice j, isto é,

$$y_j = 1, \qquad y_i = 0, i \neq j,$$

podemos determinar β_j explicitamente por meio de (4.29):

$$fl\left(q(\hat{x}, \widehat{\mathbf{x}}, \mathbf{e^{(j)}}, \widehat{\mathbf{w}})\right) \ = \ \frac{(1+\beta_j)w_j}{x - x_j} \Bigg/ \sum_{i=0}^{n} \frac{w_i}{x - x_i} \ = \ (1+\beta_j)q(x, \mathbf{x}, \mathbf{e^{(j)}}, \mathbf{w}),$$

ou seja,

$$\beta_j = \frac{fl\left(q(\hat{x}, \widehat{\mathbf{x}}, \mathbf{e^{(j)}}, \widehat{\mathbf{w}})\right)}{q(x, \mathbf{x}, \mathbf{e^{(j)}}, \mathbf{w})} - 1. \tag{4.31}$$

Sensibilidade com relação à $\boldsymbol{\zeta}$

Para analisar a sensibilidade do erro backward com relação ao parâmetro $\boldsymbol{\zeta}$, realizamos um experimento com os interpoladores (4.3) e (4.4) discutidos no Exemplo 1.

Nesse caso, temos $\mathbf{x} = \widehat{\mathbf{x}} = \widehat{\mathbf{x_{cheb2}}}$ e, então, $\delta = 0$. Além disso, o Lema 4 em [MdC16] mostra que, se $||\mathbf{x} - \mathbf{x_{cheb2}}||_\infty \leq 4.6 \times 10^{-16}$, então

$$\Lambda(\mathbf{x}, \gamma(\mathbf{x})) \ \leq \ 1.063\Lambda(\mathbf{x_{cheb2}}, \gamma(\mathbf{x_{cheb2}})) \ \overset{(3.22)}{\leq} \ 1.063\left(\tfrac{2}{\pi}\log(n) + 1.01\right).$$

Isso mostra que, para cada um dos dois casos considerados $\hat{\mathbf{w}} = \mathbf{w}^{num}$ e $\hat{\mathbf{w}} = \mathbf{w}^{sal}$, a estimativa (4.30) para a ordem do erro backward é essencialmente descrita pela magnitude ($O(n\epsilon)$ e $O(n^2\epsilon)$, respectivamente) dos vetores de erros relativos $\boldsymbol{\zeta}^{num}$ e $\boldsymbol{\zeta}^{sal}$.

Para ilustrar esse fenômeno, calculamos β_0 em (4.31) no número de ponto flutuante x_j^* mais próximo (à direita) de $\hat{x}_k = fl\left(-cos\left(\frac{k\pi}{n}\right)\right)$, para diversos valores de n e k, com $k \leq n/2$. Os numeradores em (4.31), em cada caso, foram obtidos avaliando-se as expressões $q\left(x_k^*, \mathbf{x}, \mathbf{e^{(0)}}, \mathbf{w}^{num}\right)$ e $q\left(x_k^*, \mathbf{x}, \mathbf{e^{(0)}}, \mathbf{w}^{sal}\right)$ em precisão dupla. O denominador $q(x_k^*, \mathbf{x}, \mathbf{e^{(0)}}, \gamma(\mathbf{x}))$ em (4.31) foi obtido pela avaliação numérica dessa expressão em precisão elevada (50 casas decimais), com o auxílio da biblioteca de precisão múltipla MPFR.

As retas de quadrados mínimos na Figura 4.3 mostram que, de fato, $|\beta_0|$ é da ordem de $n\epsilon$,para $\mathbf{w}^{num}$, e da ordem de $n^2\epsilon$, para $\mathbf{w}^{sal}$, conforme já havíamos previsto em teoria.

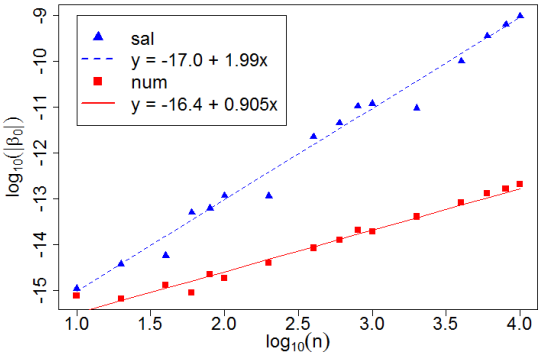


Figura 4.3: *O erro backward* $\max\limits_{1\leq k\leq n/2} \log_{10}(|\beta_0|)$ *para os pesos* $\widehat{\mathbf{w}}^{num}$ *e* $\widehat{\mathbf{w}}^{sal}$.

Sensibilidade com relação à Λ

Para analisar a relação entre o erro backward e a constante de Lebesgue Λ, realizamos um experimento com os interpoladores polinomiais de Lagrange[4] com nós igualmente espaçados para $n \in \{2, 3, \ldots, 50\}$. Para cada valor de n, escolhemos $n+1$ nós igualmente espaçados $\mathbf{x}$ em $[0, n]$, de modo a termos nós inteiros (representáveis em precisão finita) e, portanto, $\delta = 0$. Para cada valor de n, calculamos β_0 em (4.31) em 1002 pontos igualmente espaçados em $[0, n]$ e com pesos arredondados obtidos pela avaliação numérica de $\mathbf{w} = \gamma(\mathbf{x})$. Nesse caso, o argumento acima da equação (4.15) também se aplica e temos que vetor de erros relativos $\boldsymbol{\zeta}$ possui magnitude da ordem de $n\epsilon$. Dessa forma, temos que a estimativa (4.30) para a ordem do erro backward é essencialmente descrita por $\Lambda(\mathbf{x}, \gamma(\mathbf{x}))$, uma vez que a constante de Lebesgue para o interpolador de Lagrange com nós igualmente espaçados possui crescimento exponencial com relação à n (ver Seção 3.1.4.)

Os valores mostrados na Figura 4.4, mostram que, de fato, $|\beta_0|$ possui crescimento exponencial em função de n.

[4]Na Seção 5.4.1 há experimentos similares com os interpoladores de Floater-Hormann.

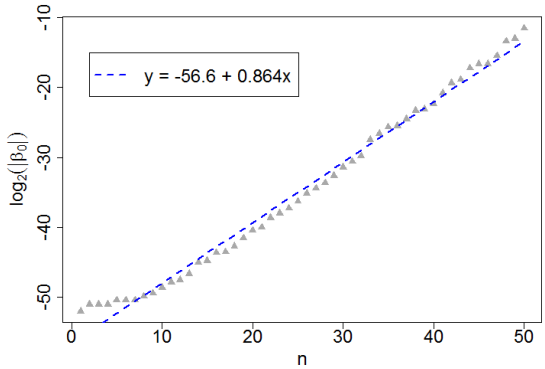


Figura 4.4: *O erro backward máximo* $\max \log_2 (|\beta_0|)$ *sob* 10002 *pontos igualmente espaçados em* $[0, n]$.

Capítulo 5

A estabilidade numérica do interpolador de Floater-Hormann

Em 2012, Webb, Trefethen e Gonnet [WTG12] mostraram, por meio de experimentos com interpoladores de Lagrange, que a segunda fórmula baricêntrica (2.1) não é estável para extrapolação no plano complexo e, em particular, na reta real. Por outro lado, os resultados apresentados na Seção 5 de [MdC16] e em [Hig04] afirmam que a primeira fórmula baricêntrica (3.5) para o interpolador de Lagrange é backward estável sobre toda a reta real (tomando-se como referência os nós arredondados $\widehat{\mathbf{x}}$.) Neste capítulo apresentamos uma generalização da primeira fórmula baricêntrica (3.5) para o interpolador de Floater-Hormann e um algoritmo backward estável para avalia-la.

A fórmula em questão é

$$p(x, \mathbf{x}, \mathbf{y}, \mu(\mathbf{x})) \quad = \quad \sum_{i=0}^{n} \frac{\mu(\mathbf{x})_i y_i}{x - x_i} \Bigg/ \sum_{i=0}^{n-d} \lambda_{i,d}(x, \mathbf{x}) \ , \tag{5.1}$$

onde as funções $\lambda_i(x, \mathbf{x})$ e os pesos de interpolação $\mu_d(\mathbf{x})_i, i = 0, 1, \ldots, n-d$ são dados por (3.26) e (3.27), respectivamente. A identidade (3.29) mostra que as expressões definidas por (3.28) e (5.1) são idênticas. Observe que, para $d = n$, a fórmula (5.1) coincide com a primeira fórmula baricêntrica (3.5) para o interpolador de Lagrange.

Como estamos essencialmente interessados na comparação entre algoritmos e o erro causado pelo arredondamento dos nós (Erro Tipo II na Figura 4.1) é devido somente às características da máquina e não dos algoritmos em si, assumiremos, na nossa análise, que

$$\mathbf{x} = \widehat{\mathbf{x}} \ \text{ e } \ \mathbf{y} = \widehat{y}.$$

Na prática, iremos manter a notação original e comparar o valor calculado $fl(r_d(x, \widehat{\mathbf{x}}, \widehat{y}))$ com

$$p(x, \widehat{\mathbf{x}}, \widehat{\mathbf{y}}, \mu(\widehat{\mathbf{x}})) \ = \ q(x, \widehat{\mathbf{x}}, \widehat{\mathbf{y}}, \mu(\widehat{\mathbf{x}}))$$

ao invés de (3.28) ou (5.1). A Seção 4 de [MdC16] explica por que isso é razoável também do ponto de vista de aproximação.

5.1 Algoritmos

Nesse trabalho consideramos dois tipos de algoritmos para a avaliação do interpolador de Floater-Hormann:

- **Tipo I:** Utiliza a formula (5.1). Atenção especial é dada ao cálculo do seu denominador.
- **Tipo II:** Utiliza a formula (3.28).

A distinção entre dois algoritmos de um mesmo tipo se dá pelo modo como os pesos de interpolação $\mu(\widehat{\mathbf{x}})$ são calculados. Os erros de arredonadamento oriundos desse processo são mensurados por

$$z_i := z(\widehat{\mu}, \mu(\widehat{\mathbf{x}}))_i = \frac{\widehat{\mu}_i}{\mu(\widehat{\mathbf{x}})_i} - 1, \ i = 0, 1, \ldots, n, \tag{5.2}$$

onde $\widehat{\mu}$ denota o vetor de pesos arredondados que serão efetivamente utilizados na computação.

Dados $x, \widehat{\mathbf{x}}, \widehat{\mathbf{y}}$ e $\widehat{\mu}$, os algoritmos do Tipo II calculam o numerador e o denominador de $q(x, \widehat{\mathbf{x}}, \widehat{\mathbf{y}}, \widehat{\mu})$ com um laço simples e retornam o valor computado do quociente entre eles. Os algoritmos do Tipo I também calculam o numerador de $p(x, \widehat{\mathbf{x}}, \widehat{\mathbf{y}}, \widehat{\mu})$ com um laço simples, porém o denominador $\sum_{i=0}^{n-d} \lambda_i(x, \widehat{\mathbf{x}})$ é calculado como indicado pelo lado direito da igualdade (3.32). Mostramos, na Seção 3.2.2, que todas as parcelas de (3.32) possuem o mesmo sinal e, como observado anteriormente em [Mas14], essa propriedade é útil para construir algoritmos estáveis para a avaliação de funções racionais.

5.2 Análise da estabilidade backward dos algoritmos do Tipo I e do Tipo II

A nossa análise se baseia na decomposição do erro total de interpolação de acordo com o diagrama da Figura 4.1.

5.2.1 Erro no Passo II

Os pesos de interpolação $\widehat{\mu}$ podem ser obtidos, por exemplo, pela avaliação numérica de $\mu(\widehat{\mathbf{x}})$. Porém, em alguns casos de interesse, os pesos $\mu(\mathbf{x})$ possuem formas algébricas fechadas simples que podem ser calculadas numericamente de forma eficiente, como no *caso de Salzer* para interpolação polinomial ($d = n$) abordado no Exemplo 1 do Capítulo 4, ou no caso dos nós igualmente espaçados, cujos pesos simplificados podem ser obtidos por meio da relação (3.41).

Com relação à primeira estratégia de obtenção dos pesos, temos o seguinte

Lema 3. *Se $\widehat{\mu}$ é obtido por meio da avaliação numérica de $\mu(\widehat{\mathbf{x}})$ de acordo com a fórmula (3.27) e a precisão da máquina ϵ é tal que $3d\epsilon < 0.001$, então o vetor de erros relativos $\mathbf{z}$ definido por (5.2) satisfaz*

$$||\mathbf{z}||_\infty \quad \leq \quad 3.03 d\epsilon, \quad d \geq 1. \tag{5.3}$$

A análise é feita somente para $d \geq 1$, pois, para $d = 0$, os pesos dados por (3.27) já são formados por números de ponto flutuante (± 1), os quais não dependem dos nós $\widehat{\mathbf{x}}$.

Demonstração. Cada parcela em (3.27) pode ser calculada com alta precisão relativa. De fato, por (2.20), obtemos

$$fl\left((-1)^j \left[\prod_{\substack{\tau=j \\ \tau \neq i}}^{j+d} \widehat{x}_i - \widehat{x}_\tau\right]^{-1}\right) \quad = \quad (-1)^j \left(\prod_{\substack{\tau=j \\ \tau \neq i}}^{j+d} \frac{1}{\widehat{x}_i - \widehat{x}_\tau}\right) \langle 2d \rangle. \tag{5.4}$$

Observe que o produtório em (5.4) possui exatamente d fatores, pois (3.27) mostra que $j \leq i \leq j+d$.

Como a soma em (3.27) possui no máximo $d+1$ parcelas e todas possuem o mesmo sinal, podemos utilizar a propriedade (2.22), para $m = d$ e $q = 2d$, para concluir que

$$\widehat{\mu}_i \quad = \quad fl\left(\mu(\widehat{\mathbf{x}})_i\right) \quad = \quad \mu(\widehat{\mathbf{x}})_i \langle 3d \rangle.$$

Logo,

$$|z(\widehat{\mu}, \mu(\widehat{\mathbf{x}}))_i| \;=\; \left|\tfrac{\widehat{\mu}_i - \mu(\widehat{\mathbf{x}})_i}{\mu(\widehat{\mathbf{x}})_i}\right| \;=\; |\langle 3d\rangle - 1| \;\overset{(2.19)}{\leq}\; 3.03d\epsilon.$$

□

Com relação à segunda estratégia, se os erros relativos na avaliação numérica dos pesos analíticos $\mu(\mathbf{x})$ são

$$\upsilon_i = \frac{\widehat{\mu}_i}{\mu(\mathbf{x})_i} - 1,\; i = 0, 1, \ldots, n \tag{5.5}$$

e definimos

$$\theta_i = \tfrac{\mu(\mathbf{x})_i}{\mu(\widehat{\mathbf{x}})_i} - 1,\; i = 0, 1, \ldots, n,$$

então

$$z_i \;=\; (1+\theta_i)(1+\upsilon_i) - 1 \;=\; \theta_i + \upsilon_i + \theta_i\upsilon_i, \quad i = 0, 1, \ldots, n. \tag{5.6}$$

Como o uso dos pesos analíticos $\mu(\mathbf{x})$ só é vantajoso quando os erros (5.5) são pequenos, é razoável assumir que o termo dominante em (5.6) é θ_i. De fato, para a segunda fórmula baricêntrica (3.28), por exemplo, os pesos simplificados para nós igualmente espaçados são formados por números inteiros (ver (3.41)) para os quais $\upsilon_i = 0,\; i = 0, 1, \ldots, n$. Temos o seguinte:

Lema 4. *Se*

$$\frac{4||\mathbf{x} - \widehat{\mathbf{x}}||_\infty}{\eta(\mathbf{x})}[1 + \log(d)] \leq 0.1, \tag{5.7}$$

para

$$\eta(\mathbf{x}) := \min_{0 \leq i < n} |x_{i+1} - x_i|, \tag{5.8}$$

então

$$||\boldsymbol{\theta}||_\infty \leq 4.45\frac{||\mathbf{x} - \widehat{\mathbf{x}}||_\infty}{\eta(\mathbf{x})}[1 + \log(d)]. \tag{5.9}$$

Demonstração. Começamos analisando cada parcela de $\mu(\widehat{\mathbf{x}})_i$ em (3.27):

$$w(\widehat{\mathbf{x}})_{i,j} \;:=\; (-1)^j \prod_{\substack{\tau = j \\ \tau \neq i}}^{j+d} \tfrac{1}{\widehat{x}_i - \widehat{x}_\tau}.$$

Para $\max\{0, i-d\} \leq j \leq \min\{n-d, i\}$ e $j \leq \tau \leq j+d$, defina

$$\vartheta(\mathbf{x}, \widehat{\mathbf{x}})_{i,j} \;=\; \frac{w(\mathbf{x})_{i,j}}{w(\widehat{\mathbf{x}})_{i,j}} - 1 \;=\; \left(\prod_{\substack{\tau = j \\ \tau \neq i}}^{j+d} \frac{\widehat{x}_i - \widehat{x}_\tau}{x_i - x_\tau}\right) - 1, \tag{5.10}$$

$$r(\mathbf{x}, \widehat{\mathbf{x}})_{\tau,i} \;=\; \tfrac{\widehat{x}_i - \widehat{x}_\tau}{x_i - x_\tau} - 1 \quad \text{e} \quad s(\mathbf{x}, \widehat{\mathbf{x}})_{i,j} \;=\; \sum_{\substack{\tau = j \\ \tau \neq i}}^{j+d} |r(\mathbf{x}, \widehat{\mathbf{x}})_{\tau,i}|.$$

Por (5.8), vale que

$$|r(\mathbf{x},\widehat{\mathbf{x}})_{\tau,i}| \leq \tfrac{2||\mathbf{x}-\widehat{\mathbf{x}}||_\infty}{|i-\tau|\eta(\mathbf{x})}.$$

Como, também, $\max\{0, i-d\} \leq j \leq \min\{n-d, i\}$ e $j \leq \tau \leq j+d$, obtemos

$$s(\mathbf{x},\widehat{\mathbf{x}})_{i,j} \;\leq\; \tfrac{2||\mathbf{x}-\widehat{\mathbf{x}}||_\infty}{\eta(\mathbf{x})} \sum_{\substack{\tau=j\\ \tau\neq i}}^{j+d} \tfrac{1}{|i-\tau|} \;\leq\; \tfrac{4||\mathbf{x}-\widehat{\mathbf{x}}||_\infty}{\eta(\mathbf{x})}[1+\log(d)] \;\leq\; 0.1. \tag{5.11}$$

Portanto, por (5.7), podemos aplicar o Lema 9 de [MdC16] para concluir que $|\vartheta(\mathbf{x},\widehat{\mathbf{x}})_{i,j}| \leq$

$$s(\mathbf{x},\widehat{\mathbf{x}})_{i,j} + \frac{s(\mathbf{x},\widehat{\mathbf{x}})_{i,j}^2}{1-s(\mathbf{x},\widehat{\mathbf{x}})_{i,j}} \;\leq\; \frac{s(\mathbf{x},\widehat{\mathbf{x}})_{i,j}}{1-s(\mathbf{x},\widehat{\mathbf{x}})_{i,j}} \overset{(5.7),(5.11)}{\leq} 4.45\frac{||\mathbf{x}-\widehat{\mathbf{x}}||_\infty}{\eta(\mathbf{x})}[1+\log(d)] \;\leq\; 0.445. \tag{5.12}$$

