

**Technische Universität Ilmenau**  
**Fakultät für Mathematik**  
**und Naturwissenschaften**  
**Institut für Mathematik**

[http://www.mathematik.tu-ilmenau.de/Math-Net/index\\_de.html](http://www.mathematik.tu-ilmenau.de/Math-Net/index_de.html)

Postfach 10 05 65  
D - 98684 Ilmenau  
Germany  
Tel.: 03677/69 3267  
Fax: 03677/69 3272  
Telex: 33 84 23 tuil d.  
email: [werner.neundorf@tu-ilmenau.de](mailto:werner.neundorf@tu-ilmenau.de)

Preprint No. M 16/04

# **Polynome, Interpolation, Splines und Differentiation**

Werner Neundorf

September 2004

---

<sup>‡</sup>MSC (2000): 11-01, 11C08, 65D05, 65D07, 65T05

## Zusammenfassung

Rechnerunterstütztes Entwerfen ist eine Disziplin, die in vielen Bereichen des Ingenieurwesens von zunehmender Bedeutung ist.

Man bezeichnet diesen Prozess als Computer Aided Design oder bei Schwerpunktlegung auf seine mathematische Seite als Computer Aided Geometric Design und versteht darunter all die Techniken, bei denen Computer zum Entwurf von Produkten Verwendung finden. Diese Produkte (Modelle) basieren auf einer mathematischen Beschreibung der geometrischen Form als Ganzes oder auf diskrete Weise, die es beispielsweise erlaubt, Zeichnungen zu erstellen oder Befehle für numerisch gesteuerte Werkzeugmaschinen zu erzeugen.

Ob in der Computergrafik, in der Auswertung von Messdaten oder bei der Funktionsdarstellung, es bedarf dazu der Approximation zumeist komplizierter Funktionen und speziell auch der Interpolation dieser mittels unterschiedlicher einfacherer Ansatzfunktionen.

Hier geben wir einen Überblick über die Interpolation mittels Polynomen und trigonometrischen Funktionen sowie Splines, wie dies auch in meinem Buch *Numerische Mathematik – Vorlesungen, Übungen, Algorithmen und Programme* zu finden ist.

Natürlich empfiehlt sich bei der praktischen Lösung solcher Aufgaben auch der Einsatz des Computers und von Software. Hier sind einige Rechnungen mit dem Computeralgebrasystem Maple V durchgeführt worden.

# Inhaltsverzeichnis

<b>1</b>	<b>Polynome, Interpolation, Splines und Differentiation</b>	<b>1</b>
1.1	Polynome . . . . .	1
1.1.1	Einfaches Horner-Schema . . . . .	1
1.1.2	Vollständiges Horner-Schema . . . . .	2
1.1.3	Inverses Horner-Schema . . . . .	3
1.2	Allgemeines Interpolationsproblem . . . . .	5
1.3	Polynominterpolation im $\mathbb{R}^1$ . . . . .	5
1.3.1	Lagrange-Interpolation . . . . .	8
1.3.2	Kondition der Interpolation . . . . .	11
1.3.3	Newton-Interpolation . . . . .	19
1.3.4	Polynomwertberechnung . . . . .	29
1.4	Hermite-Interpolation . . . . .	31
1.4.1	Bestimmung des Hermite-Interpolationspolynoms . . . . .	32
1.4.2	Allgemeine Referenz . . . . .	35
1.4.3	Kubische Hermite-Basis . . . . .	36
1.5	Fehler und Konvergenz . . . . .	38
1.5.1	Fehlerfortpflanzung in Schemata . . . . .	38
1.5.2	Konvergenz der Interpolation . . . . .	38
1.5.3	Konvergenzsätze . . . . .	43
1.5.4	Bemerkungen zur Wahl der Tschebyscheff-Referenz . . . . .	45
1.6	Trigonometrische Interpolation . . . . .	47
1.6.1	Übersicht und Merkmale . . . . .	47
1.6.2	Komplexe und reelle trigonometrische Interpolation . . . . .	50
1.6.3	Diskrete Fourier-Transformation . . . . .	55

1.7	Splineinterpolation im $\mathbb{R}^1$ . . . . .	64
1.7.1	Einfache Typen von Splines . . . . .	65
1.7.2	Kubische Splines . . . . .	70
1.7.3	B-Splines . . . . .	74
1.7.4	Krümmung einer Kurve . . . . .	89
1.7.5	Parametrische Splines zur Kurvendarstellung . . . . .	92
1.8	Differentiation . . . . .	93
	Literaturverzeichnis . . . . .	94
	Index . . . . .	98

# Kapitel 1

## Polynome, Interpolation, Splines und Differentiation

### 1.1 Polynome

Gegeben sei das reelle Polynom  $n$ -ten Grades in seiner Normalform

$$p_n(x) = \sum_{i=0}^n a_i x^{n-i} = a_0 x^n + a_1 x^{n-1} + \dots + a_{n-2} x^2 + a_{n-1} x + a_n. \quad (1.1)$$

Auf das Polynom kann man verschiedene Varianten des Horner-Schemas zur Berechnung von Funktions- und Ableitungswerten sowie zur Transformation zwischen den Darstellungsformen anwenden.

#### 1.1.1 Einfaches Horner-Schema

Sei  $p_n(x)$  das Polynom und  $x_0$  ein Argument. Das einfache Horner-Schema zur Berechnung von  $p_n(x_0)$  entsteht durch sukzessives Ausklammern von  $x$  in der Normalform.

$$\begin{aligned} p_n(x) &= a_0 x^n + a_1 x^{n-1} + \dots + a_{n-2} x^2 + a_{n-1} x + a_n \\ &= (a_0 x^{n-1} + a_1 x^{n-2} + \dots + a_{n-2} x + a_{n-1}) x + a_n \\ &= ([a_0 x^{n-2} + a_1 x^{n-3} + \dots + a_{n-2}] x + a_{n-1}) x + a_n \\ &= \dots \\ &= ([\dots \underbrace{\{ \underbrace{a_0}_{b_0} x + a_1 \}}_{b_1} x + \dots + a_{n-2}] x + a_{n-1}) x + a_n \\ &\quad \underbrace{\hspace{10em}}_{b_{n-1}} \\ &\quad \underbrace{\hspace{15em}}_{b_n} \end{aligned}$$

Die schematische Darstellung ist

$$\begin{array}{c|cccccc}
 & a_0 & a_1 & a_2 & \dots & a_{n-1} & a_n \\
 x_0 & & b_0x_0 & b_1x_0 & \dots & b_{n-2}x_0 & b_{n-1}x_0 \\
 \hline
 & b_0 & b_1 & b_2 & \dots & b_{n-1} & b_n = p_n(x_0)
 \end{array} \quad (1.2)$$

Die Werte  $b_i$  der letzten Zeile werden rekursiv berechnet gemäß

$$\begin{aligned}
 b_0 &= a_0, \\
 b_i &= b_{i-1}x_0 + a_i, \quad i = 1, 2, \dots, n.
 \end{aligned} \quad (1.3)$$

Die Größen  $b_i$  sind die Koeffizienten des Polynoms

$$p_{n-1}(x) = \sum_{i=0}^{n-1} c_i x^{n-1-i}, \quad (1.4)$$

also  $c_i = b_i$ , das den Beziehungen

$$\begin{aligned}
 p_n(x) &= p_{n-1}(x)(x - x_0) + p_n(x_0), \\
 p'_n(x) &= p'_{n-1}(x)(x - x_0) + p_{n-1}(x), \\
 p'_n(x_0) &= p_{n-1}(x_0)
 \end{aligned} \quad (1.5)$$

genügt.

Ist  $x_0$  Nullstelle von  $p_n(x)$ , so beinhaltet das Horner-Schema die Abspaltung des Linearfaktors  $(x - x_0)$ . Das um einen Grad reduzierte Polynom  $p_{n-1}(x)$  kann für eine weitere Nullstellenuntersuchung betrachtet werden. Der Prozess der schrittweisen Abspaltung heißt auch **Deflation**. Er ist aber numerisch instabil.