Voltando para (5.10), escrevemos

$$\mu(\mathbf{x})_i = \sum_{j=\max\{0,i-d\}}^{\min\{n-d,i\}} w(\widehat{\mathbf{x}})_{i,j}[1+\vartheta(\mathbf{x},\widehat{\mathbf{x}})_{i,j}] \tag{5.13}$$

e, como todas as parcelas de (5.13) possuem o mesmo sinal (ver (3.27)), segue que

$$\mu(\mathbf{x})_i \;=\; [1+\rho_i] \sum_{j=\max\{0,i-d\}}^{\min\{n-d,i\}} w(\widehat{\mathbf{x}})_{i,j} \;=\; [1+\rho_i]\mu(\widehat{\mathbf{x}})_i,$$

para algum número real ρ_i tal que $|\rho_i| \leq \max\limits_{\max\{0,i-d\}\leq j\leq \min\{n-d,i\}} |\vartheta(\mathbf{x},\widehat{\mathbf{x}})_{i,j}|$.
Portanto, por (5.12), obtemos

$$\max_i \left|\tfrac{\mu(\mathbf{x})_i}{\mu(\widehat{\mathbf{x}})_i} - 1\right| \leq 4.45\tfrac{||\mathbf{x}-\widehat{\mathbf{x}}||_\infty}{\eta(\mathbf{x})}[1+\log(d)].$$

□

Observação 3. *Os resultados deste capítulo foram publicados no artigo [dC15]. Em [dC15], cometemos um pequeno erro na estimativa (5.11). Dessa forma, os fatores 2δ e 2.24δ no enunciaodo do Lema 2 em [dC15] devem ser substituídos por 4δ e 4.45δ, respectivamente.*

Para algumas familias de nós para as quais o espaçamento máximo e mínimo entre os nós é bem diferente, a estimativa (5.9) pode ser melhorada. Para os nós de Chebyshev do segundo tipo, por exemplo, o fator $[1+\log(d)]$ poderia ser substituído por uma constante. De fato, observando a relação

$$\theta_i = -\tfrac{\zeta_i^{sal}}{1+\zeta_i^{sal}}$$

entre $\boldsymbol{\theta}$ e $\boldsymbol{\zeta}^{\boldsymbol{sal}}$ (definido no Exemplo 1) e a inequação (4.16) obtemos

$$||\boldsymbol{\theta}||_\infty \leq 1.12||\boldsymbol{\zeta}^{sal}||_\infty \leq 10.08n^2\epsilon,$$

para $||\zeta^{sal}||_\infty \leq 0.1$, e vale que

$$\eta(\mathbf{x_{cheb2}}) \;=\; \left|\cos\left(\tfrac{\pi}{n}\right) - \cos\left(\tfrac{0\pi}{n}\right)\right| \;=\; \left|2\sin\left(\tfrac{\pi}{2n}\right)^2\right| \;\leq \tfrac{1}{2n^2}.$$

Os valores apresentados nas Tabelas 5.1 e 5.2 mostram que a estimativa (5.9) fornece a ordem correta para $||\boldsymbol{\theta}||_\infty$. Consequentemente, para os nós igualmente espaçados, a primeira estratégia é qualitativamente melhor, pois nesse, caso a estimativa superior (5.3) é apenas $O(d\epsilon)$, enquanto que a estimativa superior (5.9) é $O(n\epsilon[1+\log(d)])$, para $||\mathbf{x}-\widehat{\mathbf{x}}||_\infty \approx \epsilon$. Essa discrepância aumenta dramaticamente no caso dos nós de Chebyshev do segundo tipo definidos em $[a,b]$, para os quais $\eta(\mathbf{x})$ é $O\left(\frac{b-a}{n^2}\right)$. No entanto, enfatizamos que os valores da Tabela 5.2 são meramente ilustrativos, pois não há formas analíticas simples conhecidas para os pesos associados aos nós de Chebyshev do segundo tipo para $d<n$. O caso de interesse ($d=n$) abordado no Exemplo 1, para o qual $\theta \approx \zeta^{\mathbf{sal}}$, é analisado em detalhe em [MdC16].

$d \setminus n$	10^2	10^3+1	10^4	10^5+1	10^6
1	$9.33\ 10^{-2}$	$6.74\ 10^{-2}$	$8.90\ 10^{-2}$	$1.09\ 10^{-1}$	$5.75\ 10^{-1}$
4	$1.00\ 10^{-1}$	$6.66\ 10^{-2}$	$9.35\ 10^{-2}$	$1.29\ 10^{-1}$	$7.31\ 10^{-2}$
7	$1.15\ 10^{-1}$	$7.31\ 10^{-2}$	$9.10\ 10^{-2}$	$1.28\ 10^{-1}$	$7.94\ 10^{-2}$
10	$1.70\ 10^{-1}$	$7.00\ 10^{-2}$	$8.47\ 10^{-2}$	$1.31\ 10^{-1}$	$7.79\ 10^{-2}$
13	$1.13\ 10^{-2}$	$6.97\ 10^{-2}$	$7.86\ 10^{-2}$	$1.30\ 10^{-1}$	$7.93\ 10^{-2}$

Tabela 5.1: *Valores de* $\max\limits_{0\le i\le n} \frac{|\theta_i|}{n||\mathbf{x}-\widehat{\mathbf{x}}||_\infty(1+\log(d))}$ *para nós igualmente espaçados definidos em* $[-1,1]$*;* $||\mathbf{x}-\widehat{\mathbf{x}}||_\infty := 10^{-15}$.

$d \setminus n$	10^2	10^3+1	10^4	10^5+1	10^6
1	$1.74\ 10^{-2}$	$2.02\ 10^{-2}$	$1.69\ 10^{-2}$	$1.02\ 10^{-2}$	$1.40\ 10^{-2}$
2	$1.47\ 10^{-2}$	$2.37\ 10^{-2}$	$1.24\ 10^{-2}$	$8.65\ 10^{-3}$	$1.31\ 10^{-2}$
3	$1.40\ 10^{-2}$	$2.38\ 10^{-2}$	$1.25\ 10^{-2}$	$7.88\ 10^{-3}$	$1.42\ 10^{-2}$
4	$1.30\ 10^{-2}$	$2.37\ 10^{-2}$	$1.27\ 10^{-2}$	$7.78\ 10^{-3}$	$1.42\ 10^{-2}$
5	$1.27\ 10^{-2}$	$2.39\ 10^{-2}$	$1.24\ 10^{-2}$	$7.59\ 10^{-3}$	$1.42\ 10^{-2}$

Tabela 5.2: *Valores de* $\max\limits_{0\le i\le n} \frac{|\theta_i|}{||\mathbf{x}-\widehat{\mathbf{x}}||_\infty}\frac{2}{n^2}$ *para os nós de Chebyshev do segundo tipo;* $||\mathbf{x}-\widehat{\mathbf{x}}||_\infty := 10^{-15}$.

Os valores apresentados nas Tabelas 5.1 e 5.2 foram obtidos com o auxílio da biblioteca MPFR. Os nós $\mathbf{x}$ foram calculados com precisão de 30 casas decimais. Os nós $\widehat{\mathbf{x}}$ foram calculados em precisão dupla ($\approx$ 16 casas decimais) e transformados para números com 30 casas decimais. Todos os cálculos intermediários foram realizados com precisão de 30 casas decimais e o resultado final foi arredondado para precisão dupla.

5.2.2 Erro no Passo III

Teorema 7 (Estabilidade backward dos algoritmos do Tipo I)**.** *Se x é um número de ponto flutuante e a precisão da máquina ϵ é tal que*

$$\left(\frac{3n+5d+1}{2}+11\right)\epsilon < 0.001, \tag{5.14}$$

então o valor calculado $fl(p(x,\widehat{\mathbf{x}},\widehat{\mathbf{y}},\widehat{\mu}))$ de $p(x,\widehat{\mathbf{x}},\widehat{\mathbf{y}},\mu(\widehat{\mathbf{x}}))$ é igual a $p(x,\widehat{\mathbf{x}},\widetilde{\mathbf{y}},\mu(\widehat{\mathbf{x}}))$, para algum vetor $\widetilde{\mathbf{y}} \in \mathbb{R}^{n+1}$ tal que

$$\widetilde{y}_i \;=\; \widehat{y}_i\,(1+z_i)\,(1+\kappa_i),$$

com $\mathbf{z}$ definido em (5.2) e

$$|\kappa_i| \le 1.01\left(\tfrac{3n+5d+1}{2}+11\right)\epsilon,\ \ i=0,1,\ldots,n.$$

Demonstração. Vamos iniciar analisando o cálculo do denominador (3.32). Temos que

$$fl(\xi_i(x,\widehat{\mathbf{x}})) \;=\; fl\left((-1)^i fl\left(\left[\prod_{j=i+1}^{i+d} x-\widehat{x}_j\right]^{-1}\right) fl\left(\tfrac{\widehat{x}_i\;-\;\widehat{x}_{i+d+1}}{(x-\widehat{x}_i)(x-\widehat{x}_{i+d+1})}\right)\right).$$

Como

$$fl\left(\left[\prod_{j=i+1}^{i+d} x-\widehat{x}_j\right]^{-1}\right) \overset{(2.20)}{=} \left(\prod_{j=i+1}^{i+d} \tfrac{1}{(x-\widehat{x}_j)}\right)\langle 2d\rangle_i$$

e

$$fl\left(\tfrac{\widehat{x}_i\;-\;\widehat{x}_{i+d+1}}{(x-\widehat{x}_i)(x-\widehat{x}_{i+d+1})}\right) \overset{(2.20)}{=} \left(\tfrac{\widehat{x}_i\;-\;\widehat{x}_{i+d+1}}{(x-\widehat{x}_i)(x-\widehat{x}_{i+d+1})}\right)\langle 5\rangle_i,$$

$0\le i\le n-d-1$, então $fl(\xi_i(x,\widehat{\mathbf{x}})) \overset{(2.16),(2.17)}{=}$

$$(-1)^i\left[\left(\prod_{j=i+1}^{i+d}\tfrac{1}{x-\widehat{x}_j}\right)\tfrac{\widehat{x}_i\;-\;\widehat{x}_{i+d+1}}{(x-\widehat{x}_i)(x-\widehat{x}_{i+d+1})}\right]\langle 2d+6\rangle_i \;=\; \xi_i(x,\widehat{\mathbf{x}})\langle 2d+6\rangle_i. \tag{5.15}$$

Analogamente, obtemos

$$fl(\xi_i(x,\widehat{\mathbf{x}})) = \xi_i(x,\widehat{\mathbf{x}})\langle 2(d+1)\rangle_i,\;\; i=-1,n-d \tag{5.16}$$

e

$$fl(\lambda_i(x,\widehat{\mathbf{x}})) = \lambda_i(x,\widehat{\mathbf{x}})\langle 2(d+1)\rangle_i,\;\; i=0,1,\ldots,n-d. \tag{5.17}$$

Assim, por (2.18), (3.32), (5.15), (5.16) e (5.17), obtemos $fl\left(\sum_{i=0}^{n-d}\lambda_i(x,\widehat{\mathbf{x}})\right) =$

$$fl\Bigg(\delta_{k-d}\,\xi_{-1}(x,\hat{\mathbf{x}})\langle 2d+6\rangle_{-1} + \sum_{0\le k-d-2j-1\le k-d}\xi_{k-d-2j-1}(x,\widehat{\mathbf{x}})\langle 2d+6\rangle_{k-d-2j-1} \;+$$

$$\sum_{k-d<i\le k}\lambda_i(x,\widehat{\mathbf{x}})\langle 2d+6\rangle_i \;+\; \sum_{k<k+1+2j\le n-d}\xi_{k+1+2j}(x,\widehat{\mathbf{x}})\langle 2d+6\rangle_{k+1+2j}\Bigg),$$

para $x_k<x<x_{k+1}$. Como não há mais que $d+\left\lfloor\frac{n+1-d}{2}\right\rfloor+2$ parcelas na expressão acima e todas possuem o mesmo sinal (ver Seção (3.2.2)), podemos aplicar (2.22) e (2.18) para obter

$$fl\left(\sum_{i=0}^{n-d}\lambda_i(x,\widehat{\mathbf{x}})\right) = \left(\sum_{i=0}^{n-d}\lambda_i(x,\widehat{\mathbf{x}})\right)\left\langle 3d+7+\left\lfloor\frac{n+1-d}{2}\right\rfloor\right\rangle_1. \tag{5.18}$$

Agora, voltando as atenções para o numerador de (5.1), temos

$$fl\left(\tfrac{\widehat{\mu}_i\widehat{y}_i}{x-\widehat{x}_i}\right) \;=\; \left(\tfrac{\widehat{\mu}_i\widehat{y}_i}{x-\widehat{x}_i}\right)\langle 3\rangle_i,\;\; i=0,1,\ldots,n$$

e, portanto,

$$fl\left(\sum_{i=0}^{n}\tfrac{\widehat{\mu}_i\widehat{y}_i}{x-\widehat{x}_i}\right) \overset{(2.21)}{=} \sum_{i=0}^{n}\left(\tfrac{\widehat{\mu}_i\widehat{y}_i}{x-\widehat{x}_i}\right)\langle n+3\rangle_i.$$

Logo,

$$
\begin{aligned}
fl(p(x,\widehat{\mathbf{x}},\widehat{\mathbf{y}},\mu(\widehat{\mathbf{x}}))) &= fl\left(\frac{fl\left(\sum\limits_{i=0}^{n}\frac{\widehat{\mu}_i\widehat{y}_i}{x-\widehat{x}_i}\right)}{fl\left(\sum\limits_{i=0}^{n-d}\lambda_i(x,\widehat{\mathbf{x}})\right)}\right) \overset{(2.16),(2.17)}{=} \frac{\sum\limits_{i=0}^{n}\frac{\widehat{\mu}_i}{x-\widehat{x}_i}\widehat{y}_i\langle 1\rangle\langle n+3\rangle_i\left\langle 3d+7+\left\lfloor\frac{n+1-d}{2}\right\rfloor\right\rangle_2}{\sum\limits_{i=0}^{n-d}\lambda_i(x,\widehat{\mathbf{x}})} \\
&= \frac{\sum\limits_{i=0}^{n}\frac{\mu(\widehat{\mathbf{x}})_i}{x-\widehat{x}_i}\widetilde{y}_i}{\sum\limits_{i=0}^{n-d}\lambda_i(x,\widehat{\mathbf{x}})} = p(x,\widehat{\mathbf{x}},\widetilde{\mathbf{y}},\mu(\widehat{\mathbf{x}})),
\end{aligned}
$$

para

$$
\widetilde{y}_i = \widehat{y}_i\frac{\widehat{\mu}_i}{\mu(\widehat{\mathbf{x}})_i}\left\langle n+3d+11+\left\lfloor\frac{n+1-d}{2}\right\rfloor\right\rangle_i.
$$

Definindo $\kappa_i = \left\langle n+3d+11+\left\lfloor\frac{n+1-d}{2}\right\rfloor\right\rangle_i - 1$, então (5.14) e (2.19) mostram que

$$
|\kappa_i| \leq 1.01\left(\tfrac{3n+5d+1}{2}+11\right)\epsilon, i=0,1,\ldots,n
$$

e isso completa a prova. □

Corolário 2 (Avaliação estável da função de Lebesgue). *Sob as hipóteses do Teorema 7, se $x \in [a,b]$ é um número de ponto flutuante e o vetor $\mathbf{z}$ em (5.2) satisfaz*

$$
||\mathbf{z}||_\infty < 1, \tag{5.19}
$$

então a função de Lebesgue (2.6) pode ser avaliada de forma que

$$
\left|\frac{fl(L(x,\widehat{\mathbf{x}},a,b))}{L(x,\widehat{\mathbf{x}},a,b)}-1\right| \leq 1.01(1+||\mathbf{z}||_\infty)\left(\frac{3n+5d+1}{2}+11\right)\epsilon + ||\mathbf{z}||_\infty. \tag{5.20}
$$

Demonstração. Para cada número de ponto flutuante $x \in [a,b]$, podemos definir um vetor $\widehat{\mathbf{y}}$, $\widehat{y}_i \in \{-1,1\}, i=0,1,\ldots,n$ tal que

$$
L(x,\widehat{\mathbf{x}},a,b) = p(x,\widehat{\mathbf{x}},\widehat{\mathbf{y}},\mu(\widehat{\mathbf{x}})). \tag{5.21}
$$

Assim, aplicando o Teorema 7, obtemos

$$
fl(L(x,\widehat{\mathbf{x}},a,b)) = fl(p(x,\widehat{\mathbf{x}},\widehat{\mathbf{y}},\widehat{\mu})) = p(x,\widehat{\mathbf{x}},\widetilde{\mathbf{y}},\mu(\widehat{\mathbf{x}})), \tag{5.22}
$$

para algum vetor $\widetilde{\mathbf{y}} \in \mathbb{R}^{n+1}$ tal que

$$
\widetilde{y}_i = \widehat{y}_i\,(1+z_i)\,(1+\kappa_i),
$$

com

$$
|\kappa_i| \leq 1.01\left(\tfrac{3n+5d+1}{2}+11\right)\epsilon,\ i=0,1,\ldots,n.
$$

Além disso, as condições (5.14) e (5.19) implicam que $\widetilde{y}_i$ e $\widehat{y}_i$ possuem o mesmo sinal para todos os índices i. Consequentemente, todas as parcelas do numerador da expressão do lado esquerdo de (5.21) são positivas, e

$$
\min_{0\leq i\leq n}\frac{\widetilde{y}_i}{\widehat{y}_i} = \min_{0\leq i\leq n}|\widetilde{y}_i| \leq \frac{fl(p(x,\widehat{\mathbf{x}},\widehat{\mathbf{y}},\widehat{\mu}))}{p(x,\widehat{\mathbf{x}},\widehat{\mathbf{y}},\widehat{\mu})} \leq \max_{0\leq i\leq n}|\widetilde{y}_i| = \max_{0\leq i\leq n}\frac{\widetilde{y}_i}{\widehat{y}_i}
$$

e (5.20) segue de (5.21) e (5.22). □

A estabilidade backward para os algoritmos do Tipo II segue diretamente do Teorema 6 no Capítulo 4, tomando-se $\delta = 0$.

Teorema 8 (Estabilidade backward para algoritmos do Tipo II). *Se os vetores $\widehat{\mathbf{x}}$, $\widehat{\mu}$ e a precisão da máquina ϵ são tais que*

$$(2n+6)\epsilon \leq 0.001$$

e

$$Z(\Lambda(\widehat{\mathbf{x}}, \mu(\widehat{\mathbf{x}})) + 1) < 1,$$

para

$$Z := \frac{||\zeta||_\infty + (n+2)\epsilon}{1-(n+2)\epsilon}, \quad e \quad \zeta_k := \frac{-z_k}{1+z_k}, \; k = 0, 1, \ldots, n, \tag{5.23}$$

então o valor calculado $fl(q(x, \widehat{\mathbf{x}}, \widehat{\mathbf{y}}, \widehat{\mu}))$ *de* $q(x, \widehat{\mathbf{x}}, \widehat{\mathbf{y}}, \mu(\widehat{\mathbf{x}}))$ *é igual a* $q(x, \widehat{\mathbf{x}}, \widetilde{\mathbf{y}}, \mu(\widehat{\mathbf{x}}))$*, para algum vetor* $\widetilde{\mathbf{y}} \in \mathbb{R}^{n+1}$ *tal que*

$$\widetilde{y}_i = \widehat{y}_i(1+\alpha_i)(1+\nu_i), \; i = 0, 1, \ldots n,$$

com

$$||\alpha||_\infty \leq \frac{Z[1+\Lambda(\widehat{\mathbf{x}}, \mu(\widehat{\mathbf{x}}))]}{1-Z[\Lambda(\widehat{\mathbf{x}}, \mu(\widehat{\mathbf{x}}))+1]} \quad e \quad ||\nu||_\infty \leq 1.01(2n+6)\epsilon.$$

5.3 Análise da estabilidade forward dos algoritmos do Tipo I e do Tipo II

Por estabilidade backward de um algoritmo do Tipo I ou do Tipo II, queremos dizer que, para cada conjunto de parâmetros compostos por números de ponto flutuante $\widehat{\mathbf{x}}$, $\widehat{\mathbf{y}}$ e $x \in [a, b]$ há um vetor $\boldsymbol{\beta} \in \mathbb{R}^{n+1}$ (pequeno em magnitude e que depende de x, $\widehat{\mathbf{x}}$ e $\widehat{\mathbf{y}}$) tal que o valor final calculado $fl(u(x, \widehat{\mathbf{x}}, \widehat{\mathbf{y}}, \mu(\widehat{\mathbf{x}})))$ de $u(x, \widehat{\mathbf{x}}, \widehat{\mathbf{y}}, \mu(\widehat{\mathbf{x}}))$ satisfaz

$$fl(u(x, \widehat{\mathbf{x}}, \widehat{\mathbf{y}}, \mu(\widehat{\mathbf{x}}))) = u(x, \widehat{\mathbf{x}}, \widetilde{\mathbf{y}}, \mu(\widehat{\mathbf{x}})), \quad \widetilde{y}_i \; = \; (1+\beta)\widehat{y}_i, \quad i = 0, 1, \ldots, n \tag{5.24}$$

($u = p$ para um algoritmo do Tipo I e $u = q$ para um algoritmo do Tipo II.) Como de praxe, podemos limitar o erro forward em função do erro backward e da constante de Lebesgue de acordo com (2.7), isto é, $|fl(u(x, \widehat{\mathbf{x}}, \widehat{\mathbf{y}}, \mu(\widehat{\mathbf{x}}))) - u(x, \widehat{\mathbf{x}}, \widehat{\mathbf{y}}, \mu(\widehat{\mathbf{x}}))| \; =$

$$|u(x, \widehat{\mathbf{x}}, \widetilde{\mathbf{y}}, \mu(\widehat{\mathbf{x}})) - u(x, \widehat{\mathbf{x}}, \widehat{\mathbf{y}}, \mu(\widehat{\mathbf{x}}))| \; \leq \; \Lambda(\widehat{\mathbf{x}}, \mu(\widehat{\mathbf{x}})) \; ||\widetilde{\mathbf{y}} - \widehat{\mathbf{y}}||_\infty \; \leq \; \Lambda(\widehat{\mathbf{x}}, \mu(\widehat{\mathbf{x}})) \; ||\boldsymbol{\beta}||_\infty ||\widehat{\mathbf{y}}||_\infty. \tag{5.25}$$

O Teorema 7 afirma que o vetor de erros relativos $\boldsymbol{\beta}$ sempre existe para os algoritmos do Tipo I e fornece uma estimativa superior

$$O(||\mathbf{z}||_\infty + n\epsilon) \tag{5.26}$$

para $||\boldsymbol{\beta}||_\infty$. Quando a constante de Lebesgue $\Lambda(\widehat{\mathbf{x}}, \mu(\widehat{\mathbf{x}}))$ é pequena, o vetor $\boldsymbol{\beta}$ também existe para os algoritmos do Tipo II e, nesse caso, o Teorema 8 fornece uma estimativa superior

$$O(\Lambda(\widehat{\mathbf{x}}, \mu(\widehat{\mathbf{x}}))(||\mathbf{z}||_\infty + n\epsilon)) \tag{5.27}$$

para $||\boldsymbol{\beta}||_\infty$, conforme discutido na Seção 4.1.2.

Como $\Lambda(\widehat{\mathbf{x}}, \mu(\widehat{\mathbf{x}})) \geq 1$, as relações (5.25), (5.26) e (5.27) podem dar a impressão de que os algoritmos do Tipo I são mais forward estáveis do que os algoritmos do Tipo II com mesmos parâmetros ($\widehat{\mathbf{x}}, \widehat{\mathbf{y}}$, x e $\widehat{\mu}$). Porém, a análise da estabilidade forward é mais sutíl, por que um erro backward grande não necessariamente leva à um erro forward grande. O exemplo mais simples desse fenômeno é a interpolação da função constante $f(x) \equiv 1$ (ver nota de rodapé na página A3011 de [WTG12]) para o qual o valor calculado de (2.1) é sempre idêntico a 1, não importando o quão

grande é o erro cometido na avaliação do numerador e do denominador de (2.1). Além disso, Higham [Hig04] apresentou as seguintes estimativas para os erros relativos forward

$$er_1[\widehat{\mathbf{y}}](x) = \frac{fl(p(x,\widehat{\mathbf{x}},\widehat{\mathbf{y}},\widehat{\mu}))}{p(x,\widehat{\mathbf{x}},\widehat{\mathbf{y}},\widehat{\mu})} - 1 \quad \text{e} \quad er_2[\widehat{\mathbf{y}}](x) = \frac{fl(q(x,\widehat{\mathbf{x}},\widehat{\mathbf{y}},\widehat{\mu}))}{q(x,\widehat{\mathbf{x}},\widehat{\mathbf{y}},\widehat{\mu})} - 1 \tag{5.28}$$

para os algoritmos dos Tipos I e II para o interpolador de Lagrange ($d = n$):

$$|er_1[\widehat{\mathbf{y}}](x)| \;\leq\; 1.01(5n+5)\epsilon\; cond(x,n,\widehat{\mathbf{y}}), \quad n\epsilon < 0.001$$

e

$$|er_2[\widehat{\mathbf{y}}](x)| \;\leq\; (3n+4)\epsilon\; cond(x,n,\widehat{\mathbf{y}}) \;+\; (3n+2)\epsilon\; cond(x,n,\mathbb{1}) + O\left(\epsilon^2\right), \tag{5.29}$$

$\mathbb{1} = (1,1,\ldots,1)$, $cond(x,n,\widehat{\mathbf{y}}) \;:=\; \dfrac{\sum_{i=0}^{n}\left|\frac{\mu(\widehat{\mathbf{x}})_i \hat{y}_i}{x-x_i}\right|}{\left|\sum_{i=0}^{n}\frac{\mu(\widehat{\mathbf{x}})_i \hat{y}_i}{x-x_i}\right|}$ e Celis [Cel08] obteve uma relação similar para o erro forward da fórmula baricêntrica genérica (2.1) na sua Tese de doutorado. Dessa forma, como

$$|er_1[\widehat{\mathbf{y}}](x)| \overset{(5.24)}{\leq} ||\boldsymbol{\beta}_{\widehat{\mathbf{y}}}||_\infty\; cond(x,n,\widehat{\mathbf{y}}) \quad \text{com} \quad ||\boldsymbol{\beta}_{\widehat{\mathbf{y}}}||_\infty \overset{(5.26)}{=} O(||\mathbf{z}||_\infty + n\epsilon), \tag{5.30}$$

a relação (5.29) sugere que os algoritmos do Tipo II são significativamente menos estáveis do que os algoritmos do Tipo I somente nos casos em que $|er_1[\mathbb{1}](x)| >> |er_1[\widehat{\mathbf{y}}](x)|$.

Ainda assim, a relação (5.29) é viezada em favor dos algoritmos do Tipo I, pois não permite identificar as situações nas quais os algoritmos do Tipo II são mais estáveis do que os algoritmos do Tipo I. Para preencher essa lacuna, apresentamos a seguir uma versão de (5.29) para d e n quaisquer

Teorema 9. *Se os pesos arredondados $\widehat{\mu}$ são utilizados por ambos os algoritmos do Tipo I e do Tipo II, $er_2[\widehat{\mathbf{y}}](x) \neq -1$ e a precisão da máquina ϵ é tal que*

$$3\epsilon \leq 0.001,$$

então os erros relativos definidos em (5.28) satisfazem

$$\frac{er_1[\widehat{\mathbf{y}}](x)+1}{er_2[\widehat{\mathbf{y}}](x)+1} \;=\; 1 \;+\; er_1[\mathbb{1}](x) \;+\; \sigma, \tag{5.31}$$

com

$$|\sigma| \;\leq\; 3.03\epsilon(1+|er_1[\mathbb{1}](x)|).$$

Demonstração. Por (5.28) e pelas propriedades (2.16) e (2.17), temos $\frac{er_1[\widehat{\mathbf{y}}](x)+1}{er_2[\widehat{\mathbf{y}}](x)+1} =$

$$\frac{fl(p(x,\widehat{\mathbf{x}},\widehat{\mathbf{y}},\widehat{\mu}))}{fl(q(x,\widehat{\mathbf{x}},\widehat{\mathbf{y}},\widehat{\mu}))} \;=\; \frac{\langle 1\rangle_1\; fl\left(\sum_{i=0}^{n}\frac{\mu(\widehat{\mathbf{x}})_i y_i}{x-\widehat{x}_i}\right)\Big/ fl\left(\sum_{i=0}^{n-d}\lambda_i(x,\widehat{\mathbf{x}})\right)}{\langle 1\rangle_2\; fl\left(\sum_{i=0}^{n}\frac{\mu(\widehat{\mathbf{x}})_i y_i}{x-\widehat{x}_i}\right)\Big/ fl\left(\sum_{i=0}^{n}\frac{\mu(\widehat{\mathbf{x}})_i}{x-\widehat{x}_i}\right)} \;=\; \langle 2\rangle_3 \frac{fl\left(\sum_{i=0}^{n}\frac{\mu(\widehat{\mathbf{x}})_i}{x-\widehat{x}_i}\right)}{fl\left(\sum_{i=0}^{n-d}\lambda_i(x,\widehat{\mathbf{x}})\right)}.$$

De forma análoga, (2.16) mostra que

$$1 + er_1[\mathbb{1}](x) \;=\; \langle 1\rangle_4 \frac{fl\left(\sum_{i=0}^{n}\frac{\mu(\widehat{\mathbf{x}})_i}{x-\widehat{x}_i}\right)}{fl\left(\sum_{i=0}^{n-d}\lambda_i(x,\widehat{\mathbf{x}})\right)}.$$

Portanto,

$$\frac{er_1[\widehat{\mathbf{y}}](x)+1}{er_2[\widehat{\mathbf{y}}](x)+1} \overset{(2.17)}{=} \langle 3\rangle_5(1+er_1[\mathbb{1}](x)) \;=1\;+\;er_1[\mathbb{1}](x)\;+\;\sigma,$$

com

$$\sigma \;=\; (\langle 3\rangle_5-1)(1+er_1[\mathbb{1}](x)) \;\text{ e }\; |\sigma| \overset{(2.19)}{\leq} 3.03\epsilon(1+|er_1[\mathbb{1}](x)|).$$

□

Observe que, no caso $\widehat{\mathbf{y}}=\mathbb{1}$ mencionado após a equação (5.27), (5.31) nos mostra claramente que $er_2[\widehat{\mathbf{y}}](x)$ deve ser bem pequeno.

Uma estimativa análoga à (5.29) pode ser obtida por (5.30) e (5.31), rearranjando os termos em (5.31)

$$er_2[\widehat{\mathbf{y}}](x) \;=\; \frac{er_1[\widehat{\mathbf{y}}](x)-\sigma-er_1[\mathbb{1}](x)}{1\;+\;er_1[\mathbb{1}](x)\;+\;\sigma} \tag{5.32}$$

e utilizando a expansão de Taylor $\frac{1}{1+t}=1-t+\frac{2}{(1+\xi)^3}\,t^2,\;\; 0<\xi<t,$ para limitar o denominador em (5.32).

5.4 Experimentos numéricos

Nessa seção apresentamos os resultados de diversos experimentos numéricos para mostrar ao leitor como a teoria desenvolvida nas duas seções anteriores se aplica em situações concretas.

Nos experimentos descritos a seguir, $\widehat{\mathbf{x}},\widehat{\mathbf{y}},\widehat{\mu}$ e x correspondem a números de ponto flutuante IEEE 754 padrão (precisão dupla). Os experimentos foram realizados com dois algoritmos fixos (um do Tipo I e um do Tipo II) para os quais $\widehat{\mu}$ é calculado com erro relativo (5.2) satisfazendo $||\mathbf{z}||_\infty \leq 3.03d\epsilon$ (veja o Lema 3 na Seção 5.2.) O valor calculado (fl) de uma expressão matemática foi obtido com precisão dupla e o seu valor exato foi obtido avaliando-a com aritmética de alta precisão (50 casas decimais) com o auxílio da biblioteca MPFR.

Observação 4. *Na nossa implementação dos algoritmos do Tipo I nós utilizamos (e recomendamos fortemente o uso) a estratégia proposta na Seção 3.1 de [Mas14] para lidar com underflow/overflow no cálculo de produtos com um número grande de fatores.*

5.4.1 Interpolação dos polinômios Lagrange

Análogo ao desenvolvimento da Seção 4.1.2, podemos determinar o erro backward por meio da relação (4.31)

$$\beta_j = \frac{fl(u(x,\widehat{\mathbf{x}},\mathbf{e}^{(j)},\mu(\widehat{\mathbf{x}})))}{u(x,\widehat{\mathbf{x}},\mathbf{e}^{(j)},\mu(\widehat{\mathbf{x}}))}-1, \tag{5.33}$$

quando o vetor de valores observados $\mathbf{y}$ satisfaz $\widehat{\mathbf{y}}=\mathbf{e}^{(j)}$, para

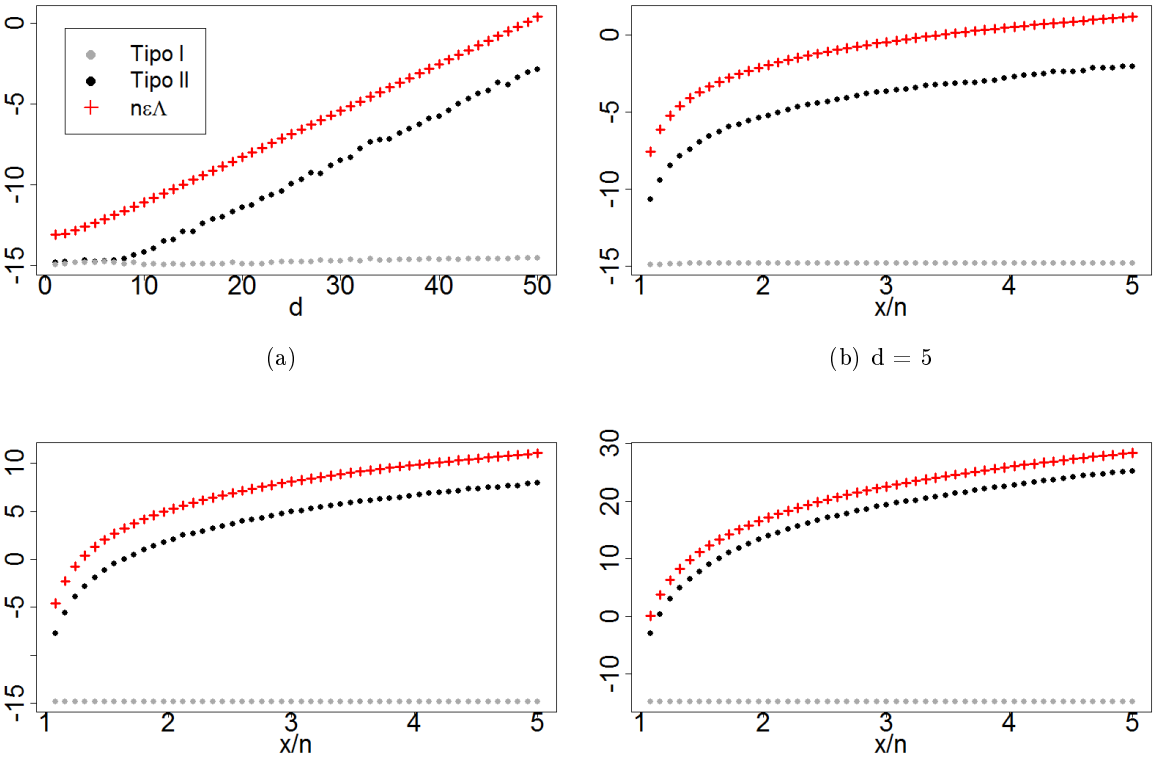
$$e_j^{(j)}=1,\;\; e_k^{(j)}=0,\;\; k\neq j$$

(esse caso corresponde à interpolação do j-ésimo polinômio de Lagrange com nós $\widehat{\mathbf{x}}$.) Quando a fração em (5.33) é muito pequena, então a magnitude do lado direito de (5.33) indicará um erro relativo de cerca de 100% e a verdadeira relação entre os valores calculados e exato de $u(x,\widehat{\mathbf{x}},\mathbf{e}^{(j)},\mu(\widehat{\mathbf{x}}))$ ($u=p$ ou $u=q$) não será devidamente representada. Nesse caso, a quantidade

$$\beta_j^* = -\frac{\beta_j}{1+\beta_j} = \frac{u(x,\widehat{\mathbf{x}},\mathbf{e}^{(j)},\mu(\widehat{\mathbf{x}}))}{fl(u(x,\widehat{\mathbf{x}},\mathbf{e}^{(j)},\mu(\widehat{\mathbf{x}})))}-1 \tag{5.34}$$

é mais indicada para descrever o fenômeno.

No experimento descrito abaixo, calculamos as expressões dadas pelos lados direitos de (5.33) e (5.34) para medir a magnitude dos vetores de erros relativos $\boldsymbol{\beta}$ e $\boldsymbol{\beta}^*$ na avaliação numérica dos algoritmos dos Tipos I e II. Escolhemos $n+1 = 101$ nós igualmente espaçados em $[0, n]$ (de modo a ter nós formados por números inteiros) e calculamos o valor absoluto das expressões em (5.33) e (5.34) para diversos valores de x e para $j \in I = \{0, 9, 18, 27, 36, 45\}$. Os resulados são mostrados na Figura 5.1, em escala logaritmica (base = 10.)



(a) (b) d = 5

(c) d = 10 (d) d = 20

Figura 5.1: *Valor absoluto do erro backward (em escala logaritmica)* $\max_{j \in I} \log_{10}(|\beta_j|)$ *para interpolação (a) e o erro forward reverso* $\max_{j \in I} \log_{10}(|\beta_j^*|)$ *para extrapolação (b), (c) e (d).*

O gráfico (a) mostra o comportamento do erro relativo backward (5.33) para interpolação em função do parâmetro d que define a ordem de convergência do interpolador de Floater-Hormann. Para cada valor de $d, 1 \leq d \leq 50$, o valor plotado corresponde ao máximo valor absoluto do lado direito de (5.33) sobre 10000 pontos igualmente espaçados x em $[0, n]$ e sobre $j \in I$. Os valores neste gráfico mostram que o erro backward para o algoritmo do Tipo II cresce exponencialmente com relação à d e sugere que a estimativa superior (5.27) fornece, de fato, a ordem correta da magnitude desses erros neste caso. Esses valores também mostram que o erro relativo backward para o algoritmo do Tipo I não depende da constante de Lebesgue, como já havíamos comentado em (5.26).