### 1.1.2 Vollständiges Horner-Schema

Sei  $p_n(x)$  ein reelles Polynom  $n$ -ten Grades und  $x_0$  ein gegebenes Argument.

In den Formeln (1.5) ist eine Möglichkeit der Fortsetzung des einfachen Horner-Schemas zu erkennen. Zur Ermittlung von Funktionswert und 1. Ableitung von  $p_n(x)$  an der Stelle  $x_0$  nimmt man das zweizeilige Horner-Schema, denn  $p'_n(x_0) = p_{n-1}(x_0)$ . Mit der allgemeinen Version erfolgt die Berechnung von Funktionswert und aller Ableitungen.

Es gilt im nachfolgenden Schema

$$\begin{aligned}
 p_n(x_0) &= b_n^{(0)}, \quad p'_n(x_0) = b_{n-1}^{(1)}, \quad p''_n(x_0) = 2!b_{n-2}^{(2)}, \dots, \quad p_n^{(n)}(x_0) = n!b_0^{(n)}, \\
 p_n^{(n)}(x_0) &= n!a_0, \quad p_n^{(k)}(x) = 0 \quad \text{für } k > n.
 \end{aligned} \quad (1.6)$$

## Vollständiges Horner-Schema

$x_0$	$a_0$	$a_1$	$a_2$	$\dots$	$a_{n-3}$	$a_{n-2}$	$a_{n-1}$	$a_n$
	$b_0^{(0)} x_0$	$b_1^{(0)} x_0$	$\dots$	$b_{n-4}^{(0)} x_0$	$b_{n-3}^{(0)} x_0$	$b_{n-2}^{(0)} x_0$	$b_{n-1}^{(0)} x_0$	$b_n^{(0)} = p_n(x_0)$
$x_0$	$b_0^{(0)}$	$b_1^{(0)}$	$b_2^{(0)}$	$\dots$	$b_{n-3}^{(0)}$	$b_{n-2}^{(0)}$	$b_{n-1}^{(0)}$	
	$b_0^{(1)} x_0$	$b_1^{(1)} x_0$	$\dots$	$b_{n-4}^{(1)} x_0$	$b_{n-3}^{(1)} x_0$	$b_{n-2}^{(1)} x_0$		
$x_0$	$b_0^{(1)}$	$b_1^{(1)}$	$b_2^{(1)}$	$\dots$	$b_{n-3}^{(1)}$	$b_{n-2}^{(1)}$	$b_{n-1}^{(1)} = p_{n-1}(x_0)$	
	$b_0^{(2)} x_0$	$b_1^{(2)} x_0$	$\dots$	$b_{n-4}^{(2)} x_0$	$b_{n-3}^{(2)} x_0$			
	$b_0^{(2)}$	$b_1^{(2)}$	$b_2^{(2)}$	$\dots$	$b_{n-3}^{(2)}$	$b_{n-2}^{(2)} = p_{n-2}(x_0)$		
	.....							
$x_0$	.....							
	$b_0^{(n-2)} \quad b_1^{(n-2)}$		$b_2^{(n-2)} = p_2(x_0)$					
$x_0$	$b_0^{(n-1)} x_0$							
	$b_0^{(n-1)}$	$b_1^{(n-1)} = p_1(x_0)$						
$x_0$								
	$b_0^{(n)} = p_0(x_0)$							

## 1.1.3 Inverses Horner-Schema

Die Newtonsche Form des Polynoms mit gegebenen Stützstellen und Koeffizienten ist

$$\begin{aligned}
 p_n(x) &= c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \dots + c_n(x - x_0) \cdot \dots \cdot (x - x_{n-1}) \\
 &= \sum_{j=0}^n c_j \prod_{i=0}^{j-1} (x - x_i).
 \end{aligned} \tag{1.7}$$

Die Ermittlung seiner Normalform  $\sum_{i=0}^n a_i x^{n-i}$  erfolgt unter Anwendung des inversen Horner-Schemas auf der Basis der geklammerten Darstellung

$$\begin{aligned}
 p_n(x) &= c_0 + (x - x_0) \{ c_1 + (x - x_1) [ c_2 + \dots + (x - x_{n-2}) ( \underbrace{c_{n-1}}_{a_1^{(0)}} + (x - x_{n-1}) \underbrace{c_n}_{a_0^{(0)}} ) \dots ] \}, \\
 &= c_0 + (x - x_0) \{ c_1 + (x - x_1) [ c_2 + \dots + (x - x_{n-2}) ( \underbrace{c_n}_{a_0^{(1)}} x + \underbrace{c_{n-1} + (-x_{n-1})c_n}_{a_1^{(1)}} ) \dots ] \}.
 \end{aligned}$$

Analog berechnet man die Koeffizienten  $a_j^{(2)}$  mit  $a_2^{(1)} = c_{n-2}$  aus

$$c_{n-2} + (x - x_{n-2}) ( \underbrace{a_0^{(1)} x + a_1^{(1)}}_{a_0^{(2)}} ) = \underbrace{a_0^{(1)}}_{a_0^{(2)}} x^2 + [ \underbrace{a_1^{(1)} + (-x_{n-2})a_0^{(1)}}_{a_1^{(2)}} ] x + \underbrace{a_2^{(1)} + (-x_{n-2})a_1^{(1)}}_{a_2^{(2)}}.$$

**Inverses Horner-Schema** zur Berechnung der Normalform des Polynoms aus der Newtonschen Darstellung

	$c_n$					
$-x_{n-1}$	$a_0^{(0)} = c_n$	$a_1^{(0)} = c_{n-1}$				
		$-x_{n-1}a_0^{(0)}$				
$-x_{n-2}$	$a_0^{(1)}$	$a_1^{(1)}$	$a_2^{(1)} = c_{n-2}$			
		$-x_{n-2}a_0^{(1)}$	$-x_{n-2}a_1^{(1)}$			
$-x_{n-3}$	$a_0^{(2)}$	$a_1^{(2)}$	$a_2^{(2)}$			
	$\dots$					
$-x_0$	$a_0^{(n-1)}$	$a_1^{(n-1)}$	$a_2^{(n-1)}$	$\dots$	$a_{n-1}^{(n-1)}$	$a_n^{(n-1)} = c_0$
		$-x_0a_0^{(n-1)}$	$-x_0a_1^{(n-1)}$	$\dots$	$-x_0a_{n-2}^{(n-1)}$	$-x_0a_{n-1}^{(n-1)}$
	$a_0^{(n)}$	$a_1^{(n)}$	$a_2^{(n)}$	$\dots$	$a_{n-1}^{(n)}$	$a_n^{(n)}$
	$a_0$	$a_1$	$a_2$	$\dots$	$a_{n-1}$	$a_n$
	$x^n$	$x^{n-1}$	$x^{n-2}$	$\dots$	$x$	$1$

In den jeweiligen Spalten wird die Summe gebildet, d. h. für  $k = 1, 2, \dots, n$  berechnet man sukzessiv die Größen

$$a_0^{(k)} = a_0^{(k-1)}, \quad a_j^{(k)} = a_j^{(k-1)} + (-x_{n-k})a_{j-1}^{(k-1)}, \quad j = 1, 2, \dots, k. \quad (1.8)$$

Die vorletzte Zeile und somit die Koeffizienten der Normalform des Polynoms ergeben sich zu  $a_i = a_i^{(n)}$ .

Die Anwendung des Schemas ist natürlich auch in den speziellen Fällen wie

- $c_0 = c_1 = \dots = c_{n-1} = 0$ ,
- $x_0 = x_1 = \dots = x_{n-1}$ ,
- $c_0 = c_1 = \dots = c_{n-1} = 0, \quad x_0 = x_1 = \dots = x_{n-1}$ ,

möglich.



## 1.2 Allgemeines Interpolationsproblem

Interpolationsaufgaben treten in zahlreichen Anwendungen auf, aus denen drei ausgewählt werden.

- **Auswertung von Messungen**  
Die Messwerte  $(x_i, y_i)$ ,  $i = 0, 1, \dots, n$ , widerspiegeln einen unbekannten funktionalen Zusammenhang  $y = f(x)$ .  
Gesucht ist dafür näherungsweise eine glatte Funktion  $y = \phi(x)$ .
- **Computergrafik**  
Durch gegebene Punkte  $(x_i, y_i)$ ,  $i = 0, 1, \dots, n$ , durch Punkte mit erweiterten Bedingungen  $(x_i, y_i, y'_i)$  oder ähnliche sind möglichst glatte Kurven  $y = \phi(x)$  zu legen.
- **Approximation komplizierter Funktionen**  
Die bekannte Funktion  $y = f(x)$  ist im Computer effizient durch eine einfachere, aber hinreichend glatte Funktion  $y = \phi(x)$  anzunähern.

## 1.3 Polynominterpolation im $\mathbb{R}^1$

Zur Beschreibung der Polynominterpolation verwendet man die folgenden Größen und Bezeichnungen.

- **Grundintervall I**  
 $I = [a, b] \subset \mathbb{R}$  mit  $-\infty < a < b < \infty$  sowie (bekannte oder unbekannte) reelle Funktion  $f : I \rightarrow \mathbb{R}$ .
- **Stützstellen  $x_i$  und Stützwerte  $y_i$  sowie**

$$R_0 = \{(x_i, y_i) \mid a \leq x_0 < x_1 < x_2 < \dots < x_n \leq b\} \quad (1.9)$$

als **Referenz** (Punktfolge, Knotenfolge) mit den  $n+1$  paarweise verschiedenen **Stützstellen**  $x_i$  und den  $n+1$  zugehörigen **Stützwerten**  $y_i$ . Falls eine Funktion  $f(x)$  zu Grunde liegt, definiert man die Stützwerte i. Allg. gemäß  $y_i = f(x_i)$ ,  $i = 0, 1, \dots, n$ . Ohne Beschränkung der Allgemeinheit kann man die Stützstellen in geordneter Reihenfolge nehmen.

- **Interpolationspolynom**

$$p_n(x) = a_0 x^n + a_1 x^{n-1} + \dots + a_{n-1} x + a_n, \quad a_i \in \mathbb{R}. \quad (1.10)$$

Das ist der Ansatz als Polynom vom Grad  $\leq n$  mit den **Basisfunktionen**  $x^n, x^{n-1}, \dots, x, 1$ , die auch als monomiale Basis bezeichnet werden.

$\mathcal{P}_n$  sei der Raum der Polynome vom Grad  $\leq n$ , so dass  $p_n(x) \in \mathcal{P}_n$ .

- Die Interpolationsforderung (Interpolationsbedingung) ist

$$p_n(x_i) = y_i = f(x_i), \quad i = 0, 1, \dots, n. \quad (1.11)$$

- **Interpolationsaufgabe (IA1)**

Gesucht sind die Koeffizienten  $a_0, a_1, \dots, a_n$  in (1.10), so dass die Interpolationsforderung (1.11) erfüllt ist.

Weitere Interpolationsaufgaben ergeben sich durch verschiedene Varianten der Wahl der Basisfunktionen.

- (1) Die allgemeine lineare Interpolation

$$\phi(x) = a_0\phi_0(x) + a_1\phi_1(x) + \dots + a_n\phi_n(x)$$

beruht auf den **Basisfunktionen**  $\phi_i(x)$ .

- (2) Die trigonometrische Interpolation verwendet die Basis  $\{\sin(kx), \cos(kx)\}$  und den Ansatz

$$\begin{aligned} \phi(x) = & a_0 + a_1 \cos(x) + b_1 \sin(x) + a_2 \cos(2x) + b_2 \sin(2x) + \\ & \dots + a_m \cos(mx) + b_m \sin(mx). \end{aligned}$$

- (3) Bei der zweidimensionalen Polynominterpolation ist eine Notation des Ansatzes mit doppelindizierten Koeffizienten günstig, z. B. als

$$\begin{aligned} \phi(u, v) = & a_{00} + a_{10}u + a_{11}v + a_{20}u^2 + a_{21}uv + a_{22}v^2 + \\ & \dots + a_{m0}u^m + \dots + a_{mm}v^m. \end{aligned}$$

Vier Fragestellungen bzw. Probleme sind bei der Lösung der Interpolationsaufgabe (IA1) von Bedeutung.

1. Existiert zu jeder gegebenen Referenz  $R_0$  mit  $n + 1$  Punkten ein Interpolationspolynom  $p_n(x)$  vom Grad  $\leq n$ ?
2. Ist  $p_n(x)$  eindeutig bestimmt?
3. Wie kann  $p_n(x)$  effektiv und numerisch stabil konstruiert werden?
4. Wie erhält man eine Schätzung des **Interpolationsfehlers**

$$R_n(x) = f(x) - p_n(x) \quad (1.12)$$

für alle  $x \in I$ ?

### Satz 1.1 Existenz und Eindeutigkeit des Interpolationspolynoms

Die Interpolationsaufgabe (IA1) ist für beliebiges Intervall  $I$  und beliebige Referenzen  $R_0 \subset I \times \mathbb{R}$  stets eindeutig lösbar.

**Beweis.** Die Koeffizienten  $a_i$  sind Lösung des linearen Gleichungssystems (LGS)

$$\begin{pmatrix} x_0^n & x_0^{n-1} & \cdots & x_0^2 & x_0 & 1 \\ x_1^n & x_1^{n-1} & \cdots & x_1^2 & x_1 & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n-1}^n & x_{n-1}^{n-1} & \cdots & x_{n-1}^2 & x_{n-1} & 1 \\ x_n^n & x_n^{n-1} & \cdots & x_n^2 & x_n & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \cdots \\ a_{n-1} \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \cdots \\ y_{n-1} \\ y_n \end{pmatrix}. \quad (1.13)$$

Seine Koeffizientenmatrix ist wegen  $x_i \neq x_j$  bei  $i \neq j$  die reguläre Vandermondesche Matrix und führt auf die Vandermondesche Determinante

$$H = \begin{vmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{vmatrix} = \prod_{\substack{i,j=0 \\ i>j}}^n (x_i - x_j) \neq 0. \quad (1.14)$$

Letztere ist ein spezieller Fall der Haarschen Determinante bei allgemeiner Wahl von Basis- bzw. Ansatzfunktionen.

Der Nachweis von  $H \neq 0$  erfolgt durch Zurückführung der Determinante auf eine solche mit der Dimension  $n$ . Die Idee ist die, dass man ohne Änderung der Determinante in  $H$  die Spalten  $k = n+1, n, \dots, 2$  wie folgt umformt:

Man subtrahiert das  $x_0$ -fache der  $(k-1)$ -ten Spalte von der  $k$ -ten Spalte.

Damit entstehen in der ersten Zeile rechts vom Diagonalelement Nullen. Man kann mit dem Laplaceschen Entwicklungssatz nach dieser Zeile entwickeln und aus den weiteren Zeilen gemeinsame Faktoren  $x_i - x_0$  ausklammern.

$$\begin{aligned} H &= \begin{vmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & x_1 - x_0 & x_1(x_1 - x_0) & \cdots & x_1^{n-2}(x_1 - x_0) & x_1^{n-1}(x_1 - x_0) \\ 1 & x_2 - x_0 & x_2(x_2 - x_0) & \cdots & x_2^{n-2}(x_2 - x_0) & x_2^{n-1}(x_2 - x_0) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_n - x_0 & x_n(x_n - x_0) & \cdots & x_n^{n-2}(x_n - x_0) & x_n^{n-1}(x_n - x_0) \end{vmatrix} \\ &= \prod_{i=1}^n (x_i - x_0) \begin{vmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{vmatrix}. \end{aligned}$$

Mit der reduzierten Determinante wird weiter so verfahren. □

**Bemerkung 1.1** (1) Falls die Stützstellen  $x_i$  nahe beieinander liegen, ist  $H$  klein und die Kondition des LGS ist schlecht.