Em cada um dos gráficos (b), (c) e (d), observa-se um comportamento similar, porém agora para extrapolação sobre o intervalo $[n, 5n]$, para valores fixos de d ($d = 5$ no gráfico (b), $d = 10$ no gráfico (c) e $d = 20$ no gráfico (d).) Os valores das abscissas nesses gráficos correspondem a 51 nós igualmente espaçados $n = t_0, t_1, \ldots, t_{50} = 5n$ em $[n, 5n]$ e a ordenada correspondente à abscissa t_k ($1 \leq k \leq 50$) é o máximo valor absoluto do lado direito de (5.34) sobre 1001 nós igualmente espaçados x em $[t_{k-1}, t_k]$ e sobre $j \in I$. A curva superior nos gráficos (b), (c) e (d) são formadas pelos valores correspondentes da constante de Lebesgue (sobre $[t_{k-1}, t_k]$) vezes o produto $n\epsilon$, para $\epsilon := 2.3 * 10^{-16}$.

Os valores no gráfico (a) também indicam que $er_1[\widehat{\mathbf{y}}](x) = O(\epsilon)$, a partir do que poderia-se deduzir, por (5.32), que $er_2[\widehat{\mathbf{y}}](x) \approx er_1[\mathbb{1}](x) = O(n\epsilon\Lambda(\widehat{\mathbf{x}}, \mu(\widehat{\mathbf{x}})))$ e isso corrobora os valores plotados para o algoritmo do Tipo II no mesmo gráfico (o intervalo de referência para a constante de Lebesgue nesse caso é o intervalo de interpolação $[a, b] = [0, n]$.)

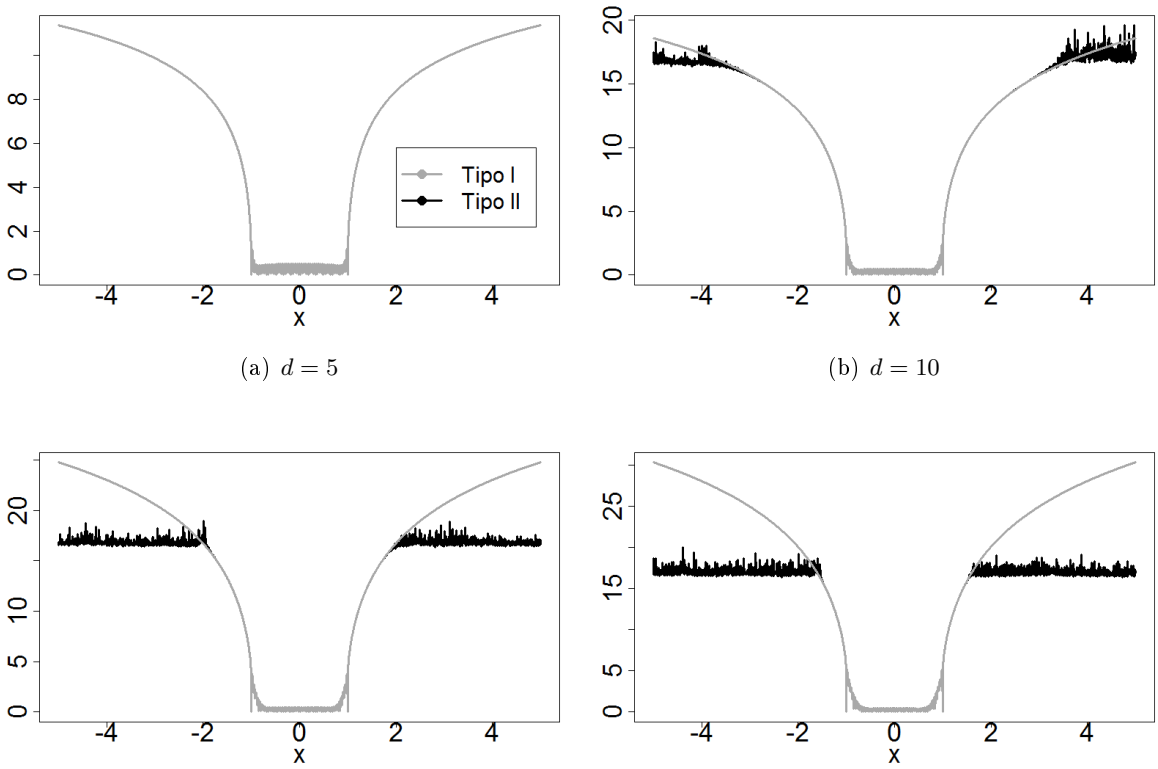
5.4.2 Avaliação estável da função/constante de Lebesgue

As estimativas (3.36) nos dão informações úteis sobre o comportamento da constante de Lebesgue para nós igualmente espaçados, porém obter informações mais precisas sobre a constante de Lebesgue para essa e/ou outras familias de nós é, em geral, difícil e/ou trabalhoso. Na prática, podemos calcular a constante de Lebesgue maximizando a função de Lebesgue (2.6)

$$L(x, \widehat{\mathbf{x}}, \mu(\widehat{\mathbf{x}})) = \begin{cases} \dfrac{\sum_{i=0}^{n} \left|\frac{\mu(\widehat{\mathbf{x}})_i}{x-\widehat{x}_i}\right|}{\left|\sum_{i=0}^{n} \frac{\mu(\widehat{\mathbf{x}})_i}{x-\widehat{x}_i}\right|} = \dfrac{\sum_{i=0}^{n} \left|\frac{\mu(\widehat{\mathbf{x}})_i}{x-\widehat{x}_i}\right|}{\left|\sum_{i=0}^{n-d} \lambda_i(x, \widehat{\mathbf{x}})\right|}, & x \notin \{x_0, x_1, \ldots, x_n\}, \\ 1, & \text{caso contrário}, \end{cases} \tag{5.35}$$

a qual coincide com a definição do número de condição $cond(x, n, \mathbb{1})$ em (5.29). As expressões em (5.35) sugerem que os algoritmos dos tipos I e II podem ser facilmente adaptadados para calcular a função de Lebesgue. Observe que (5.35) nada mais é do que a soma dos valores absolutos dos interpoladores tratados na seção anterior e vimos que os algoritmos do Tipo I são estáveis para calculá-los. Consequentemente, é possível avaliar a função de Lebesgue por meio dos algoritmos do Tipo I com erro relativo pequeno $O(||\mathbf{z}||_\infty + n\epsilon)$, como já afirmado no Corolário 2.

A Figura 5.2 mostra as consequências dessa propriedade na prática.



(a) $d = 5$ (b) $d = 10$

(c) $d = 15$ (d) $d = 20$

Figura 5.2: $\log_{10} L(x, \widehat{\mathbf{x}}, \mu(\widehat{\mathbf{x}}))$ *em* $[-5, 5]$, *para* $d = 5, 10, 15$ *and* 20. $\widehat{\mathbf{x}}$ *corresponde à versão arredondada (em precisão dupla) de* $n + 1 = 52$ *pontos igualmente espaçados em* $[-1, 1]$.

Em cada gráfico, a curva em cinza, obtida pelos valores calculados com o algoritmo do Tipo I, mostra o gráfico de uma função “bem comportada", suave por partes, como deveríamos esperar pela definição (5.35). A curva em preto, obtida pelos valores calculados com o algoritmo do Tipo II, apresenta um comportamento similar dentro de uma certa vizinhança do intervalo de interpolação $[-1, 1]$. Porém, fora dessa vizinhança, a curva apresenta um comportamento típico de ruído. Esse fenômeno pode ser explicado brevemente da seguinte forma: quando x está longe do intervalo de interpolação, o valor exato do denominador de (3.28) (dado pelo lado esquerdo de (3.29)) é muito menor do que cada uma de suas parcelas, porém os erros de arredondamento originados nos cálculos dessas parcelas são proporcionais às suas magnitudes e esses erros geralmente não se cancelam na prática. Além disso, sob o modelo de aritmética de ponto flutuante descrito na Seção 2.3, é fácil mostrar que o numerador de (5.35) pode ser calculado com alta precisão relativa. Portanto, o verdadeiro valor de (5.35) é ofuscado pelos grandes erros de arredonadamento originados no cálculo do lado esquerdo de (3.29) e isso explica porque os valores plotados na Figura 5.2 possuem comportamento de ruído. Um fenômeno similar de instabilidade para a segunda fórmula baricêntrica para o interpolador de Lagrange ($d = n$) foi reportado em [WTG12]. A vantagem dos algoritmos do Tipo I é que o denominador de (5.1) (dado pelo lado direito de (3.32)) é calculado com alta precisão relativa e isso previne instabilidade nesse caso.

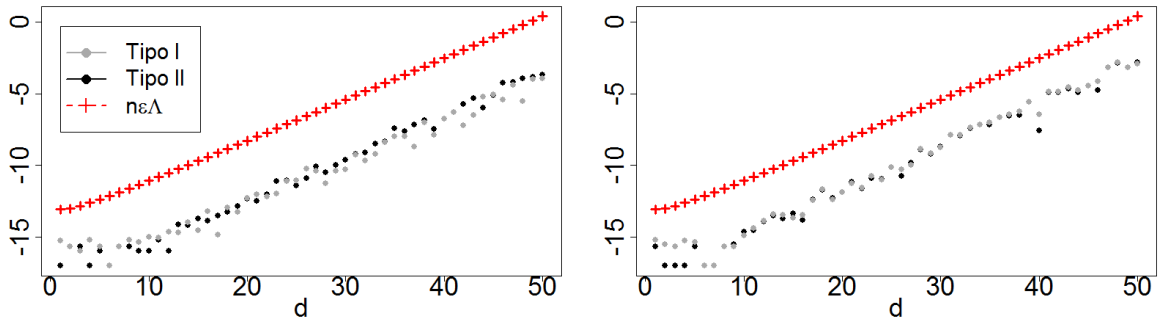
Uma vez constatado que o erro relativo para o algoritmo do Tipo I é $O(n\epsilon)$ nesse caso, poderíamos, também, explicar a instabilidade do algoritmo do Tipo II em termos da equação (5.32), pois, nesse caso, temos $er_2[\widehat{\mathbf{y}}](x) \approx er_1[\mathbb{1}](x)$, e $er_1[\mathbb{1}](x)$ é geralmente grande para extrapolação.

5.4.3 Funções ordinárias

Os exemplos explorados nas seções anteriores são importantes (o exemplo da Seção (5.4.1) ilustra o comportamento do erro backward e o exemplo da Seção 5.4.2 mostra como calcular a função de Lebesgue com alta precisão relativa.) Porém, eles representam apenas uma pequena parcela dos casos possíveis, a saber quando os numeradores de (3.28) e (5.1) podem ser calculados com alta precisão relativa, não importando o valor de d ou a magnitude da constante de Lebesgue. Para interpolação de funções mais comuns, como

$$f(x) = \sin(x) \qquad \text{e} \qquad f(x) = \tfrac{1}{1+x^2} \quad \text{(função de Runge)},$$

por exemplo, tal condição geralmente não se aplica e há perda significativa de precisão nos cálculos desses numeradores conforme a ordem de aproximação (parâmetro d) aumenta. A Figura 5.3 ilustra esse efeito.

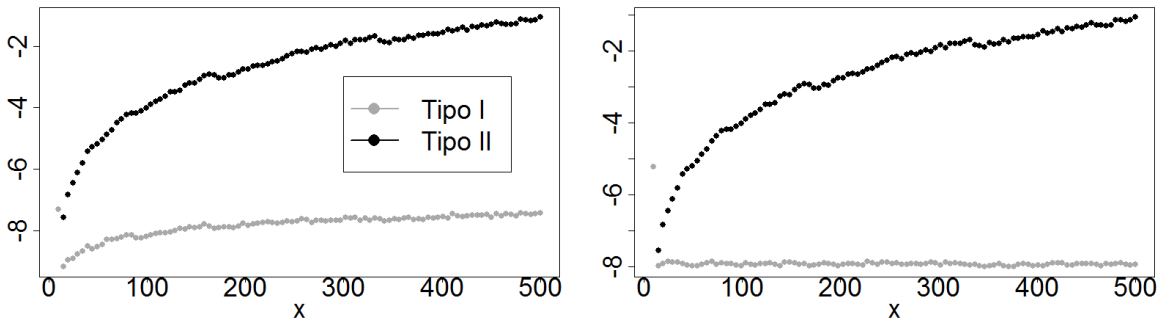


(a) $f(x) = \sin(x)$ (b) $f(x) = \frac{1}{1+x^2}$

Figura 5.3: $\max_{x \in [-5,5]} |fl(u(x, \widehat{\mathbf{x}}, f(\widehat{\mathbf{x}}), \mu(\widehat{\mathbf{x}}))) - u(x, \widehat{\mathbf{x}}, f(\widehat{\mathbf{x}}), \mu(\widehat{\mathbf{x}}))|$, *$u = p$ e $u = q$, com $n + 1 = 201$ pontos igualmente espaçados em $[a = -5, b = 5]$.*

Para cada valor de d, a ordenada correspondente nos gráficos (a) e (b) corresponde ao máximo erro forward (em escala logaritmica, base = 10) de (3.28) e (5.1) para 10000 pontos igualmente espaçados em $[a = -5, b = 5]$. Esses gráficos mostram que as estimativas superiores dadas por (5.25) e (5.26) fornecem uma boa estimativa sobre a magnitude do erro forward para o algoritmo do Tipo I. A similaridade entre as performances dos algoritmos dos Tipos I e II, nesse caso, pode ser explicada em termos de (5.32) e o fato do erro $er_1[\mathbb{1}](x)$ para interpolação da função constante não ser muito maior do que o erro $er_1[\widehat{\mathbf{y}}](x)$.

Agora para extrapolação, na Figura (5.4) vemos o erro relativo forward (também em escala logaritmica) na avaliação numérica das fórmulas (3.28) e (5.1) em $[5, 500]$, para $d = 3$ fixo e para os mesmos 201 nós igualmente espaçados em $[-5, 5]$. Os valores das abscissas nesses gráficos correspondem a 100 pontos igualmente espaçados $5 = t_0, t_1, \ldots, t_{99} = 500$ em $[5, 500]$ e a ordenada correspondente à abscissa t_k $(1 \leq k \leq 99)$ é o máximo valor absoluto do erro relativo forward sobre 1000 pontos igualmente espaçados em $[t_{k-1}, t_k]$.



(a) $f(x) = \sin(x)$ (b) $f(x) = \frac{1}{1+x^2}$

Figura 5.4: $\max\limits_{x \in [t_{k-1}, t_k]} \left| \frac{fl(u(x, \widehat{\mathbf{x}}, f(\widehat{\mathbf{x}}), \mu(\widehat{\mathbf{x}})))}{u(x, \widehat{\mathbf{x}}, f(\widehat{\mathbf{x}}), \mu(\widehat{\mathbf{x}}))} - 1 \right|$, $k = 0, 1, \ldots, 99$ *($u = p$ e $u = q$).*

O erro relativo no cálculo do denominador de (5.1) é $O(n\epsilon)$ (ver (5.18)) e, consequentemente, o erro relativo forward que observamos para o algoritmo do Tipo I nesses gráficos é, essencialmente, o erro relativo no cálculo do numerador de (5.1). Além disso, lembrando que os numeradores de (3.28) e (5.1) são calculados de forma idêntica, a comparação entre os erros dos algoritmos dos Tipos I e II mostra o grande impacto dos erros de arredondamento na avaliação numérica do denominador de (3.28) para extrapolação. Em vista de (5.29), esse fenômeno também mostra que, nese caso, o número de condição $cond(x, n, \mathbb{1})$ é muito maior do que $cond(x, n, \widehat{\mathbf{y}})$ para x longe do intervalo de interpolação.

5.4.4 Discussão

Os exemplos mostrados nas seções anteriores sugerem que os algoritmos do Tipo I são mais estáveis do que os algoritmos do Tipo II para extrapolação. Para interpolação, os algoritmos do Tipo I são significamente mais estáveis do que aqueles do Tipo II somente nos casos nos quais a constante de Lebesgue é alta e o numerador de (5.1) pode ser calculado com alta precisão relativa e os resultados dos experimentos da seção anterior mostraram que isso parece ser incomum na prática. Além disso, [MdC16] mostrou que os algoritmos do Tipo II podem ser efetivamente mais estáveis do que os algoritmos do Tipo I para funções com baixa constante de Lipschitz, no contexto da interpolação de Lagrange ($d = n$). Portanto, a análise da estabilidade forward desses algoritmos pode ser sutil e os resultados apresentados neste capítulo e o estudo [MdC16] fornecem informações relevantes para ajudar na escolha do algoritmo mais apropriado para uma aplicação particular.

Capítulo 6

A estabilidade numérica dos interpoladores de Floater-Hormann estendidos

Os exemplos numéricos da Seção 5.4.3 mostraram que os interpoladores de Floater-Hormann são instáveis para valores grandes do parâmetro d (o qual define a ordem de convergência desses interpoladores, ver Teorema 2) para nós igualmente espaçados e que essa instabilidade é devida, essencialmente, ao crescimento exponencial da constante de Lebesgue em função de d.

Em uma tentativa de aumentar a estabilidade numérica desses interpoladores para nós igualmente espaçados, Klein [Kle13] introduziu uma nova classe de interpoladores, denominada classe dos interpoladores de Floater-Hormann estendidos. Dados nós igualmente espaçados $\mathbf{x} = \mathbf{x_{eq}^n}$, valores $\mathbf{y}$ e $d \geq 0$, o interpolador de Floater-Hormann estendido genérico é definido por

$$\tilde{r}_d(x, \mathbf{x}, \mathbf{y}) = r_d(x, \tilde{\mathbf{x}}, \tilde{\mathbf{y}}), \ x \ \in [x_0, x_n],$$

onde a malha $\tilde{\mathbf{x}} = (x_{-d}, \ldots, x_{-1}, x_0, \ldots, x_n, x_{n+1}, \ldots, x_{n+d})$ é obtida a partir de $\mathbf{x}$ adicionando-se d pontos igualmente espaçados em cada extremo do intervalo $[x_0, x_n]$ e, supondo $\mathbf{y} = f(\mathbf{x})$, para alguma função $f : [x_{-d}, x_{n+d}] \longrightarrow \mathbb{R}$, o vetor $\tilde{\mathbf{y}} \ \in \ \mathbb{R}^{n+2d+1}$ é definido como alguma aproximação de $f(\tilde{\mathbf{x}})$. No único método apresentado em [Kle13] para construir efetivamente essas aproximações, $\tilde{\mathbf{y}} = Y_{\tilde{d},\tilde{n}}(\mathbf{y})$ é uma função linear em relação à $\mathbf{y}$, definida por um certo processo de extrapolação sobre os $2d$ nós adicionados (o qual descreveremos formalmente na próxima seção), determinado por dois parâmetros $\tilde{d}$ e $\tilde{n}$.

Contudo, a análise apresentada em [Kle13] ignora os efeitos de $\tilde{d}$ e $\tilde{n}$ no proceso de aproximação e, consequentemente, superestima o condicionamento (e também a estabilidade) dos interpoladores estendidos. Por exemplo, o Teorema 3.1 de [Kle13] afirma que, "sob certa interpretação", a constante de Lebesgue $\tilde{\Lambda}$ dos interpoladores estendidos satisfaz

$$\tilde{\Lambda} \leq 0.65(2 + \log(n + 2d)), \quad \text{para } d \ \geq 5 \tag{6.1}$$

(observe que não há nenhuma dependência de $\tilde{d}$ ou $\tilde{n}$ no limitante no lado direito da equação acima.) Por outro lado, a norma $\tilde{\Lambda}_{d,\tilde{d},\tilde{n}}(\mathbf{x})$ do operador linear $\mathbf{y} \longmapsto \tilde{r}_d(x, \mathbf{x}, \mathbf{y}) \ = \ r_d\left(x, \tilde{\mathbf{x}}, Y_{\tilde{d},\tilde{n}}(\mathbf{y})\right)$, a qual corresponde à constante de Lebesgue (2.5) para interpolação baricêntrica, está bem definida e a desigualdade (6.1) é falsa, em geral, para $\tilde{\Lambda}_{d,\tilde{d},\tilde{n}}(\mathbf{x})$. Os efeitos de $\tilde{d}$ e $\tilde{n}$ sobre a ordem de aproximação também são subestimados. Por exemplo, a Figura 6 de [Kle13] analisa a estabilidade dos interpoladores estendidos em função de $d \ \in \{1, 2, \ldots, 50\}$, com $\tilde{d} = 7$ e $\tilde{n} = 11$ fixos, porém não é mencionado que a ordem de aproximação desses interpoladores não é sensível ao parâmetro d para $d > \tilde{d}$.