(2) Die Gültigkeit des Satzes bleibt bei einer Referenz mit paarweise verschiedenen Stützstellen erhalten.

**Beispiel 1.1** Die Wahl der Referenz erfolgt auf der Basis von  $x!$  bei  $n = 4$ .

$x_i$	0	1	2	3	4
$y_i$	1	1	2	6	24

Das LGS lautet

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 16 & 8 & 4 & 2 & 1 \\ 81 & 27 & 9 & 3 & 1 \\ 256 & 64 & 16 & 4 & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 2 \\ 6 \\ 24 \end{pmatrix}.$$

Seine Lösung und das Interpolationspolynom sind

$$p_4(x) = \frac{3}{8}x^4 - \frac{23}{12}x^3 + \frac{29}{8}x^2 - \frac{25}{12}x + 1.$$

Würde man mit zwei zusätzlichen Stützstellen 5 und 6 sowie den zugehörigen Stützwerten das Interpolationspolynom  $p_6(x)$  berechnen, so stellt man fest, dass dieses im Intervall  $[0,2]$  schlechter ist als  $p_4(x)$ .

### 1.3.1 Lagrange-Interpolation

Die **allgemeine Referenz** aus Stützstellen  $x_i$ ,  $i = 0, 1, \dots, n$ , mit Stützwerten  $y_i$  ist

$$R = \{(x_i, y_i) \mid a \leq x_i \leq b, \ i = 0, 1, \dots, n, \text{ und } x_i \neq x_j \text{ für } i \neq j\}. \quad (1.15)$$

Als **Ansatz für das Lagrange-Interpolationspolynom** schreibt man

$$L_n(x) = \sum_{k=0}^n y_k \varphi_k(x), \quad \varphi_k(x) = \prod_{\substack{i=0 \\ i \neq k}}^n \frac{x - x_i}{x_k - x_i},$$

wobei  $\varphi_k(x) = L_n^{(k)}(x)$  die Lagrange-Basispolynome, Knotenpunktpolynome oder Lagrange-Terme bezeichnen. Die Basispolynome genügen der Bedingung

$$\varphi_k(x_i) = \delta_{ik}, \quad \delta_{ik} \text{ Kronecker-Symbol.}$$

Damit ist einfach nachzurechnen, dass das **Lagrange-Interpolationspolynom**

$$L_n(x) = \sum_{k=0}^n y_k \prod_{\substack{i=0 \\ i \neq k}}^n \frac{x - x_i}{x_k - x_i} \quad (1.16)$$

höchstens  $n$ -ten Grades ist und die Bedingung  $L_n(x_j) = y_j$  erfüllt, also das eindeutig bestimmte Interpolationspolynom darstellt.

**Satz 1.2 Interpolationsfehler**

Sei  $f(x)$   $(n+1)$ -mal stetig differenzierbar auf dem Intervall  $I = [a, b]$  und  $L_n(x)$  das eindeutig bestimmte Interpolationspolynom vom Grad  $\leq n$  zur Referenz  $R$  mit  $y_i = f(x_i)$ .

Dann existiert eine Stelle  $\xi \in \text{int}(x_0, x_1, \dots, x_n) = I_1 \subset I$ , so dass für den Interpolationsfehler (Restglied der Interpolation) gilt

$$R_n(x) = f(x) - L_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x - x_0)(x - x_1) \cdots (x - x_n). \quad (1.17)$$

**Beweis.** Wir definieren die Hilfsfunktion

$$\varphi(t) = f(t) - L_n(t) - K(t - x_0)(t - x_1) \cdots (t - x_n), \quad K \text{ const.}$$

Wegen der Interpolationseigenschaft von  $L_n(x)$  gilt

$$\varphi(x_0) = 0, \quad \varphi(x_1) = 0, \dots, \quad \varphi(x_n) = 0.$$

Man kann nun die Konstante  $K$  so wählen, dass  $\varphi(t)$  an einem weiteren Punkt  $t = \bar{x} \neq x_i$ ,  $\bar{x} \in I_1$ , Null wird. Das passiert mit

$$K = \frac{f(\bar{x}) - L_n(\bar{x})}{\Pi_n(\bar{x})}, \quad \text{wobei } \Pi_n(x) = (x - x_0)(x - x_1) \cdots (x - x_n).$$

Somit ist  $\varphi(t)$  gleich Null für  $t = x_0, x_1, \dots, x_n$  und  $\bar{x}$ , also an  $n+2$  Stellen.

Jetzt wenden wir den Satz von Rolle an, der sagt, dass zwischen zwei Nullstellen einer differenzierbaren Funktion mindestens eine Nullstelle ihrer ersten Ableitung liegt.

$\varphi'(t)$  besitzt zwischen den  $n+2$  Nullstellen von  $\varphi(t)$  seine Nullstellen, das sind  $n+1$ .

Weiter hat  $\varphi''(t)$  zwischen den Nullstellen von  $\varphi'(t)$  seine, und das sind  $n$  Stück, usw.

Damit hat  $\varphi^{(n+1)}(t)$  mindestens eine Nullstelle  $\xi$  im Intervall  $I_1$ , das die Punkte  $x_i$  und  $\bar{x}$  enthält. Folglich erhält man

$$\varphi^{(n+1)}(\xi) = 0, \quad L_n^{(n+1)}(x) = 0, \quad \Pi_n^{(n+1)}(x) = (n+1)! \quad (\text{immer}),$$

$$\varphi^{(n+1)}(t) = f^{(n+1)}(t) - K(n+1)!, \quad t = \xi,$$

$$K = \frac{f^{(n+1)}(\xi)}{(n+1)!},$$

$$f(\bar{x}) - L_n(\bar{x}) = K \Pi_n(\bar{x}) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \Pi_n(\bar{x}).$$

Da  $\bar{x}$  beliebig verschieden von  $x_i$  ist, gilt auch

$$f(x) - L_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \Pi_n(x), \quad x \in I_1,$$

$$|f(x) - L_n(x)| \leq \frac{M_{n+1}}{(n+1)!} |\Pi_n(x)|, \quad M_{n+1} = \max_{x \in I_1} |f^{(n+1)}(x)|. \quad (1.18)$$

□

**Satz 1.3 Eigenschaften der Lagrange-Basispolynome**

$$(1) \quad 1 = \sum_{k=0}^n \varphi_k(x) \leq \sum_{k=0}^n |\varphi_k(x)|.$$

$$(2) \quad \sum_{k=0}^n \varphi_k(0) x_k^j = \begin{cases} 1 & \text{für } j = 0, \\ 0 & \text{für } j = 1, 2, \dots, n, \\ (-1)^n x_0 \cdot \dots \cdot x_n & \text{für } j = n+1. \end{cases}$$

(3) Die Polynome  $\varphi_k(x)$  bilden ein Orthogonalsystem (OGS) und damit eine Basis in  $\mathcal{P}_n$  mit dem Skalarprodukt  $(P, Q) = \sum_{i=0}^n P(x_i)Q(x_i)$ , d. h.  $(\varphi_k, \varphi_j) = \sum_{i=0}^n \varphi_k(x_i)\varphi_j(x_i) = \delta_{kj}$ .

(4) Das Lagrange-Interpolationspolynom  $L_n(x)$  für äquidistante Stützstellen basiert auf den Beziehungen

$$\begin{aligned} x_i &= x_0 + ih, \quad x = x_0 + th, \quad h > 0, \\ \varphi_k(x) &= \varphi_k(x_0 + th) = l_k(t) = \frac{(-1)^{n-k}}{k!(n-k)!} \prod_{\substack{i=0 \\ i \neq k}}^n (t-i), \\ L_n(x) &= \sum_{k=0}^n y_k l_k\left(\frac{x-x_0}{h}\right) = \sum_{k=0}^n y_k \frac{(-1)^{n-k}}{k!(n-k)!} \prod_{\substack{i=0 \\ i \neq k}}^n \left(\frac{x-x_0}{h} - i\right). \end{aligned} \tag{1.19}$$

**Beweis.** Wir zeigen nur die Aussagen (1) und (2).

Zu (1). Die Beziehung  $1 = \sum_{k=0}^n \varphi_k(x)$  bedeutet die Interpolation der konstanten Funktion  $f(x) = 1$  mit den Stützwerten 1 an den  $n+1$  Stützstellen  $x_k$ .