Neste capítulo apresentamos uma nova análise da estabilidade numérica dos interpoladores de Floater-Hormann estendidos, a qual considera propriamente os efeitos de $\tilde{d}$ e $\tilde{n}$ no processo total de aproximação. Nosso resultado principal (Teorema 10) mostra que a constante de Lebesgue $\tilde{\Lambda}_{d,\tilde{d},\tilde{n}}(\mathbf{x})$

pode ter crescimento exponencial em função do parâmetro que define a ordem de aproximação desses interpoladores. Além disso, nossos experimentos numéricos mostram que o processo de extrapolação na construção desses interpoladores é uma fonte significativa de instabilidade numérica. A conclusão é que as vantagens dos interpoladores estendidos sobre os interpoladores de Floater-Hormann são muito mais limitadas do que se afirmava na literatura corrente.

6.1 Definição formal

Conforme mencionado na Seção 3.2.5, a definição dos interpoladores de Floater-Hormann estendidos é motivada pela observação de que, dados nós de interpolação $\mathbf{x}$ igualmente espaçados, a função de Lebesgue (2.6) associada ao interpolador de Floater-Hormann com nós $\mathbf{x}$ e d possui magnitude da ordem de $\log(n)$ no intervalo $[x_d, x_{n-d}]$ para $n > 2d$,. Mais precisamente, vale que (ver Teorema 3.1 em [Kle13])

$$\Lambda(\mathbf{x}, \mu_d(\mathbf{x}))|_{x_d}^{x_{n-d}} \leq 0.65(2 + \log(n)), \quad d \geq 5. \tag{6.2}$$

Dessa forma, a teoria vista no capítulo anterior mostra ambos os algoritmos dos Tipos I e II são estáveis para a valiação numérica dos interpoladores de Floater-Hormann no intervalo $[x_d, x_{n-d}]$.

Seguindo essa linha de raciocínio, a idéia por trás do interpoladores estendidos é clara: se a função f, a ser interpolada em $[x_0, x_n]$, admite uma extensão f^* (com algum grau de suavidade) em $[x_{-d}, x_{n+d}]$ e se os valores de f^* são conhecidos também nos pontos $x_{-d}, ...x_{-1}, x_{n+1}, \ldots, x_{n+d}$, então o interpolador de Floater-Hormann $r_d(., \tilde{\mathbf{x}}, f^*(\tilde{\mathbf{x}}))$ é numericamente estável no intervalo original de interpolação $[x_0, x_n]$. Como, em geral, não conhecemos valores outros do que os valores originais $\mathbf{y} = f(\mathbf{x})$, é proposto que tais valores sejam obtidos por meio da extrapolação (sobre os nós adicionados) de alguns interpoladores locais. Para os interpoladores estendidos apresentados em [Kle13], esse processo de extrapolaçao é definido em função de dois parâmetros $\tilde{n} < n$ e $\tilde{d} \leq \tilde{n}$: para $x_i \in \{x_{-d}, x_{-d+1}, ...x_{-1}\}$, por exemplo, o valor aproximado de $f^*(x_i)$ é obtido pela avaliação (no ponto x_i) do polinômio de Taylor de grau $\tilde{d}$ do interpolador de Floater-Hormann com parâmetro $\tilde{d}$ e nós $x_0, x_1, ...x_{\tilde{n}}$. Mais precisamente, dados nós igualmente espaçados $\mathbf{x} = (x_0, x_1, \ldots, x_n)$ e valores $\mathbf{y} = (y_0, y_1, \ldots, y_n)$, o interpolador de Floater-Hormann estendido com parâmetros $d, \tilde{n}, \tilde{d}$ e κ é definido por

$$\tilde{r}_{d,\tilde{n},\tilde{d},\kappa}(x, \mathbf{x}, \mathbf{y}) = r_d(x, \tilde{\mathbf{x}}, \tilde{\mathbf{y}}), \tag{6.3}$$

com $\tilde{\mathbf{x}}, \tilde{\mathbf{y}} \in \mathbb{R}^{n+1+2\kappa}$ dados por

$$\begin{cases} \tilde{x}_i = x_0 + ih, \quad \tilde{y}_i = \sum_{k=0}^{\tilde{d}} \frac{\partial^k r_d\left(x_0, \mathbf{x^{(0)}}, \mathbf{y^{(0)}}\right)}{\partial x^k} \frac{(x_i - x_0)^k}{k!}, & -\kappa \leq i \leq -1, \\ \tilde{x}_i = x_i, \quad \tilde{y}_i = y_i, & 0 \leq i \leq n, \\ \tilde{x}_i = x_n + (i-n)h, \quad \tilde{y}_i = \sum_{k=0}^{\tilde{d}} \frac{\partial^k r_d\left(x_n, \mathbf{x^{(n)}}, \mathbf{y^{(n)}}\right)}{\partial x^k} \frac{(x_i - x_n)^k}{k!}, & n+1 \leq i \leq n+\kappa, \end{cases} \tag{6.4}$$

com $\mathbf{x^{(0)}} := (x_0, x_1, \ldots, x_{\tilde{n}})$, $\mathbf{y^{(0)}} := (y_0, y_1, \ldots, y_{\tilde{n}})$, $\mathbf{x^{(n)}} := (x_{n-\tilde{n}}, x_{n-\tilde{n}+1}, \ldots, x_n)$ e $\mathbf{y^{(n)}} := (y_{n-\tilde{n}}, y_{n-\tilde{n}+1}, \ldots, y_n)$. Klein originalmente trabalhou com $\kappa = d$. Porém, para fins de análise, convém estender essa definição (e não há nenhum esforço adicional nisso) para o contexto mais geral.

6.2 Críticas

6.2.1 Interpretação incomum da constante de Lebesgue

Segundo Klein, como os valores aproximados $\tilde{\mathbf{y}} \approx f^*(\tilde{\mathbf{x}})$ podem ser obtidos também por algum outro processo de aproximação e como o interpolador $\tilde{r}_{d,\tilde{n},\tilde{d},\kappa}(x,\mathbf{x},\mathbf{y})$ só será avaliado em $[x_0,x_n]$, as incertezas nesse processo de aproximação poderiam, para efeito geral, ser desprezadas. Consequentemente, assumindo que $\tilde{\mathbf{y}} = f(\tilde{\mathbf{x}})$, a quantidade

$$\tilde{\Lambda} \;=\; \Lambda(\tilde{\mathbf{x}},\mu_d(\tilde{\mathbf{x}}))|_{x_0}^{x_n} \;=\; \max_{x\,\in\,[x_0,x_n]} \;\sum_{i=-\kappa}^{n+\kappa}\left|\frac{\mu_d(\tilde{\mathbf{x}})_i}{\tilde{x}-x_i}\right| \Bigg/ \sum_{i=-\kappa}^{n+\kappa}\left|\frac{\mu_d(\tilde{\mathbf{x}})_i}{\tilde{x}-x_i}\right| \tag{6.5}$$

(onde o vetor $\mu_d(\tilde{\mathbf{x}}) \in \mathbb{R}^{n+1+2\kappa}$ é indexado por $-\kappa,\ldots,-1,0,\ldots,n,n+1\ldots n+\kappa$) poderia ser interpretada como a constante de Lebesgue do interpolador estendido (6.3) no sentido usual de amplificador da magnitude do erro devido a perturbações nos valores interpolados, isto é

$$\left|\tilde{r}_{d,\tilde{n},\tilde{d},\kappa}(x,\mathbf{x},\mathbf{y}+\mathbf{\Delta y}) - \tilde{r}_{d,\tilde{n},\tilde{d},\kappa}(x,\mathbf{x},\mathbf{y})\right| \;\le\; \tilde{\Lambda}||\mathbf{\Delta y}||_\infty, \quad \text{para} \quad x \in [x_0,x_n]. \tag{6.6}$$

Porém, como todas as derivadas em (6.4) são funções lineares em $\mathbf{y}$ (ver equação (2.13)), isto é:

$$\frac{\partial^k r_d\left(x_0,\mathbf{x^{(0)}},\mathbf{y^{(0)}}\right)}{\partial x^k} = \sum_{\ell=0}^{\tilde{n}} c_{0,\ell}^{(k)}\; y_\ell, \quad \frac{\partial^k r_d\left(x_n,\mathbf{x^{(n)}},\mathbf{y^{(n)}}\right)}{\partial x^k} = \sum_{\ell=n-\tilde{n}}^{n} c_{n,\ell}^{(k)}\; y_\ell, \quad k=0,1,\ldots,\tilde{d}, \tag{6.7}$$

então a norma (a constante de Lebesgue no sentido usual de (2.5))

$$\tilde{\Lambda}_{d,\tilde{n},\tilde{d},\kappa}(\mathbf{x}) \tag{6.8}$$

do operador linear $\mathbf{y} \longmapsto \tilde{r}_{d,\tilde{n},\tilde{d},\kappa}(x,\mathbf{x},\mathbf{y})$ está bem definida e, supondo (6.6), valeria

$$\tilde{\Lambda}_{d,\tilde{n},\tilde{d},\kappa}(\mathbf{x}) \;\le\; \tilde{\Lambda} \;\overset{(6.2)}{\le}\; 0.65(2+\log(n+2d)), \quad d \ge 5 \quad (\text{para } \kappa = d.) \tag{6.9}$$

Uma primeira objeção ao trabalho de Klein é que a inequação (6.9) é obviamente incorreta para valores grandes de $\tilde{d}$. Por exemplo, se $\tilde{d} = \tilde{n} = d = n$, então $r_d\left(x,\mathbf{x^{(0)}},\mathbf{y^{(0)}}\right) = r_d\left(x,\mathbf{x^{(n)}},\mathbf{y^{(n)}}\right) = r_d\left(x,\mathbf{x},\mathbf{y}\right)$ são polinômios de grau $\tilde{d}$, os quais coincidem com sua expansão de Taylor de ordem d. Logo, o polinômio $\tilde{r}_{d,\tilde{n},\tilde{d},\kappa}(x,\mathbf{x},\mathbf{y}) = r_d(x,\tilde{\mathbf{x}},\tilde{\mathbf{y}})$ interpola o polinômio $r_d\left(x,\mathbf{x},\mathbf{y}\right)$ em $n+1+2\kappa$ pontos distintos e, portanto, devem ser idênticos (lembrando que o interpolador de Floater-Hormann com parâmetro d reproduz identicamente polinômios de grau até d.) Portanto, temos que, nesse caso, o interpolador estendido é idêntico ao interpolador de Floater-Hormann o qual, por sua vez, é idêntico ao interpolador polinomial de Lagrange e, como vimos no Capítulo 3, a constante de Lebesgue para interpolação polinomial em nós igualmente espaçados possui crescimento exponencial com relação ao grau $\tilde{d} = \tilde{n} = d = n$ do polinômio e isso é incompatível com (6.9).

Podemos, também, fazer uma análise empírica da ordem de magnitude da constante de Lebesgue (6.8) com base na seguinte representação dos interpoladores estendidos

$$
\begin{aligned}
\tilde{r}_{d,\tilde{n},\tilde{d},\kappa}(x,\mathbf{x},\mathbf{y}) &= \sum_{i=-\kappa}^{n+\kappa} \frac{\mu_d(\tilde{\mathbf{x}})_i \tilde{y}_i}{x-\tilde{x}_i} \Big/ D(x) \overset{(6.4)}{=} \left[\sum_{i=0}^{n} \frac{\mu_d(\tilde{\mathbf{x}})_i y_i}{x-\tilde{x}_i}\right] \Big/ D(x) \\
&+ \left[\sum_{i=-\kappa}^{-1} \frac{\mu_d(\tilde{\mathbf{x}})_i \left(\sum_{k=0}^{\tilde{d}} \frac{\partial^k r_d\left(x_0,\mathbf{x^{(0)}},\mathbf{y^{(0)}}\right)}{\partial x^k} \frac{(x_i-x_0)^k}{k!}\right)}{x-\tilde{x}_i}\right] \Big/ D(x) \\
&+ \left[\sum_{i=n+1}^{n+\kappa} \frac{\mu_d(\tilde{\mathbf{x}})_i \left(\sum_{k=0}^{\tilde{d}} \frac{\partial^k r_d\left(x_n,\mathbf{x^{(n)}},\mathbf{y^{(n)}}\right)}{\partial x^k} \frac{(x_i-x_n)^k}{k!}\right)}{x-\tilde{x}_i}\right] \Big/ D(x) \\
&= \left[\sum_{i=0}^{n} \frac{\mu_d(\tilde{\mathbf{x}})_i y_i}{x-\tilde{x}_i}\right] \Big/ D(x) \\
&+ \sum_{k=0}^{\tilde{d}} \frac{\partial^k r_d\left(x_0,\mathbf{x^{(0)}},\mathbf{y^{(0)}}\right)}{\partial x^k} \left[\sum_{i=-\kappa}^{-1} \frac{\mu_d(\tilde{\mathbf{x}})_i \frac{(x_i-x_0)^k}{k!}}{x-\tilde{x}_i}\right] \Big/ D(x) \\
&+ \sum_{k=0}^{\tilde{d}} \frac{\partial^k r_d\left(x_n,\mathbf{x^{(n)}},\mathbf{y^{(n)}}\right)}{\partial x^k} \left[\sum_{i=n+1}^{n+\kappa} \frac{\mu_d(\tilde{\mathbf{x}})_i \frac{(x_i-x_n)^k}{k!}}{x-\tilde{x}_i}\right] \Big/ D(x),
\end{aligned}
$$

para $D(x) = \sum_{i=-\kappa}^{n+\kappa} \frac{\mu_d(\tilde{\mathbf{x}})_i}{x-\tilde{x}_i}$, ou seja,

$$
\begin{aligned}
\tilde{r}_{d,\tilde{n},\tilde{d},\kappa}(x,\mathbf{x},\mathbf{y}) &\overset{(6.7)}{=} \left[\sum_{i=0}^{n} \frac{\mu_d(\tilde{\mathbf{x}})_i y_i}{x-\tilde{x}_i}\right] \Big/ D(x) \\
&+ \sum_{k=0}^{\tilde{d}} \sum_{\ell=0}^{\tilde{n}} c_{0,\ell}^{(k)}\, y_\ell \left[\sum_{i=-\kappa}^{-1} \frac{\mu_d(\tilde{\mathbf{x}})_i \frac{(x_i-x_0)^k}{k!}}{x-\tilde{x}_i}\right] \Big/ D(x) \\
&+ \sum_{k=0}^{\tilde{d}} \sum_{\ell=n-\tilde{n}}^{n} c_{n,\ell}^{(k)}\, y_\ell \left[\sum_{i=n+1}^{n+\kappa} \frac{\mu_d(\tilde{\mathbf{x}})_i \frac{(x_i-x_n)^k}{k!}}{x-\tilde{x}_i}\right] \Big/ D(x) \\
&\overset{(**)}{=} \sum_{\ell=0}^{\tilde{n}} \left(\frac{\mu_d(\tilde{\mathbf{x}})_\ell}{x-\tilde{x}_\ell} + \sum_{k=0}^{\tilde{d}} c_{0,\ell}^{(k)} \left[\sum_{i=-\kappa}^{-1} \frac{\mu_d(\tilde{\mathbf{x}})_i \frac{(x_i-x_0)^k}{k!}}{x-\tilde{x}_i}\right] \Big/ D(x)\right) y_\ell \\
&+ \sum_{\ell=\tilde{n}+1}^{n-\tilde{n}-1} \left(\left[\frac{\mu_d(\tilde{\mathbf{x}})_\ell}{x-\tilde{x}_\ell}\right] \Big/ D(x)\right) y_\ell \\
&+ \sum_{\ell=n-\tilde{n}}^{\tilde{n}} \left(\frac{\mu_d(\tilde{\mathbf{x}})_\ell}{x-\tilde{x}_\ell} + \sum_{k=0}^{\tilde{d}} c_{n,\ell}^{(k)} \left[\sum_{i=n+1}^{n+\kappa} \frac{\mu_d(\tilde{\mathbf{x}})_i \frac{(x_i-x_n)^k}{k!}}{x-\tilde{x}_i}\right] \Big/ D(x)\right) y_\ell.
\end{aligned}
$$

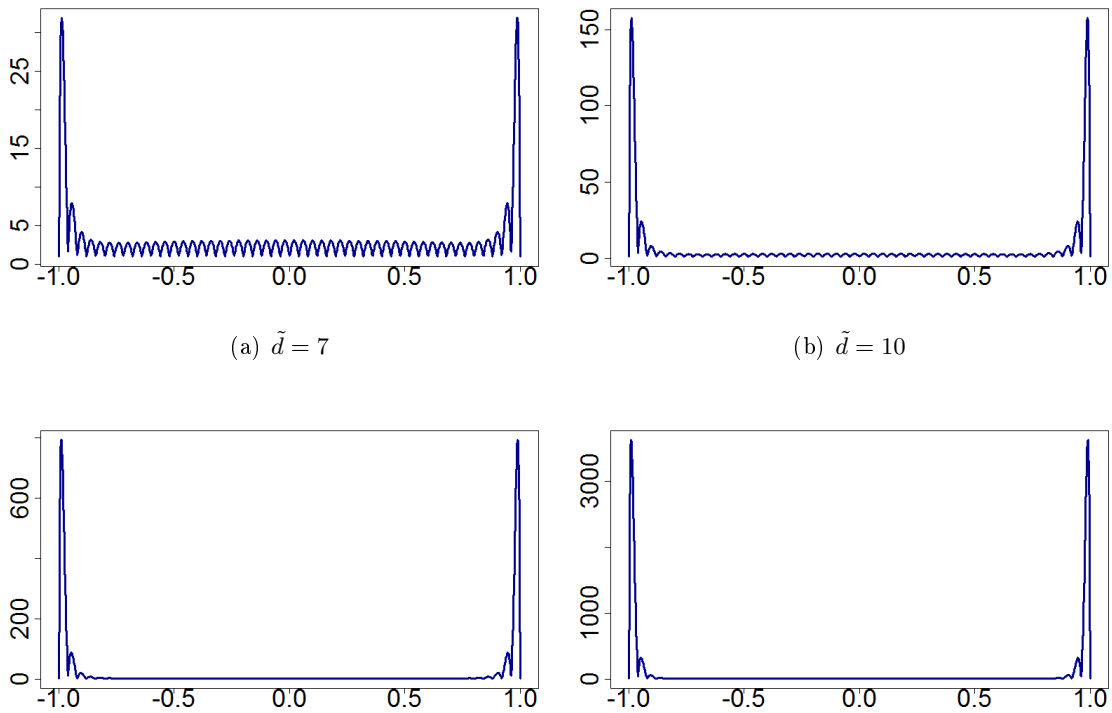
(**) Estamos assumindo que $\tilde{n} < \lfloor n/2 \rfloor$.

Dessa forma, a função de Lebesgue para o interpolador estendido fica $Leb_{d,\tilde{n},\tilde{d},\kappa}(x,\mathbf{x}) =$

$$
\left(\begin{array}{c}
\displaystyle\sum_{\ell=0}^{\tilde{n}} \left|\frac{\mu_d(\tilde{\mathbf{x}})_\ell}{x-\tilde{x}_\ell} + \sum_{k=0}^{\tilde{d}} c_{0,\ell}^{(k)} \left[\sum_{i=-\kappa}^{-1} \frac{\mu_d(\tilde{\mathbf{x}})_i \frac{(x_i-x_0)^k}{k!}}{x-\tilde{x}_i}\right]\right| + \sum_{\ell=\tilde{n}+1}^{n-\tilde{n}-1} \left|\frac{\mu_d(\tilde{\mathbf{x}})_\ell}{x-\tilde{x}_\ell}\right| \\
\displaystyle\sum_{\ell=n-\tilde{n}}^{\tilde{n}} \left|\frac{\mu_d(\tilde{\mathbf{x}})_\ell}{x-\tilde{x}_\ell} + \sum_{k=0}^{\tilde{d}} c_{n,\ell}^{(k)} \left[\sum_{i=n+1}^{n+\kappa} \frac{\mu_d(\tilde{\mathbf{x}})_i \frac{(x_i-x_n)^k}{k!}}{x-\tilde{x}_i}\right]\right|
\end{array}\right) \Big/ |D(x)|. \tag{6.10}
$$

Na Figura 6.1 exibimos os gráficos da função de Lebesgue (6.10) para alguns valores de $\tilde{d}$, com $n = 50, d = \kappa = 18$ e $\tilde{n} = 20$ fixos. Como podemos observar, o máximo da função de Lebesgue em

cada caso é bem distinto do que o suposto por (6.9). Na Seção 6.3 falaremos mais sobre a constante de Lebesgue para os interpoladores de Floater-Hormann estendidos.



(a) $\tilde{d} = 7$ (b) $\tilde{d} = 10$

(c) $\tilde{d} = 13$ (d) $\tilde{d} = 16$

Figura 6.1: *A função de Lebesgue* $Leb_{d,\tilde{n},\tilde{d},\kappa}(x, \mathbf{x})$*, para* $n + 1 = 51$ *nós igualmente espaçados em* $[-1, 1]$*, com* $d = \kappa = 18$ *e* $\tilde{n} = 20$ *fixos.*

6.2.2 Estabilidade versus convergência

Em princípio, a discrepância entre (6.5) e (6.8) para valores grandes de $\tilde{d}$ apontada na seção anterior não chega a ser uma grande contradição no trabalho de Klein, pois, ao final da página 2275 de [Kle13], está explícito que se deve tomar $\tilde{d} \leq \tilde{n} << n$ e, para valores pequenos de $\tilde{d}$, a constante de Lebesgue (6.8) é, de fato, pequena. Por exemplo, não é difícil ver que a relação (6.9) é verdadeira para $\tilde{d} = 0$. Basta combinar a relação $\Lambda(\tilde{\mathbf{x}}, \mu_d(\tilde{\mathbf{x}}))|_{x_0}^{x_n} \overset{(6.2)}{\leq} 0.65(2 + \log(n + 2d)), \quad d \geq 5$, com o fato de que $||\tilde{\mathbf{y}}||_\infty = ||\mathbf{y}||$, nesse caso. Com base na estimativa (6.9) para a constante de Lebesgue dos interpoladores estendidos e nos experimentos numéricos apresentados em [Kle13], os quais são realizados com valores relativamente baixos de $\tilde{n}$ e $\tilde{d}$, Klein conlui que os interpoladores estendidos são mais estáveis do que os interpoladores usuais de Floater-Hormann com relação ao parâmetro d.

Para fins didáticos, reproduzimos abaixo o experimento correspondente à Figura 6 de [Kle13], porém com a função $f(x) = \sin(x)$ e com diferentes parâmetros. A Figura 6.2 mostra o comportamento do erro de interpolação para a função f, adicionando-se uma perturbação de 10^{-10} (com sinais alternados) nos valores interpolados, para $\tilde{d} = \tilde{n} = 3, n = 1000$ fixos e $d = \kappa$ variando de 1 até 50. Os parâmetros utilizados por Klein são $n = 1000$, $\tilde{n} = 11$ e $\kappa = \tilde{d} = 7$, fixos e a perturbação adicionada aos valores interpolados é da ordem de 10^{-12}. Optamos por utilizar a função seno no nosso experimento pois o erro Tipo I (ver Figura 4.1) para o interpolador de Floater-Hormann para a função de Runge (3.11) (utilizada no experimento de [Kle13]) pode não ser desprezível para valores grandes de d, ou seja, o erro total de interpolação para a função de Runge pode não refletir somente as características da estabilidade numérica do método. Observe, por exemplo, que no caso do interpolador de Lagrange ($d = n$), a convergência não é obtida para função de Runge no caso

de nós igualmente espaçados, conforme já mencionado na Seção 3.1.3. Por outro lado, o limitante superior (3.35) para o erro de interpolação é decrescente em função de d para a função seno, pois suas derivadas são todas limitadas por uma mesma constante.

Os resulados apresentados na Figura 6.2 e na Figura 6 em [Kle13] são similares e mostram que, para valores fixos de $\tilde{n}$ e $\kappa = d$, os interpoladores estendidos (EFH) são, de fato, mais estáveis do que os interpoladores de Floater-Hormann (FH) com relação ao parâmetro d.

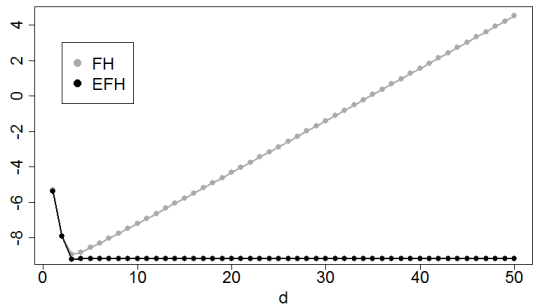


Figura 6.2: *O erro logaritmico* $\log_{10}\left(\max_{x\,\in\,[-5,5]}|I(x)-f(x)|\right)$ *para* $I = FH$ *e* $I = EFH$ *para* $n+1 = 10^3+1$ *nós igualmente espaçados em* $[-5,5]$ *(*$\tilde{d} = \tilde{n} = 3$*,* $\kappa = d$*), com perturbação de* 10^{-10} *nos valores interpolados (com sinais alternados).*

O problema relevante da análise de Klein é que a convergência e a estabilidade dos interpoladores estendidos são analisados separadamente. Klein mostra que, para funções suaves, o erro de interpolação dos interpoladores estendidos (6.3) é $O(h^D)$, com

$$D = \min\{d, \tilde{d}\}.$$

Porém, essa informação parece ser ignorada no restante do artigo e o resultado é uma análise cujas conclusões não são realistas. Observe na Figura 6.3, por exemplo, o comportamento do erro de interpolação para a mesma função quando a perturbação de 10^{-10} nos valores interpolados é ausente: o parâmetro $d > \tilde{d}$ parece não ter efeito algum sobre a ordem de aproximação dos interpoladores estendidos e isso mostra que o parâmetro que define a ordem de convergência desses interpoladores é, de fato, $D = \min\{d, \tilde{d}\}$.

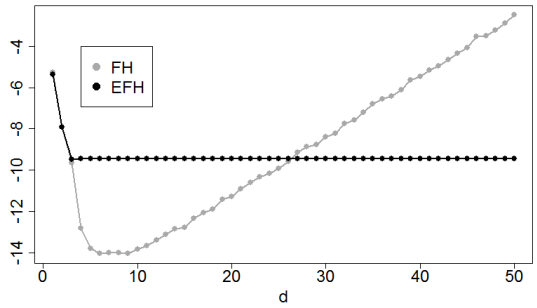


Figura 6.3: *O erro logaritmico* $\log_{10}\left(\max_{x\,\in\,[-5,5]}|I(x)-f(x)|\right)$ *para* $I = FH$ *e* $I = EFH$ *para* $n+1 = 10^3+1$ *nós igualmente espaçados em* $[-5,5]$ *(*$\tilde{d} = \tilde{n} = 3$*,* $\kappa = d$*).*

Isso nos leva, inevitavelmente, às seguintes conlusões:

- Os interpoladores estendidos com parâmetros $d, \tilde{n}, \tilde{d}$ e κ só podem ser comparados aos interpoladores de Floater-Hormann com parâmetro d quando $\tilde{d} \geq d$.

- A comparação de estabilidade feita na Figura 6.2 (ou na Figura 6 de [Kle13]) não faz sentido, pois os interpoladores comparados possuem ordens de convergência distintas.
- A análise de Klein [Kle13] **NÃO** implica que os interpoladores estendidos são mais estáveis do que os interpoladores de Floater-Hormann.

Na próxima seção apresentamos uma análise mais apurada sobre a estabilidade numérica dos interpoladores de Floater-Hormann estendidos quando $\tilde{d} \geq d$. Em particular, mostramos que a constante de Lebesgue associada aos interpoladores estendidos (6.3) cresce exponencialmente em relação ao parâmetro d, quando $\tilde{n} = \tilde{d} = d$. Além disso, apresentamos experimentos numéricos os quais mostram que o processo de extrapolação utilizado na construção dos interpoladores estendidos é uma fonte substancial de instabilidade numérica.

6.3 Estabilidade numérica

6.3.1 O caso minimal $\tilde{d} = \tilde{n} = d$

A nossa análise sobre a estabilidade dos interpoladores estendidos baseia-se predominantemente no estudo do caso minimal

$$\tilde{d} = \tilde{n} = d, \tag{6.11}$$

para o qual temos a mesma ordem de aproximação para os interpoladores estendidos e de Floater-Hormann com parâmetro d, conforme discutido na Seção 6.2.2.

A Figura 6.4 mostra o erro de interpolação com a mesma configuração do experimento relativo à Figura 6.3, porém agora sob (6.11).

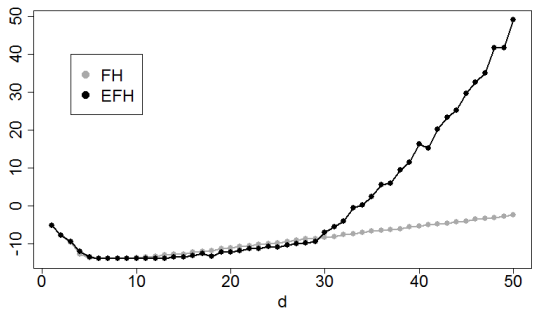


Figura 6.4: *O erro logaritmico* $\log_{10}\left(\max_{x\,\in\,[-5,5]} |I(x) - f(x)|\right)$ *para* $I = FH$ *e* $I = EFH$ *para* $n+1 = 10^3+1$ *nós igualmente espaçados em* $[-5, 5]$ $(\kappa = \tilde{d} = \tilde{n} = d)$.

Como podemos observar, não só os interpoladores estendidos (EFH) não apresentam nenhuma melhora expressiva sobre os interpoladores de Floater-Hormann (FH), como se mostram até mais instáveis para valores grandes de d. O interpolador estentido foi calculado utilizando-se as matrizes definidas em (2.12) para o cálculo das derivadas em (6.4), como sugerido em [Kle13]. Recordamos que, para esse experimento, o erro de interpolação abstrata cometido no Passo 1 da Figura 4.1 do Capítulo 4 é desprezível em relação aos erros de arredondamento para $d \geq 10$ e, portanto, o que vemos na Figura 6.4 é essencialmente o erro cometido no processo de avaliação numérica desses interpoladores.

A Figura 6.5 mostra que esse erro possui três componentes principais:

1. A primeira componente é devida ao arredondamento dos valores interpolados. Para ilustrar o feito causado por esses arredondamentos, calculamos os nós $\hat{\mathbf{x}}$ e valores interpolados $\hat{\mathbf{y}}$ com precisão dupla e avaliamos o interpolador resultante $\tilde{r}_{d,\tilde{n},\tilde{d},\kappa}(x, \hat{\mathbf{x}}, \hat{\mathbf{y}})$ com aritmética de precisão mais alta (50 casas decimais) com o auxílo da biblioteca MPFR (EFH*).

2. A segunda componente é devida ao truncamento do valores extrapolados para precisão dupla. Para ilustrar esse efeito, avaliamos o interpolador estendido $\tilde{r}_{d,\tilde{n},\tilde{d},\kappa}(x, \hat{\mathbf{x}}, \hat{\mathbf{y}})$ em precisão dupla, porém calculando-se os valores extrapolados gerados a partir de $\hat{\mathbf{y}}$ com precisão mais alta (50 casas decimais) e depois arredondado-os para precisão dupla (EFH**).

3. A terceira componente é devida aos erros de arredondamento uriundos da avaliação numérica dos valores extrapolados já truncados. Esse efeito pode ser visualizado comparando-se (EFH**) com (EFH).

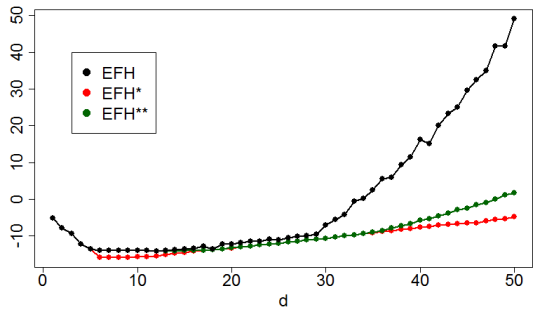


Figura 6.5: *O erro logaritmico* $\log_{10}\left(\max_{x\,\in\,[-5,5]} |I(x) - f(x)|\right)$ *para* $I = FH$ *e* $I = EFH$ $(\kappa = \tilde{d} = \tilde{n} = d)$ *e* $n = 10^3$.

A comparação entre (EFH**) e (EFH) mostra que o processo de extrapolação sugerido em [Kle13] para construir os interpoladores estendidos (o qual utiliza as matrizes (2.12) para calcular as derivadas em (6.4)) é uma fonte significativa de instabilidade numérica. Os valores com a legenda (EFH*) na Figura 6.5 mostram, ainda, que, mesmo caso fossemos capazes de calcular os valores extrapolados com boa precisão, os interpoladores estendidos, ainda assim, seriam instáveis na prática para valores grandes de d, pois, em geral, os valores interpolados possuem erros de arredondamento. Isso, por sua vez, remonta à tradicional análise de perturbação com relação aos valores interpolados, isto é: a análise da constante de Lebesgue.

Análise da constante de Lebesgue

Vimos, na Figura 6.1 da Seção 6.2.1, que a constante de Lebesgue para os interpoladores estendidos não possui crescimento logaritmico com relação ao parâmetro $\tilde{d}$ como fora suposto por Klein. Mais ainda, os valores (EFH*) na Figura 6.5 sugerem que a constante de Lebesgue $\tilde{\Lambda}_{d,\tilde{n},\tilde{d},\kappa}(\mathbf{x})$, para o caso minimal (6.11), possui crescimento exponencial com relação ao parâmetro $d = \tilde{d} = \tilde{n}$ que define a ordem de convergência para os interpoladores estendidos. A seguir provamos efetivamente que essa constante de Lebesgue cresce exponencialmente com relação ao parâmetro d.

Recordemos que $\Lambda\left(\mathbf{x_{eq}^d}, \mu_d\left(\mathbf{x_{eq}^d}\right)\right)\Big|_{x_0}^{x_n} = \Lambda(\mathbf{x_{eq}^d}, \gamma(\mathbf{x_{eq}^d}))\Big|_{x_0}^{x_n}$ denota a constante de Lebesgue para o interpolador de Lagrange com $d+1$ nós igualmente espaçados. Vimos, em (3.21), que essa constante de Lebesgue possui crescimento exponencial com relação à d, ou seja

$$\frac{2^{d-2}}{d^2} < \Lambda(\mathbf{x_{eq}^d}, \gamma(\mathbf{x_{eq}^d}))\Big|_{x_0}^{x_n} < \frac{2^{d+3}}{d}. \tag{6.12}$$

Teorema 10. *Se* $n > d + 2$ *e* $d \geq 2$*, então a constante de Lebesgue para o interpolador estendido com parâmetros* $d = \tilde{n} = \tilde{d}$ *e* $\kappa \geq 1$ *satisfaz*

$$\tilde{\Lambda}_{d,\tilde{n},\tilde{d},\kappa}(\mathbf{x})\Big|_{x_0}^{x_n} \geq \left(1 - \tfrac{2}{d-1} - \tfrac{2}{d(d-1)}\right) \Lambda(\mathbf{x_{eq}^d}, \gamma(\mathbf{x_{eq}^d}))\Big|_{x_0}^{x_n}. \tag{6.13}$$

Demonstração. De acordo com (3.25) e (6.3), dados vetores $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{n+1}$ temos

$$\tilde{r}_{d,d,d,\kappa}(x, \mathbf{x}, \mathbf{y}) = r_d(x, \tilde{\mathbf{x}}, \tilde{\mathbf{y}}) = \frac{\sum_{i=-\kappa}^{n+\kappa-d} \tilde{\lambda}_i(x,\tilde{\mathbf{x}})\tilde{p}_i(x,\tilde{\mathbf{x}},\tilde{\mathbf{y}})}{\sum_{i=-\kappa}^{n+\kappa-d} \tilde{\lambda}_i(x,\tilde{\mathbf{x}})},$$

onde

$$\tilde{\lambda}_i(x, \tilde{\mathbf{x}}) = \frac{(-1)^i}{(x - \tilde{x}_i) \dots (x - \tilde{x}_{i+d})}, \quad \text{para} \quad -\kappa \leq i \leq n + \kappa - d \tag{6.14}$$

e $\tilde{p}_i(x, \tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ é o polinômio de grau menor ou igual a d que interpola os dados $(\tilde{y}_i, \tilde{y}_{i+1}, \dots, \tilde{y}_{i+d})$ em $(\tilde{x}_i, \tilde{x}_{i+1}, \dots, \tilde{x}_{i+d})$. Observe que (6.14) não é a extensão da fórmula (3.26) para vetores com índices negativos, porém essas quantidades diferem apenas em um fator $(-1)^\kappa$ e o interpolador de Floater-Hormann (3.25) é invariante sob dilatações das funções (3.26) por um mesmo fator.

Lembrando que, sob (6.11), os interpoladores $r_d\left(x, \mathbf{x^{(0)}}, \mathbf{y^{(0)}}\right)$ e $r_d\left(x, \mathbf{x^{(n)}}, \mathbf{y^{(n)}}\right)$ em (6.4) são polinomiais (ver comentário após (3.26)), a definição (6.4), o fato das séries de Taylor serem exatas para polinômios e a unicidade do polinômio interpolador de Lagrange mostram que

$$\tilde{p}_i(x, \tilde{\mathbf{x}}, \tilde{\mathbf{y}}) = \begin{cases} \tilde{p}_0(x, \tilde{\mathbf{x}}, \tilde{\mathbf{y}}), & \text{para} \quad -\kappa \leq i \leq 0. \\ \tilde{p}_{n-d}(x, \tilde{\mathbf{x}}, \tilde{\mathbf{y}}), & \text{para} \quad n - d \leq i \leq n - d + \kappa. \end{cases}$$

Assim, para obtemos $\tilde{r}_{d,d,d,\kappa}(x, \mathbf{x}, \mathbf{y}) = \dfrac{\dfrac{\left(\sum_{i=-\kappa}^{0} \tilde{\lambda}_i(x,\tilde{\mathbf{x}})\right)}{\tilde{\lambda}_0(x,\mathbf{x})} \tilde{\lambda}_0(x,\mathbf{x})\tilde{p}_0(x,\tilde{\mathbf{x}},\tilde{\mathbf{y}})}{\sum_{i=-\kappa}^{n+\kappa-d} \tilde{\lambda}_i(x,\tilde{\mathbf{x}})} +$

$$\frac{\sum_{i=1}^{n-d-1} \tilde{\lambda}_i(x, \mathbf{x})\tilde{p}_i(x, \tilde{\mathbf{x}}, \tilde{\mathbf{y}})}{\sum_{i=-\kappa}^{n+\kappa-d} \tilde{\lambda}_i(x, \tilde{\mathbf{x}})} + \frac{\dfrac{\left(\sum_{i=n-d}^{n-d+\kappa} \tilde{\lambda}_i(x,\tilde{\mathbf{x}})\right)}{\tilde{\lambda}_{n-d}(x,\mathbf{x})} \tilde{\lambda}_{n-d}(x, \mathbf{x})\tilde{p}_{n-d}(x, \tilde{\mathbf{x}}, \tilde{\mathbf{y}})}{\sum_{i=-\kappa}^{n+\kappa-d} \tilde{\lambda}_i(x, \tilde{\mathbf{x}})}. \tag{6.15}$$

Observe que, como $n > d + 2$, as 3 somas em (6.15) não possuem termos em comum.