Zu (2). Es gilt  $\varphi_k(x) \in \mathcal{P}_n$  und  $\varphi_k(x_l) = \delta_{kl}$ . Wir machen eine Fallunterscheidung.

(a)  $0 \leq j \leq n$

Man betrachtet das Polynom

$$q(x) = \sum_{k=0}^n \varphi_k(x) x_k^j \in \mathcal{P}_n.$$

Für dieses gilt

$$q(x_l) = \sum_{k=0}^n \varphi_k(x_l) x_k^j = \sum_{k=0}^n \delta_{kl} x_k^j = x_l^j, \quad l = 0, 1, \dots, n.$$

Somit ist  $q(x)$  das Interpolationspolynom zur Referenz  $(x_0, x_0^j), (x_1, x_1^j), \dots, (x_n, x_n^j)$ . Andererseits geht auch das Polynom  $x^j$  durch dieselben Punkte, also ist  $q(x) = x^j$ .

Für  $j = 0$  erhalten wir  $1 = q(x) = q(0) = \sum_{k=0}^n \varphi_k(0) x_k^0 = \sum_{k=0}^n \varphi_k(0)$ .

Für  $j > 0$  folgt  $0 = 0^j = q(0) = \sum_{k=0}^n \varphi_k(0)x_k^j$ .

(b)  $j = n + 1$

Wir definieren nun das Polynom

$$r(x) = x^{n+1} - \sum_{k=0}^n \varphi_k(x)x_k^{n+1} \in \mathcal{P}_{n+1}.$$

Sein führender Koeffizient ist Eins. Seine Nullstellen sind  $x_0, x_1, \dots, x_n$  wegen

$$r(x_l) = x_l^{n+1} - \sum_{k=0}^n \varphi_k(x_l)x_k^{n+1} = x_l^{n+1} - x_l^{n+1} = 0.$$

Folglich können wir  $r(x)$  in Produktform schreiben als  $r(x) = \prod_{k=0}^n (x - x_k)$  und erhalten daraus

$$\begin{aligned} r(0) &= \prod_{k=0}^n (0 - x_k), \\ 0^{n+1} - \sum_{k=0}^n \varphi_k(0)x_k^{n+1} &= (-1)^{n+1} \prod_{k=0}^n x_k, \\ \sum_{k=0}^n \varphi_k(0)x_k^{n+1} &= (-1)^n x_0 x_1 \cdots x_n. \end{aligned}$$

□

### 1.3.2 Kondition der Interpolation

Betrachten wir das Lagrange-Interpolationspolynom  $p_n(x) \in \mathcal{P}_n$  zu gegebener Referenz  $R$  auf der Grundlage der Funktion  $f \in \mathcal{C}[a, b]$ . Im Funktionenraum  $\mathcal{C}[a, b]$  möge eine Norm mit ihren Eigenschaften analog zu denen der Vektornorm gegeben sein.

Um die Abhängigkeit des Interpolationspolynoms von  $f(x)$  und  $R$  zu unterstreichen, führen wir die folgenden Notationen ein.

$$\begin{aligned} p_n(x) &= p_n(f, R)(x), \\ \phi(f) &= p_n(f, R), \\ \phi(f)(x_i) &= p_n(f, R)(x_i) = f(x_i), \quad i = 0, 1, \dots, n. \end{aligned}$$

Damit ist  $\phi : \mathcal{C}[a, b] \rightarrow \mathcal{P}_n$  eine lineare Abbildung, denn für  $f, g \in \mathcal{C}[a, b]$  und  $\lambda \in \mathbb{R}$  gelten wegen der Linearität in der Darstellung von  $p_n(x)$

$$p_n(f + g, R) = p_n(f, R) + p_n(g, R), \quad p_n(\lambda f, R) = \lambda p_n(f, R).$$

Somit kann man untersuchen, wie sich Störungen in der Funktion  $f(x)$  auf das Interpolationspolynom auswirken. Diese Abhängigkeit der Lösung von den Ausgangsdaten wird durch die **Kondition der Interpolation** beschrieben.

**Definition 1.1 Absolute Kondition in der Maximumnorm**

Diese ist gegeben durch

$$\mathcal{K}_{abs} = \sup_{f \in \mathcal{C}[a,b]} \frac{\|p_n(f, R)\|_\infty}{\|f\|_\infty}, \text{ wobei } \|f\|_\infty = \max_{a \leq x \leq b} |f(x)|. \quad (1.20)$$

Wegen  $\sum_{k=0}^n |\varphi_k(x)| \geq 1$  gibt es noch eine andere Größe, die mit der Kondition zusammenhängt.

**Definition 1.2 Lebesgue-Konstante**

Die Lebesgue-Konstante  $\Lambda_n$  ist definiert durch

$$\Lambda_n = \max_{x \in [a,b]} \sum_{k=0}^n |\varphi_k(x)|, \text{ wobei } \varphi_k(x) \text{ die Lagrange-Basispolynome sind.} \quad (1.21)$$

Sie wird also durch die Verteilung der Stützstellen  $x_i$  beschrieben.

**Satz 1.4 Beziehung zwischen absoluter Kondition und Lebesgue-Konstante**

Für die Interpolationsaufgabe (IA1) mit  $R_0$ ,  $y_i = f(x_i)$ ,  $i = 0, 1, \dots, n$ , und  $f \in \mathcal{C}[a, b]$  gilt  $\mathcal{K}_{abs} = \Lambda_n$ .

**Beweis.** (1) Wir zeigen zunächst die Ungleichung  $\mathcal{K}_{abs} \leq \Lambda_n$ .

$$\begin{aligned} |p_n(f, R)(x)| &= \left| \sum_{k=0}^n f(x_k) \varphi_k(x) \right| \\ &\leq \sum_{k=0}^n |f(x_k) \varphi_k(x)| \\ &\leq \|f\|_\infty \sum_{k=0}^n |\varphi_k(x)| \\ &\leq \|f\|_\infty \max_{x \in [a,b]} \sum_{k=0}^n |\varphi_k(x)| \\ |p_n(f, R)(x)| &\leq \|f\|_\infty \Lambda_n \quad \forall x, \\ \|p_n(f, R)\|_\infty &\leq \|f\|_\infty \Lambda_n \quad \forall f, \\ \sup_{f \in \mathcal{C}[a,b]} \frac{\|p_n(f, R)\|_\infty}{\|f\|_\infty} &\leq \Lambda_n. \end{aligned}$$

(2) Es gibt eine Funktion  $g \in \mathcal{C}[a, b]$  mit  $\|p_n(g, R)\|_\infty = \|g\|_\infty \Lambda_n$ .

Sei  $\bar{x} \in [a, b]$  die Stelle mit  $\sum_{k=0}^n |\varphi_k(\bar{x})| = \max_{x \in [a,b]} \sum_{k=0}^n |\varphi_k(x)|$ .

Wir wählen  $g$  so, dass  $\|g\|_\infty = 1$  ist. Dazu nehmen wir die Stützwerte  $g(x_i) = \text{sign}(\varphi_i(\bar{x}))$  und verbinden die Punkte  $(x_i, g(x_i))$  durch einen stetigen Polygonzug  $g$ . Damit erhalten wir die geforderte Gleichheit.  $\square$



**Beispiel 1.2** Wir berechnen die Kondition der Interpolation bei Verwendung

äquidistanter Knoten  $x_i = \frac{2i}{n} - 1$  oder  $x_i = \frac{2i+1}{n+1} - 1 \in [-1, 1]$ , sowie

Tschebyscheff-Knoten  $x_i = \cos\left(\frac{2i+1}{2n+2}\pi\right) \in (-1, 1)$ .

$n$	Äquidistante Knoten	Tschebyscheff-Knoten
5	3.106	2.104
10	29.890	2.489
15	512.052	2.782
20	10 986.533	2.901

**Tab. 1.1** Kondition  $\mathcal{K}_{abs} = \Lambda_n$  für ausgewählte Knotenanzahlen mit 2 Verteilungen

**Beispiel 1.3** Lagrange-Interpolation und Kondition

Zu der Referenz aus 4 Punkten ( $n = 3$ ) auf  $I = [-2, 4]$

$x_i$	-2	1	2	4
$y_i$	3	1	-3	8

erhält man das Lagrange-Interpolationspolynom

$$L_3(x) = \sum_{k=0}^3 y_k \varphi_k(x) = \frac{2}{3}x^3 - \frac{3}{2}x^2 - \frac{25}{6}x + 6.$$

Die absolute Kondition beträgt

$$\mathcal{K}_{abs} = \Lambda_n = \max_{x \in [-2, 4]} \sum_{k=0}^3 |\varphi_k(x)| \approx \sum_{k=0}^3 |\varphi_k(-0.73)| \approx 3.598.$$

### Berechnungen und Grafik in Maple

```
# nm6ip1.mws
#
> restart: with(plots): with(plottools):
> # Arbeitsverzeichnis
> pfad := 'D:/Neundorf/Nwptexte/Math93/':
> # Graphik output files
> name1 := 'nm6ip01.ps': datei1 := cat(pfad, name1):
> name2 := 'nm6ip02.ps': datei2 := cat(pfad, name2):
> name3 := 'nm6ip03.ps': datei3 := cat(pfad, name3):
> name4 := 'nm6ip04.ps': datei4 := cat(pfad, name4):
```

```

# Lagrange-Interpolation
# Referenz
> n := 3;
> xv := [-2, 1, 2, 4];
> yv := [ 3, 1,-3, 8];
#
# Lagrange-Polynome, Basispolynome
> phi0 := x-> (x-xv[2])*(x-xv[3])*(x-xv[4])/
>           ((xv[1]-xv[2])*(xv[1]-xv[3])*(xv[1]-xv[4]));
> simplify(phi0(x));
> phi1 := x-> (x-xv[1])*(x-xv[3])*(x-xv[4])/
>           ((xv[2]-xv[1])*(xv[2]-xv[3])*(xv[2]-xv[4]));
> simplify(phi1(x));
> phi2 := x-> (x-xv[1])*(x-xv[2])*(x-xv[4])/
>           ((xv[3]-xv[1])*(xv[3]-xv[2])*(xv[3]-xv[4]));
> simplify(phi2(x));
> phi3 := x-> (x-xv[1])*(x-xv[2])*(x-xv[3])/
>           ((xv[4]-xv[1])*(xv[4]-xv[2])*(xv[4]-xv[3]));
> simplify(phi3(x));

```

$$\text{phi0} := x \rightarrow \frac{(x - xv[2]) (x - xv[3]) (x - xv[4])}{(xv[1] - xv[2]) (xv[1] - xv[3]) (xv[1] - xv[4])} - \frac{1}{72} (x - 1) (x - 2) (x - 4)$$

$$\text{phi1} := x \rightarrow \frac{(x - xv[1]) (x - xv[3]) (x - xv[4])}{(xv[2] - xv[1]) (xv[2] - xv[3]) (xv[2] - xv[4])} - \frac{1}{9} (x + 2) (x - 2) (x - 4)$$

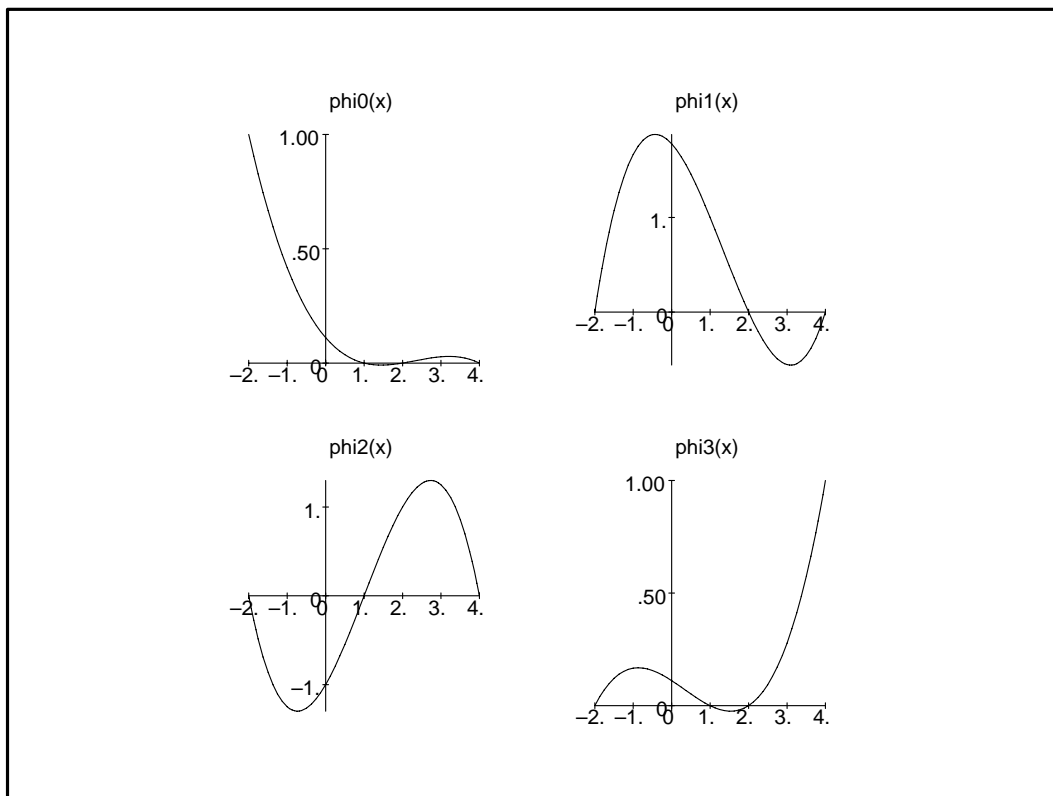
$$\text{phi2} := x \rightarrow \frac{(x - xv[1]) (x - xv[2]) (x - xv[4])}{(xv[3] - xv[1]) (xv[3] - xv[2]) (xv[3] - xv[4])} - \frac{1}{8} (x + 2) (x - 1) (x - 4)$$

$$\text{phi3} := x \rightarrow \frac{(x - xv[1]) (x - xv[2]) (x - xv[3])}{(xv[4] - xv[1]) (xv[4] - xv[2]) (xv[4] - xv[3])} - \frac{1}{36} (x + 2) (x - 1) (x - 2)$$

```
> Q := plot(phi0,-2..4,title='phi0(x)',axes=normal):
> plots[display](Q);
> interface(plotdevice=ps, plotoutput=datei1,
>           plotoptions='width=640,height=480');
> plots[display](Q); interface(plotdevice=win);
```