Seja $x \in]x_0, x_1[$ fixado. Afirmamos que $y_0^*, y_1^*, \dots, y_n^* \in \{-1, 1\}$ definidos por

$$\begin{cases} y_0^* &= y_1^* = (-1)^d \\ y_j^* &= (-1)^{d+j-1}, \quad 2 \leq j \leq n \end{cases}$$

satisfazem

$$\tilde{\lambda}_i(x, \mathbf{x})\tilde{p}_i(x, \tilde{\mathbf{x}}, \mathbf{y}^*) = \sum_{j=0}^{d} \frac{|\gamma_j|}{|x - x_{i+j}|}, \text{ para } i \in \{0\} \cup \{2, 3, \dots, n + \kappa - d\}, \tag{6.16}$$

onde $\gamma_j, j = 0, 1, \dots, d$, são dados por (3.10). De fato, a primeira fórmula baricêntrica (3.5) para o interpolador de Lagrange mostra que

$$\tilde{\lambda}_i(x, \mathbf{x})\tilde{p}_i(x, \tilde{\mathbf{x}}, \mathbf{y}^*) = (-1)^i \sum_{j=0}^{d} \frac{\gamma_j y_{i+j}^*}{x - x_{i+j}}.$$

A equação (6.16) segue de

$$\frac{\gamma_0 y_0^*}{x - x_0} = \frac{|\gamma_0|}{|x - x_0|}, \quad \frac{\gamma_1 y_1^*}{x - x_1} = \frac{|\gamma_1|}{|x - x_1|}$$

e

$$\frac{\gamma_j y^*_{i+j}}{x-x_{i+j}} = \frac{(-1)^{d-j}|\gamma_j| y^*_{i+j}}{x-x_{i+j}} = \frac{(-1)^{i-1}|\gamma_j|}{x-x_{i+j}} = \frac{(-1)^{i}|\gamma_j|}{|x-x_{i+j}|}, \quad 2 \le i+j \le n+\kappa-d \ .$$

Observamos, ainda, que

$$\frac{\left(\sum\limits_{i=-\tau}^{0} \tilde{\lambda}_i(x,\tilde{\mathbf{x}})\right)}{\tilde{\lambda}_0(x,\mathbf{x})} > 0 \quad \text{e} \quad \frac{\left(\sum\limits_{i=n-d}^{n-d+\kappa} \tilde{\lambda}_i(x,\tilde{\mathbf{x}})\right)}{\tilde{\lambda}_{n-d}(x,\mathbf{x})} > 0. \tag{6.17}$$

A equação (6.17) é verdadeira, pois $\tilde{\lambda}_{-\kappa}(x,\tilde{\mathbf{x}}), \tilde{\lambda}_{-\kappa+1}(x,\tilde{\mathbf{x}}), \ldots, \tilde{\lambda}_0(x,\tilde{\mathbf{x}})$ possuem todos o mesmo sinal e $\tilde{\lambda}_{n-d}(x,\tilde{\mathbf{x}}), \tilde{\lambda}_{n-d+1}(x,\tilde{\mathbf{x}}), \ldots, \tilde{\lambda}_{n-d+\kappa}(x,\tilde{\mathbf{x}})$ possuem sinais alternados e decrescem em magnitude. Combinando esse fato com (6.15) e (6.16), obtemos

$$|\tilde{r}_{d,d,d,\kappa}(x,\mathbf{x},\mathbf{y}^*)| \quad \ge \quad \frac{\frac{\left(\sum\limits_{i=-\kappa}^{0} \tilde{\lambda}_i(x,\tilde{\mathbf{x}})\right)}{\tilde{\lambda}_0(x,\mathbf{x})} \tilde{\lambda}_0(x,\mathbf{x})\tilde{p}_0(x,\tilde{\mathbf{x}},\mathbf{y}^*) - |\tilde{\lambda}_1(x,\mathbf{x})\tilde{p}_1(x,\tilde{\mathbf{x}},\mathbf{y}^*)|}{\left|\sum\limits_{i=-\kappa}^{n+\kappa-d} \tilde{\lambda}_i(x,\tilde{\mathbf{x}})\right|}.$$

Além disso, segue de (6.16) que $\tilde{\lambda}_0(x,\mathbf{x})\tilde{p}_0(x,\tilde{\mathbf{x}},\mathbf{y}^*) - |\tilde{\lambda}_1(x,\mathbf{x})\tilde{p}_1(x,\tilde{\mathbf{x}},\mathbf{y}^*)| \quad \ge$

$$\frac{|\gamma_0|}{|x-x_0|} + \left[\frac{|\gamma_1|-|\gamma_0|}{|x-x_1|} - \frac{|\gamma_1|}{|x-x_2|}\right] + \sum_{j=2}^{d}\left(\frac{|\gamma_j|}{|x-x_j|} - \frac{|\gamma_j|}{|x-x_{j+1}|}\right). \tag{6.18}$$

A última somatória em (6.18) é positiva para todo $x \in]x_0,x_1[$. A expressão entre colchetes também é não-negativa para $x \in]x_0,x_1[$, pois

$$\frac{|\gamma_1|-|\gamma_0|}{|\gamma_1|} \overset{(3.10)}{=} 1-\frac{1}{d} \ \ge \ \frac{1}{2} \ \ge \ \left|1+\frac{x_2-x_1}{x-x_2}\right| \ = \ \frac{|x-x_1|}{|x-x_2|}.$$

Logo, como $\tilde{\lambda}_{-\kappa}(x,\tilde{\mathbf{x}}), \tilde{\lambda}_{-\kappa+1}(x,\tilde{\mathbf{x}}), \ldots \tilde{\lambda}_0(x,\tilde{\mathbf{x}}), \tilde{\lambda}_1(x,\tilde{\mathbf{x}})$ possuem todos o mesmo sinal e $\tilde{\lambda}_1(x,\tilde{\mathbf{x}})$, $\tilde{\lambda}_2(x,\tilde{\mathbf{x}}), \ldots \tilde{\lambda}_{n+\kappa-d}(x,\tilde{\mathbf{x}})$ é uma sequência alternada e decrescente, obtemos

$$|\tilde{r}_{d,d,d,\kappa}(x,\mathbf{x},\mathbf{y}^*)| \quad \ge \quad \frac{\left|\sum\limits_{i=-\kappa}^{-1} \tilde{\lambda}_i(x,\tilde{\mathbf{x}})\right| |\tilde{p}_0(x,\tilde{\mathbf{x}},\mathbf{y}^*)|}{\left|\sum\limits_{i=-d}^{1} \tilde{\lambda}_i(x,\tilde{\mathbf{x}})\right|}. \tag{6.19}$$

Além disso, vale que

$$\frac{|\tilde{\lambda}_0(x,\tilde{\mathbf{x}})|}{|\tilde{\lambda}_{-1}(x,\tilde{\mathbf{x}})|} = \frac{|x-x_{-1}|}{|x-x_d|} \le \frac{2h}{(d-1)h}$$

e

$$\frac{|\tilde{\lambda}_1(x,\tilde{\mathbf{x}})|}{|\tilde{\lambda}_0(x,\tilde{\mathbf{x}})|} = \frac{|x-x_0|}{|x-x_{d+1}|} \le \frac{h}{dh}$$

e isso implica que $\left|\sum\limits_{i=-\kappa}^{-1} \tilde{\lambda}_i(x,\tilde{\mathbf{x}})\right| \ge \left|\sum\limits_{i=-\kappa}^{1} \tilde{\lambda}_i(x,\tilde{\mathbf{x}})\right| - |\tilde{\lambda}_0(x,\tilde{\mathbf{x}})| - |\tilde{\lambda}_1(x,\tilde{\mathbf{x}})| \ge$

$$\left|\sum_{i=-\kappa}^{1} \tilde{\lambda}_i(x,\tilde{\mathbf{x}})\right| \left(1 - \frac{2}{d-1} - \frac{2}{d(d-1)}\right).$$

Portanto, segue que

$$|\tilde{r}_{d,d,d,\kappa}(x,\mathbf{x},\mathbf{y}^*)| \;\geq\; \left(1-\tfrac{2}{d-1}-\tfrac{2}{d(d-1)}\right)|\tilde{p}_0(x,\tilde{\mathbf{x}},\mathbf{y}^*)|.$$

Por fim, (6.16) mostra que, para $x \in [x_0,x_1]$, $|\tilde{p}_0(.,\tilde{\mathbf{x}},\mathbf{y}^*)|$ é idêntico à função de Lebesgue para interpolação de Lagrange sobre $(d+1)$ nós igualmente espaçados em $[x_0,x_d]$ e é um fato bem conhecido que a função de Lebesgue para interpolação polinomial em nós igualmente espaçados assume seu máximo no primeiro subintervalo da partição (ver [Bru97], por exemplo.) Logo, podemos escolher $x^* \in]x_0,x_1[$ tal que $\tilde{\Lambda}_{d,\tilde{n},\tilde{d},\kappa}(\mathbf{x})\Big|_{x_0}^{x_n} \geq |\tilde{r}_{d,d,d,\kappa}(x^*,\mathbf{x},\mathbf{y}^*)| \geq$

$$\left(1-\tfrac{2}{d-1}-\tfrac{2}{d(d-1)}\right)|\tilde{p}_0(x^*,\mathbf{x},\mathbf{y}^*)| = \left(1-\tfrac{2}{d-1}-\tfrac{2}{d(d-1)}\right)\Lambda(\mathbf{x_{eq}^d},\gamma(\mathbf{x_{eq}^d}))\Big|_{x_0}^{x_n}.$$

□

Note que o Teorema 10 e a desigualdade (6.12) implicam que a taxa de crescimento de $\tilde{\Lambda}_{d,d,d,\kappa}(\mathbf{x})\Big|_{x_0}^{x_n}$ é, pelo menos,

$$\frac{2^{d-2}}{d^2}.$$

Dessa forma, temos que, no caso minimal (6.11), a constante de Lebesgue para os interpoladores estendidos possui crescimento exponencial com relação ao parâmetro d que define a ordem de aproximação desses interpoladores. Combinando-se essa informação com (3.36), podemos obter um limitante real para a pequena melhora da constante de Lebesgue para os interpoladores estendidos sobre os interpoladores de Floater-Hormann

$$\frac{\tilde{\Lambda}_{n,d,\tilde{n},\tilde{d},\kappa}(\mathbf{x})\big|_{x_0}^{x_n}}{\Lambda(\mathbf{x},\mu_d(\mathbf{x}))\big|_{x_0}^{x_n}} \;\geq\; \frac{\frac{2^{d-2}}{d^2}}{2^{d-1}(2+\log(n))} \geq \frac{1}{2d^2(2+\log(n))}$$

e essa estimativa contrasta drasticamente com àquela sugerida por (3.36) e (6.9):

$$\frac{\tilde{\Lambda}_{d,\tilde{n},\tilde{d},\kappa}(\mathbf{x})\big|_{x_0}^{x_n}}{\Lambda(\mathbf{x},\mu_d(\mathbf{x}))\big|_{x_0}^{x_n}} \;\leq\; \frac{0.65(2+\log(n+2d))}{\frac{1}{2^{d+2}}\begin{pmatrix}2d+1\\ d\end{pmatrix}\log\left(\frac{n}{d}-1\right)}.$$

A Figura 6.6 mostra o crescimento da contante de Lebesgue para os interpoladores de Floater-Hormann (FH) e Floater-Hormann extendidos (EFH) na prática, para $1 \leq d = \tilde{d} = \tilde{n} = \kappa \leq 50$ e $n = 1000$. Os valores (EFH*), corretos, foram obtidos calculando-se a função de Lebesgue (6.10) com precisão de 50 casas decimais. Os valores (EFH), visivelmente sem um padrão definido, foram obtidos calculando-se a fórmula (6.10) com precisão dupla. Podemos inferir, daí, que a fórmula (6.10) para a função de Lebesgue dos interpoladores estendidos é numericamente instável. Os valores (LI) correspondem à constante de Lebesgue para o interpolador de Lagrange com $d+1$ nós igualmente espaçados e corroboram a nossa estimativa inferior dada por (6.13).

Portanto, a análise do caso (6.11) mostra que os interpoladores estendidos podem não ser mais estáveis do que os interpoladores de Floater-Hormann com mesma ordem de aproximação. Além disso, os valores (EFH) e (EFH**) da Figura (6.5) mostram que os interpoladores estendidos são muito sensíveis aos erros de arredondamento cometidos no processo de extrapolação definido por (6.4). Enfatizamos, ainda, que a desigualdade (6.13) para a constante de Lebesgue independe do número (κ) de pontos adicionados em cada extremo do intervalo e isso sugere que métodos baseados em extrapolação não devem ser eficazes para melhorar o condicionamento dos interpoladores de Floater-Hormann para interpolação em nós igualmente espaçados. A Figura 6.7 mostra a sensibilidade da constante de Lebesgue com relação ao parâmetro κ e sugere que o valor $\kappa = d$ é quase ótimo.

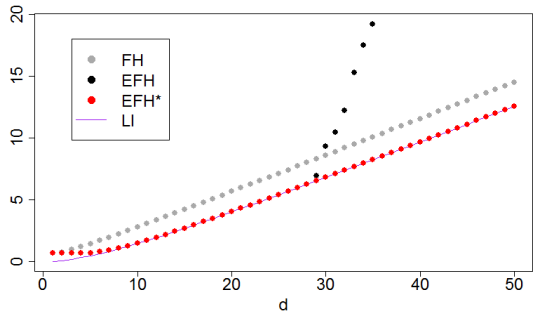


Figura 6.6: *Constantes de Lebesgue para os interpoladores de Floater-Hormann (FH), Floater-Hormann estendidos (EFH*) e para o interpolador de Lagrange (LI) com $d+1$ nós igualmente espaçados ($\kappa = \tilde{d} = \tilde{n} = d$) e $n = 10^3$.*

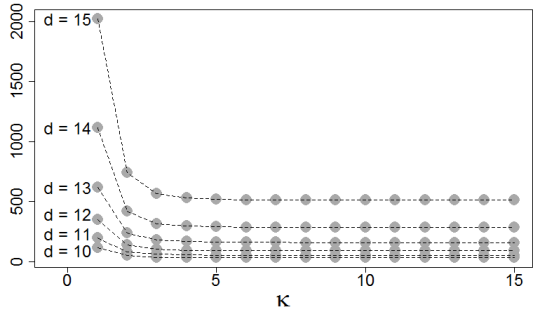


Figura 6.7: *Constantes de Lebesgue para o interpolador de Floater–Hormann extendido em função do parametro κ, com $n = 2000$ e $\tilde{d} = \tilde{n} = d$.*

6.3.2 Caso geral

No caso geral, a análise do crescimento da constante de Lebesgue para os interpoladores de Floater-Hormann estendidos não é tão simples. Porém, a Figura 6.8 sugere que, dentro do caso $\tilde{d} = d$ (para o qual a ordem de convergência é d), o valor mínimo para a constante de Lebesgue é obtido no caso minimal analisado na seção anterior, isto é: quando $\tilde{n} = \tilde{d}$.

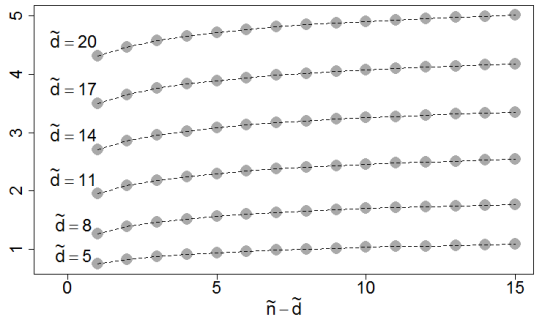


Figura 6.8: $\log_{10}\left(\tilde{\Lambda}_{n,d,\tilde{n},\tilde{d},d}(\mathbf{x})\right)$ *em função da diferença $\tilde{n} - \tilde{d}$, com $n = 100$.*

6.3.3 Uma proposta para melhorar a estabilidade numérica dos interpoladores de Floater-Horamnn estendidos

Um outro questionamento que poderíamos fazer acerca da definição dos interpoladores de Floater-Hormann estendidos em (6.3) é sobre a necessidade da utilização de polinômios de Taylor no

processo de extrapolação. Ora, é bem divulgado nos livros de Cálculo e Análise que a principal função (pelo menos a mais explorada) dos polinômios de Taylor é a de aproximação de funções suaves. Dessa forma, não há como deixar de questionar qual seria a vantagem, no processo de aproximação, de se utilizar polinômios de Taylor (de obtenção computacional relativamente complexa, devido ao cálculo de derivadas) das funções racionais $r_d\left(x, \mathbf{x^{(0)}}, \mathbf{y^{(0)}}\right)$ e $r_d\left(x, \mathbf{x^{(n)}}, \mathbf{y^{(n)}}\right)$ cujas expressões analíticas são previamente conhecidas.

Sob (6.11), as funções racionais utilizadas para gerar os valores extrapolados (6.4) para a construção dos interpoladores estendidos são funções polinômiais e, portanto, coincidem com o seus polinômios de Taylor de grau d, isto é:

$$\begin{cases} r_d\left(x, \mathbf{x^{(0)}}, \mathbf{y^{(0)}}\right) & = & \sum\limits_{k=0}^{\tilde{d}} \frac{\partial^k r_d\left(x_0, \mathbf{x^{(0)}}, \mathbf{y^{(0)}}\right)}{\partial x^k} \frac{(x-x_0)^k}{k!} \\ r_d\left(x_n, \mathbf{x^{(n)}}, \mathbf{y^{(n)}}\right) & = & \sum\limits_{k=0}^{\tilde{d}} \frac{\partial^k r_d\left(x_n, \mathbf{x^{(n)}}, \mathbf{y^{(n)}}\right)}{\partial x^k} \frac{(x-x_n)^k}{k!}, \end{cases} \tag{6.20}$$

lembrando que $\mathbf{x^{(0)}} := (x_0, x_1, \ldots, x_{\tilde{n}})$ e $\mathbf{x^{(n)}} := (x_{n-\tilde{n}}, x_{n-\tilde{n}+1}, \ldots, x_n)$. Dessa forma, podemos utilizar também os algoritmos[1] do Tipo I descritos na Seção 5.1 para calcular os valores extrapolados (6.4).