⇒ Lagrange-Basispolynom  $\varphi_0(x)$  auf  $[-2, 4]$

Lagrange-Basispolynome  $\varphi_k(x)$ ,  $k = 0, 1, 2, 3$ , auf  $[-2, 4]$

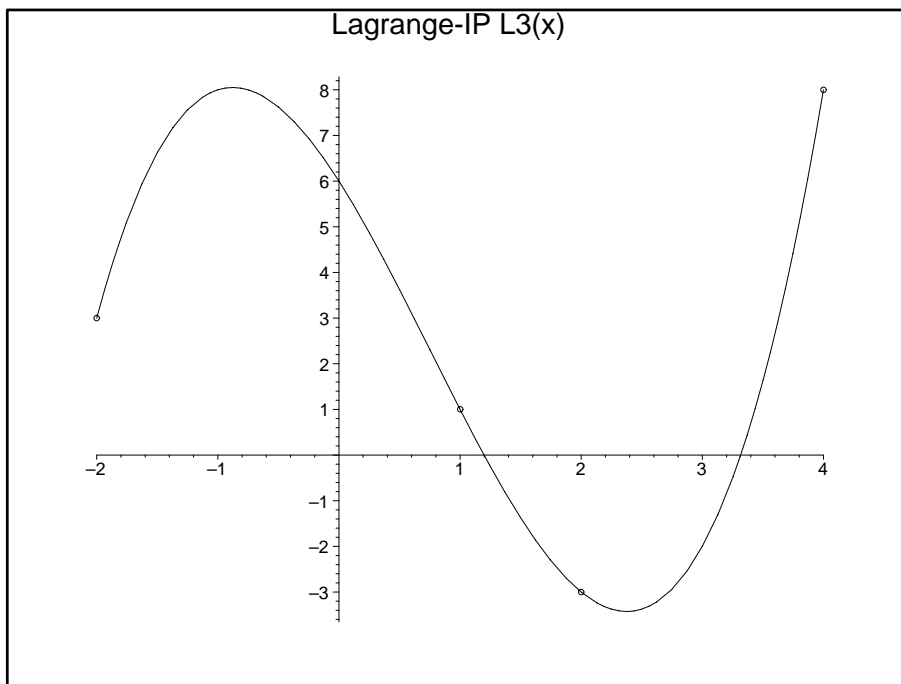


```
> a:=array(1..2,1..2):
> a[1,1]:=plot(phi0,-2..4,title='phi0(x)',color=black):
> a[1,2]:=plot(phi1,-2..4,title='phi1(x)',color=black):
> a[2,1]:=plot(phi2,-2..4,title='phi2(x)',color=black):
> a[2,2]:=plot(phi3,-2..4,title='phi3(x)',color=black):
> Q := display(a,scaling=constrained,xtickmarks=6,ytickmarks=3):
> plots[display](Q);
> interface(plotdevice=ps, plotoutput=datei2,
>           plotoptions='width=640,height=480');
> plots[display](Q);
> interface(plotdevice=win);
```

```
# Lagrange-Interpolationspolynom
> L3 := x-> yv[1]*phi0(x)+yv[2]*phi1(x)+yv[3]*phi2(x)+yv[4]*phi3(x);
> simplify(L3(x));
> pl1 := plot(L3,-2..4,ytickmarks=8,title='Lagrange-IP L3(x)');
> punkte := [seq([xv[i],yv[i]],i=1..4)]:
> pl2 := polygon(punkte,style=point,symbol=circle):
> plots[display](pl1,pl2);
> interface(plotdevice=ps,
>             plotoutput=datei3,
>             plotoptions='width=640,height=480');
> plots[display](pl1,pl2);
> interface(plotdevice=win);
```

```
L3 := x -> yv[1] phi0(x) + yv[2] phi1(x) + yv[3] phi2(x) +
          yv[4] phi3(x)
```

$$\frac{2}{3}x^3 - \frac{3}{2}x^2 - \frac{25}{6}x + 6$$



```
# Legesgue-Konstante, Kondition der Interpolation
> AbsL3 := x-> abs(phi0(x))+abs(phi1(x))+abs(phi2(x))+abs(phi3(x));
> simplify(AbsL3(x));
> seq([-0.73+i/100,AbsL3(-0.73+i/100)],i=-1..1);
> Kabs := 3.598070750;
```

```

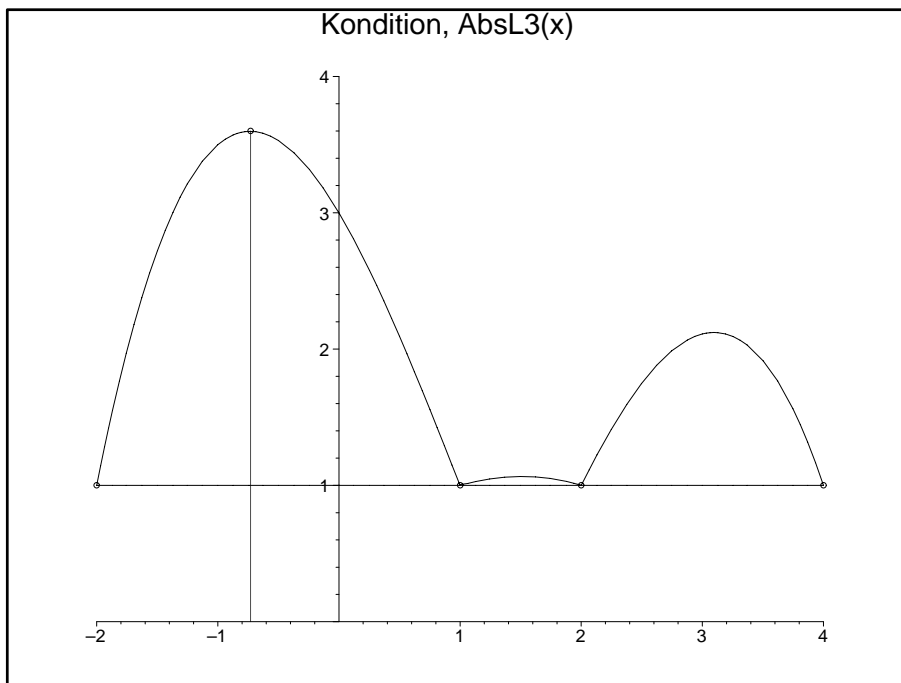
AbsL3 := x -> | phi0(x) | + | phi1(x) | + | phi2(x) | + | phi3(x) |

1/72 | (x - 1) (x - 2) (x - 4) | + 1/9 | (x + 2) (x - 2) (x - 4) |
+ 1/8 | (x + 2) (x - 1) (x - 4) | + 1/36 | (x + 2) (x - 1) (x - 2) |

[-.7400000000, 3.597994000], [-.73, 3.598070751],
[-.7200000000, 3.597888000]

```

```
Kabs := 3.598070750
```



```

> pl3 := plot([1,AbsL3],-2..4,0..4,title='Kondition, AbsL3(x)'):
> pl4 := plot([-0.73,0],[-0.73,3.6]):
> punkte := [seq([xv[i],1],i=1..4),[-0.73,3.6]]:
> pl5 := polygon(punkte,style=point,symbol=circle):
> plots[display](pl3,pl4,pl5);

> interface(plotdevice=ps,
>           plotoutput=datei4,
>           plotoptions='width=640,height=480');
> plots[display](pl3,pl4,pl5);
> interface(plotdevice=win);

```

**Beispiel 1.4 Baryzentrische Darstellung**

Dies ist eine spezielle Form des Lagrange-Interpolationspolynoms (barycenter - Gravitationszentrum) unter Verwendung der Partialbruchzerlegung.

Die Referenz mit vier Punkten sei  $\{(x_i, y_i)\}_{i=0}^3 = \{(1, 2), (2, 4), (5, 0), (6, 1)\}$ .

Das Lagrange-Interpolationspolynom berechnet man gemäß

$$\begin{aligned}
 L_3(x) &= \sum_{k=0}^3 y_k \varphi_k(x), \quad n=3 \\
 &= -\frac{1}{10}(x-2)(x-5)(x-6) + \frac{1}{3}(x-1)(x-5)(x-6) \\
 &\quad + 0(x-1)(x-2)(x-6) + \frac{1}{20}(x-1)(x-2)(x-5) \\
 &= (x-1)(x-2)(x-5)(x-6) \left[ -\frac{1}{10} \frac{1}{x-1} + \frac{1}{3} \frac{1}{x-2} + \right. \\
 &\quad \left. 0 \frac{1}{x-5} + \frac{1}{20} \frac{1}{x-6} \right] \\
 &= \frac{-\frac{1}{10} \frac{1}{x-1} + \frac{1}{3} \frac{1}{x-2} + 0 \frac{1}{x-5} + \frac{1}{20} \frac{1}{x-6}}{1} \\
 &\quad \frac{1}{(x-1)(x-2)(x-5)(x-6)} \\
 &= \frac{-\frac{1}{10} \frac{1}{x-1} + \frac{1}{3} \frac{1}{x-2} + 0 \frac{1}{x-5} + \frac{1}{20} \frac{1}{x-6}}{-\frac{1}{20} \frac{1}{x-1} + \frac{1}{12} \frac{1}{x-2} - \frac{1}{12} \frac{1}{x-5} + \frac{1}{20} \frac{1}{x-6}}
 \end{aligned}$$

Die letzte Zeile heißt baryzentrische Darstellung des Interpolationspolynoms.