No processo de extrapolação proposto por Klein, cada $\tilde{y}_{-\tau}$ é gerado pela extrapolação de funções construídas a partir dos dados $x_0, x_1, \ldots, x_{\tilde{n}}$ e $y_0, y_1, \ldots, y_{\tilde{n}}$. Porém, nada impede que utilizemos os próprios valores extrapolados já calculados para gerar os demais valores. Por exemplo, para cada valor de τ, podemos utilizar

$$x_{-\tau+1}, x_{-\tau+2}, \ldots, x_{-\tau+\tilde{n}+1} \quad \text{e} \quad \tilde{y}_{-\tau+1}, \tilde{y}_{-\tau+2}, \ldots, \tilde{y}_{-\tau+\tilde{n}+1} \tag{6.21}$$

para gerar $\tilde{y}_{-\tau}$ e, analogamente,

$$x_{n+\tau-\tilde{n}-1}, x_{n+\tau-2}, \ldots, x_{n+\tau-1} \quad \text{e} \quad \tilde{y}_{n+\tau-\tilde{n}-1}, \tilde{y}_{n+\tau-2}, \ldots, \tilde{y}_{n+\tau-1} \tag{6.22}$$

para gerar $\tilde{y}_{n+\tau}$. Denominaremos o procedimento descrito por (6.21) e (6.22) por estratégia de extrapolação de Neville, devido à analogia que podemos fazer com a estratégia de Neville no contexto de eliminação [Müh90], [AGP97], na qual cada bloco com $\tilde{n}+1$ zeros numa mesma linha de uma matriz são gerados por eliminação Gaussiana utilizando as $\tilde{n}+1$ linhas imediatamente anteriores à linha em questão.

No caso (6.11), pode-se mostrar (por indução em $\tau \geq 2$) que os polinômios de grau $\tilde{d} = d$ que interpolam os dados (6.21) são todos idênticos ao polinômio que interpola (6.21) para $\tau = 1$ e isso mostra que a estratégia de Neville é idêntica à estratégia de extrapolação proposta por Klein em aritmética exata (sem erros de arredondamento.) Porém, a estratégia de Neville se mostra mais estável numericamente na prática, como podemos constatar na Figura 6.9. Para os valores com a legenda $EFH(Tay)$, os valores extrapolados $\tilde{\mathbf{y}}$ foram obtidos pela avaliação numérica das expressões no lado direito de (6.4). Para os valores com a legenda $EFH(Lag)$, os valores extrapolados $\tilde{\mathbf{y}}$ foram obtidos pela avaliação numérica das expressões no lado esquerdo de (6.20), utilizando-se um algoritmo do Tipo I. Para os valores com a legenda $EFH_{Nev}(Lag)$, os valores extrapolados $\tilde{\mathbf{y}}$ foram obtidos utilizando-se a estratégia de extrapolação de Neville, com um algoritmo do Tipo I para avaliar os interpoladores definidos pelos pontos (6.21) e (6.22).

[1] Não recomendamos o uso dos algoritmos do Tipo II para este fim, pois os algoritmos do Tipo I são mais estáveis para extrapolação.

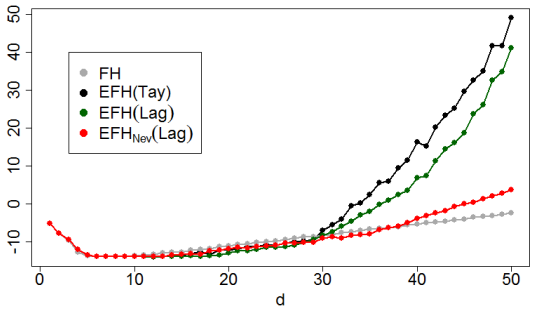
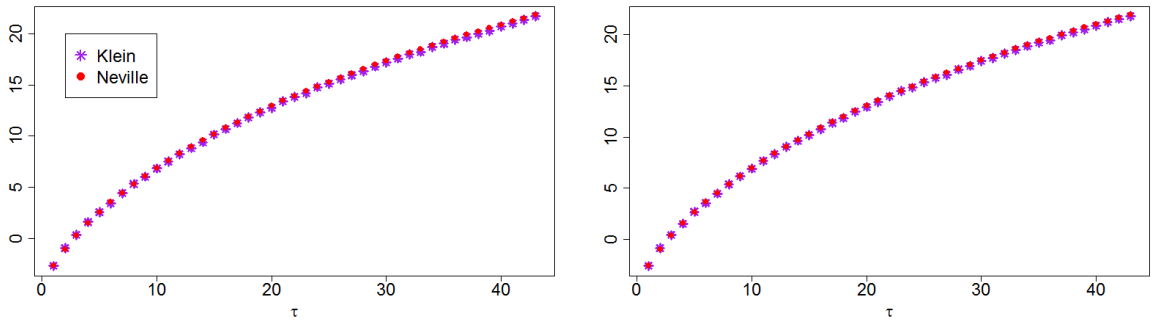


Figura 6.9: *O erro logaritmico* $\log_{10}\left(\max_{x\,\in\,[-5,5]}|I(x)-f(x)|\right)$ *para* $I = FH$ *e* $I = EFH$ $(\kappa = \tilde{d} = \tilde{n} = d)$ *e* $n = 10^3$.

A idéia de se utilizar a estratégia de Neville surgiu de um argumento heurístico baseado na convexidade da função de Lebesgue para o interpolador de Lagrange fora do intervalo de interpolação. Porém, uma análise mais detalhada sobre a estabilidade numérica dos algoritmos que utilizam o processo de Neville não aparenta ser simples. Por exemplo, embora a estratégia de Neville apresente uma melhora efetiva em relação à estratégia de Klein no processo global de aproximação, os erros introduzidos nos dois processos de extrapolação possuem a mesma ordem de magnitude, como mostrado na Figura 6.10, para $d = 43$. Em todo caso, dadas as limitações das vantagens obtidas com a estratégia de Neville em relação aos interpoladores de Floater-Hormann, tal análise não se faz tão necessária. Observe que há uma melhora efetiva em relação ao processo de extrapolação proposto por Klein, porém ainda assim não conseguimos uma melhora expressiva sobre os interpoladores de Floater-Hormann (ver Figura 6.9.)

Na Figura 6.10, os valores extrapolados de referência (exatos) foram obtidos calculando-se o interpolador de Lagrange com um algoritmo do Tipo I em precisão mais alta (50 casas decimais) com o auxílio da biblioteca MPFR. Para os valores com a legenda **Klein**, os valores extrapolados $\tilde{\mathbf{y}}$ foram obtidos pela avaliação numérica das expressões no lado esquerdo de (6.20), utilizando-se um algoritmo do Tipo I.



(a) (b)

Figura 6.10: *O erro logaritmico nos valores extrapolados para* $d = 43$*:* $\log_{10}|fl(\tilde{y}_{-\tau}) - \tilde{y}_{-\tau}|$ *(a); e* $\log_{10}|fl(\tilde{y}_{n+\tau}) - \tilde{y}_{n+\tau}|$ *(b). Em ambos os processos de extrapolação, os novos pontos foram calculados com um algoritmo do Tipo I (utilizando-se a forma de Lagrange para o polinômio interpolador.)*

Capítulo 7

Conclusões

Neste trabalho apresentamos um estudo detalhado sobre a estabilidade numérica da fórmula baricêntrica para interpolação. Nos Capítulos 1, 2 e 3 apresentamos os principais conceitos relacionados à estabilidade da fórmula baricêntrica e também uma breve revisão bibliográfica sobre alguns interpoladores e suas relações com a fórmula baricêntrica. No Capítulo 4 estudamos a sensibilidade da fórmula baricêntrica genérica com relação à perturbações dos seus parâmetros e concluímos que a fórmula baricêntrica possui a propriedade de estabilidade backward quando a constante de Lebesgue associada aos nós de interpolação é pequena. Esse resultado generaliza os resultados obtidos em [Mas14] para interpolação nos nós de Chebyshev do segundo tipo. No Capítulo 5 focamos no caso particular dado pelo interpolador de Floater-Hormann e apresentamos um tipo de algoritmo que possui a propriedade de estabilidade backward sobre toda a reta real. Nossos experimentos numéricos mostraram que esses algoritmos (Tipo I) são mais estáveis do que os algoritmos que avaliam a fórmula baricêntrica (Tipo II) para extrapolação. A nossa análise generaliza os resultados apresentados em [Hig04] e [WTG12] para o interpolador de Lagrange. No Capítulo 6 mostramos que os interpoladores de Floater-Hormann estendidos são menos estáveis do que se fora anteriormente descrito na literatura. Provamos que, em alguns casos, a constante de Lebesgue para os interpoladores estendidos possui ordem de crescimento exponencial com relação ao parâmetro que define a sua ordem de aproximação e, caso geral, verificamos esse fato experimentalmente.

De forma resumida, as principais conclusões desse trabalho são

- A fórmula baricêntrica possui a propriedade de estabilidade backward quando a constante de Lebesgue associada aos nós de interpolação é pequena.
- Os algoritmos do Tipo I para os interpoladores de Floater-Hormann possuem a propriedade de estabilidade backward sobre toda a reta real e são mais estáveis do que os algoritmos do Tipo II para extrapolação. Para interpolação, os algoritmos do Tipo I e do Tipo II apresentam comportamento similar nos casos mais comuns.
- Os interpoladores de Floater-Hormann estendidos não são significativamente mais estáveis do que os interpoladores de Floater-Hormann e a constante de Lebesgue associada a esses interpoladores possui crescimento exponencial em função da ordem de aproximação.

7.1 Considerações Finais

Neste trabalho utilizamos as propriedades fundamentais da aritmética de precisão finita para analisar a propagação dos erros de arredondamento na avaliação numérica de fórmulas baricêntricas para interpolação. Porém, é importante frisar que, embora os resultados obtidos no nosso estudo sejam específicos sobre interpolação, as técnicas de manipulação de expressões algébricas para o desenvolvimento de algoritmos estáveis são de carater geral e podem ser aplicadas nos mais variados contextos. Nesse sentido, consideramos que o aprendizado decorrente desse estudo vai além da especialização em tópicos sobre interpolação.

Ressaltamos, também, a importância da realização de experimentos numéricos para obter uma melhor compreensão dos fenômenos estudados. Por um lado, os resultados das simulações podem ajudar na obtenção de uma explicação lógica para o fenômeno. Por outro lado, os experimentos numéricos servem para comprovar (ou contradizer) hipóteses e teses formuladas a partir de argumentos teóricos. Um bom exemplo disso são as considerações sobre a estabilidade forward dos algoritmos dos Tipos I e II feitas na Seção 5.3.

Por fim, a discussão apresentada no Capítulo 6 (em especial a Seção 6.2.2) enfatiza a necessidade da análise mútua dos erros (teórico/abstrato e numérico) que compõem o processo de aproximação.

7.2 Sugestões para Pesquisas Futuras

Uma das aplicações diretas da aproximação de funções por interpolação é o desenvolvimento de fórmulas lineares para integração

$$\int_a^b f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i)$$

como, por exemplo, as integrais de Newton-Cotes, Gaussiana ([IK94], cap. 7) e de Romberg [Mil68]. Nesse sentido, um estudo sobre a estabilidade de fórmulas numéricas para integração pode ser considerado como uma complementação natural do nosso trabalho.

Em vista do insucesso dos interpoladores de Floater-Hormann estendidos com relação à melhora do condicionamento/estabilidade dos interpoladores de Floater-Hormann, o problema de minimizar os efeitos dos erros de arredondamento na avaliação numérica dos interpoladores de Floater-Hormann também pode ser considerado como um problema relevante para estudos futuros.

Uma outra direção é o estudo de interpolação polinomial em mais do que uma variável. Nesse contexto, a teoria está ainda sob construção e há diversos problemas para serem estudados. Por exemplo, a única família de pontos para interpolação em $[-1,1]^2$ com polinômios bivariados de grau completo menor ou igual a n com constante de Lebesgue quase ótima ($O(log^2(n))$) conhecida são os *Pontos de Padua* [BCM+06]. Ainda não se sabe se existe uma familia de pontos para interpolação em $[-1,1]^3$ (com polinômios trivariados de grau completo menor ou igual a n) que possua contante de Lebesgue da ordem de $\log^3(n)$. Até mesmo problemas mais fundamentais como a existência e a unicidade de interpoladores polinomiais em duas ou mais variáveis ainda não são completamente compreendidos [Bos91].

Referências Bibliográficas

[AGP97] Pedro Alonso, Mariano Gasca e Juan M. Peña. Backward error analysis of Neville elimination. *Applied Numerical Mathematics*, 23:193–204, 1997. 63

[BCM+06] Len Bos, Marco Caliari, Stefano De Marchi, Marco Vianello e Yuan Xu. Bivariate Lagrange interpolation at the Padua points: the generating curve approach. *Journal of Aproximation Theory*, 143:15–25, 2006. 66

[Ber98] Jean P. Berrut. Rational functions for guaranteed and experimentally well conditioned global interpolation. *Computers & Mathematics with Applications*, 15:1–16, 1998. 17

[BMHK12] Len Bos, Stefano De Marchi, Kai Hormann e Georges Klein. On the Lebesgue constant of barycentric rational interpolation at equidistant nodes. *Numerische Mathematik*, 121:461–471, 2012. 3, 19

[BMHS13] Len Bos, Stefano De Marchi, Kai Hormann e Jean Sidon. Bounding the Lebesgue constant for Berrut's rational interpolant at general nodes. *Journal of Approximation theory*, 169:7–22, 2013. 3

[Bos91] Len Bos. On certain configurations of points in R^n which are unisolvent for polynomial interpolation. *Journal of Aproximation Theory*, 64:271–280, 1991. 66

[BP78] Carl De Boor e Allan Pinkus. Proof of the conjecture of Bernstein and Erdös concerning the optimal nodes for polynomial interpolation. *Journal of Approximation Theory*, 24:289–303, 1978. 14

[Bru97] Lev Brutman. Lebesgue functions for polynomial interpolation – a survey. *Annals of Numerical Mathematics*, 4:111–127, 1997. 61

[Cel08] Oliver S. Celis. *Practical rational interpolation for exact and inexact data: Theory and algorithms*. Tese de Doutorado, Universidade de Antuérpia, Bélgica, Julho 2008. 45

[dC15] André P. de Camargo. On the numerical stability of Floater-Hormanns rational interpolant. *Numerical Algorithms*, DOI 10.1007/s11075-015-0037-z, 2015. 40

[Epp87] James F. Epperson. On the Runge example. *American Mathematical Monthly*, 94:329–341, 1987. 11

[Erd64] Paul Erdös. Problems and results on the theory of interpolation. II. *Acta Mathematica Hungarica*, 12:235–244, 1964. 13

[FH07] Michael S. Floater e Kai Hormann. Barycentric rational interpolation with no poles and high rates of approximation. *Numerische Mathematik*, 107:315–331, 2007. 17, 18, 19, 21

[GK12] Stefan Güttel e Georges Klein. Convergence of linear barycentric rational interpolation for analytic functions. *SIAM Journal on Numerical Analysis*, 50:2560–2580, 2012. 19

[Gün80] Rüdiger Günttner. Evaluation of Lebesgue constants. *SIAM journal on Numerical Analysis*, 17:512–520, 1980. 13

[Hen64] Peter Henrici. *Elements of Numerical Analysis*. John Wiley & Sons, Inc., primeira edição, 1964. 11

[Hen82] Peter Henrici. *Essentials of Numerical Analysis With Pocket Calculator Demonstrations*. John Wiley & Sons, Inc., primeira edição, 1982. 9, 10, 11, 17

[Hig02] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM: Society for Industrial and Applied Mathematics, segunda edição, 2002. 5, 6

[Hig04] Nicholas J. Higham. The numerical stability of barycentric Lagrange interpolation. *IMA Journal of Numerical Analysis*, 24:547–556, 2004. 1, 25, 26, 37, 45, 65

[HKM12] Kai Hormann, Georges Klein e Stefano De Marchi. Barycentric rational interpolation at quasi-equiditant nodes. *Dolomites Research Notes on Approximation*, 5:1–6, 2012. 19

[IK94] Eugene Isaacson e Herbert B. Keller. *Analysis of numerical methods*. Dover, primeira edição, 1994. 9, 11, 66

[KB12] Georges Klein e Jean P. Berrut. Linear rational finite differences from derivatives of barycentric rational interpolants. *SIAM Journal on Numerical Analysis*, 50:643–656, 2012. 5

[Kil77] Theodore A. Kilgore. Optimization of the norm of the Lagrange interpolation operator. *Bulletin of the American Mathematical Society*, 83:1069–1071, 1977. 14

[Kil78] Theodore A. Kilgore. A characterization of the Lagrange interpolating projection with minimal Tchebysheff norm. *Journal of Approximation Theory*, 24:273–288, 1978. 14

[Kle13] Georges Klein. An extension of the Floater-Hormann family of barycentric rational interpolants. *Mathematics of Computation*, 82:2273–2292, 2013. 20, 21, 51, 52, 55, 56, 57, 58

[Lev07] Randall J. Levegue. *Finite difference methods for ordinary and partial differential equations*. Society for Industrial and Applied Mathematics (SIAM), primeira edição, 2007. 10

[Mas14] Walter F. Mascarenhas. The stability of barycentric interpolation at the Chebyshev points of the second kind. *Numerische Mathematik*, 128:265–300, 2014. 1, 26, 38, 46, 65

[MdC14] Walter F. Mascarenhas e André P. de Camargo. The backward stability of the second barycentric formula for interpolation. *Dolomites research notes on approximation*, 7:1–12, 2014. 3, 33

[MdC16] Walter F. Mascarenhas e André P. de Camargo. The effects of rounding errors in the nodes on barycentric interpolation. *Numerische Mathematik*, ??:??, 2016. 26, 27, 28, 33, 37, 40, 41, 50

[Müh90] Günter Mühlbach. On extending determinantal identities. *Linear Algebra and its Applications*, 132:145–162, 1990. 63

[Mil68] Jeffrey Charles P. Miller. Neville's and Romberg's processes: A fresh apparaisal with applications. *Philosophical Transactions of the Royal Society of London Serie A*, 263:525–562, 1968. 66

[MP73] John H. McCabe e George M. Phillips. On a certain class of Lebesgue constants. *BIT Numerical Mathematics*, 13:434–442, 1973. 13

[NW99] Jorge Nocedal e Stephen J. Wright. *Numerical optimization*. Springer-Verlag, primeira edição, 1999. 14

[Pow81] Michael J. D. Powell. *Approximation theory and methods*. Cambridge university press, primeira edição, 1981. 12

[PQ08] Dilcia Pérez e Yamilet Quintana. A survey on the Weierstrass approximation theorem. *Divulgaciones Matemáticas*, 16:231–247, 2008. 12

[Sal72] Herbert E. Salzer. Lagrange interpolation at Chebyshev points $\mathbf{X}_{\mathbf{n},\nu} = \cos(\nu\pi/n), \nu = 0(1)n$; some unnoted advantages. *The computer Journal*, 15:156–159, 1972. 12, 27

[Smi06] Simon J. Smith. Lebesgue constants in polynomial interpolation. *Annales Mathematicae et Informaticae*, 33:109–123, 2006. 10

[SW86] Claus Schneider e Wilhelm Werner. Some new aspects of rational interpolation. *Mathematics of computation*, 47:285–299, 1986. 5

[Tre12] Lloyd N. Trefethen. *Approximation theory and approximation practice*. Society for Industrial and Applied Mathematics (SIAM), 2012. 10

[TW91] Lloyd N. Trefethen e Jacob Andreas. C. Weideman. Two results on polynomial interpolation in equally spaced points. *Journal of Approximation Theory*, 365:247–260, 1991. 13

[WTG12] Marcus Webb, Lloyd N. Trefethen e Pedro Gonnet. Stability of barycentric interpolation formulas for extrapolation. *SIAM Journal on Scientific Computing*, 34:A3009–A3015, 2012. 37, 44, 49, 65

Índice Remissivo