Die Verwendung dieses Begriffs hängt damit zusammen, dass sich die Summendarstellung des Interpolationspolynoms  $\sum_{k=0}^n y_k \varphi_k(x)$  im baryzentrischen Kalkül, einem Hilfsmittel der analytischen Geometrie, einordnet und auch eine physikalische Interpretation dahinter steht.<sup>1</sup> Die Basisfunktionen  $\varphi_k(x)$  entsprechen den sogenannten Ortsvektoren  $\mathbf{z}_k$  und die Stützwerte  $y_k$  den baryzentrischen Koordinaten  $m_k$ , so dass man jeden Punkt  $\mathbf{P}$  eindeutig durch die reellen Koordinaten gemäß  $\mathbf{P} = \sum_{k=0}^n m_k \mathbf{z}_k$  darstellen kann.

---

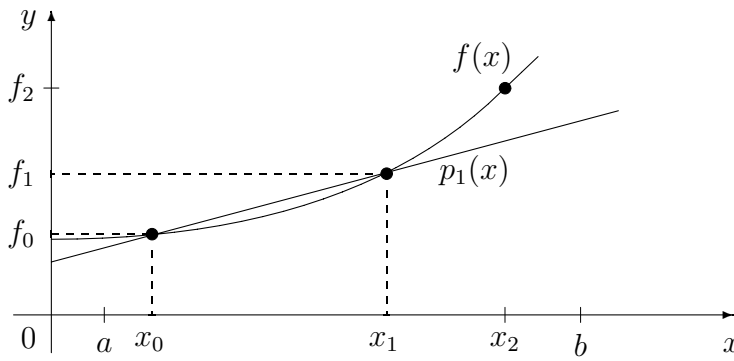
<sup>1</sup>E. Zeidler (Hrsg.): *Teubner-Taschenbuch der Mathematik*. B. G. Teubner Verlagsgesellschaft Leipzig 1996.

### 1.3.3 Newton-Interpolation

Die geometrische Veranschaulichung der linearen und quadratischen Interpolation ist die Konstruktion einer Geraden durch zwei Punkte bzw. einer quadratischen Parabel durch drei.

(1) Die lineare Interpolation basiert auf zwei Stützstellen  $x_0 < x_1$  mit  $f_i = f(x_i)$ . Die Geradengleichung hat die Gestalt

$$p_1(x) = f_0 + \frac{f_1 - f_0}{x_1 - x_0}(x - x_0). \quad (1.22)$$



**Abb. 1.1** Lineares Interpolationspolynom durch 2 Punkte  $(x_i, f_i)$ ,  $i = 0, 1$

(2) Die quadratische Interpolation braucht drei Stützstellen  $x_0 < x_1 < x_2$ . Die Gleichung der quadratischen Parabel ist

$$p_2(x) = f_0 + \frac{f_1 - f_0}{x_1 - x_0}(x - x_0) + \frac{\frac{f_2 - f_1}{x_2 - x_1} - \frac{f_1 - f_0}{x_1 - x_0}}{x_2 - x_0}(x - x_0)(x - x_1). \quad (1.23)$$

In diesen Darstellungen verwenden wir Differenzen und dividierte Differenzen (Differenzenquotienten).

Wir notieren die Differenzen für äquidistante Stützstellen  $x_i = x_{i-1} + h$ ,  $f_i = f(x_i)$ .

**Definition 1.3** Vorwärtsdifferenzen  $\Delta^k$  bzw. Rückwärtsdifferenzen  $\nabla^k$

Für die  $k$ -ten Differenzen ( $k \geq 0$ ) gelten die Beziehungen

$$\begin{aligned} \Delta^0 f(x) &= f(x), \\ \Delta^1 f(x) &= \Delta f(x) = f(x+h) - f(x), \quad h > 0, \\ \Delta^n f(x) &= \Delta(\Delta^{n-1} f(x)), \\ \nabla^0 f(x) &= f(x), \\ \nabla^1 f(x) &= \nabla f(x) = f(x) - f(x-h) = \Delta f(x-h), \\ \nabla^n f(x) &= \nabla(\nabla^{n-1} f(x)). \end{aligned} \quad (1.24)$$

Dann gilt  $\nabla^k f_i = \Delta^k f_{i-k}$ .

**Definition 1.4** Dividierte Differenzen für paarweise verschiedene  $x_i$

*Der Quotient*

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

heißt 1. dividierte Differenz von  $x_1$  und  $x_0$  bezüglich  $f$ .

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

heißt 2. dividierte Differenz von  $x_2, x_1, x_0$  bezüglich  $f$ .

Sei die  $k$ -te dividierte Differenz bereits definiert. Dann ist

$$f[x_{k+1}, x_k, \dots, x_1, x_0] = \frac{f[x_{k+1}, \dots, x_2, x_1] - f[x_k, \dots, x_1, x_0]}{x_{k+1} - x_0} \quad (1.25)$$

die  $(k+1)$ -te dividierte Differenz von  $x_{k+1}, \dots, x_1, x_0$  bezüglich  $f$ .

Dividierte Differenzen sind symmetrisch in Bezug auf die Argumente.

Sie werden auch **Steigungen** genannt. Es gilt neben (1.25) ebenfalls

$$f[x_{k+1}, x_k, \dots, x_1, x_0] = \frac{f[x_{k+1}, x_{k-1}, \dots, x_2, x_1, x_0] - f[x_k, \dots, x_1, x_0]}{x_{k+1} - x_k}. \quad (1.26)$$

Andere Schreibweisen sind

$$f[x_{k+1}, \dots, x_1, x_0] = [x_{k+1}, \dots, x_1, x_0]f = [x_{k+1}, \dots, x_1, x_0]f = [x_{k+1}, \dots, x_1, x_0] = \delta^{k+1}f_0.$$

Um die dividierten Differenzen  $f[x_0], f[x_1, x_0], \dots, f[x_{k+1}, \dots, x_1, x_0]$  systematisch und geschickt zu berechnen, bietet sich das folgende Schema an.

**Schema der dividierten Differenzen ( $n = 3$ )**

			$x_0$	$f_0$			
			$x_1 - x_0$	$f[x_1, x_0]$			
		$x_2 - x_0$	$x_1$	$f_1$	$f[x_2, x_1, x_0]$		
		$x_3 - x_0$	$x_2 - x_1$	$f[x_2, x_1]$	$f[x_3, x_2, x_1, x_0]$		
		$x_3 - x_1$	$x_2$	$f_2$	$f[x_3, x_2, x_1]$		
		$x_3 - x_2$	$x_3$	$f[x_3, x_2]$			
			$x_3$	$f_3$			

**Folgerung 1.5** Für das lineare und quadratische Interpolationspolynom erhält man mit den dividierten Differenzen die Darstellungen

$$\begin{aligned} p_1(x) &= f(x_0) + f[x_1, x_0](x - x_0), \\ p_2(x) &= f(x_0) + f[x_1, x_0](x - x_0) + f[x_2, x_1, x_0](x - x_0)(x - x_1). \end{aligned} \quad (1.27)$$



**Satz 1.6** Newton-Interpolationspolynom(1) Das Polynom höchstens  $n$ -ten Grades

$$\begin{aligned}
N_n(x) &= f(x_0) + \sum_{j=1}^n f[x_j, x_{j-1}, \dots, x_0](x - x_0)(x - x_1) \cdot \dots \cdot (x - x_{j-1}) \\
&= \sum_{k=0}^n \delta^k f_0 \omega_k(x), \text{ wobei } \omega_k(x) = \prod_{i=0}^{k-1} (x - x_i),
\end{aligned} \tag{1.28}$$

heißt Newton-Interpolationspolynom.

 $\omega_k(x) = \prod_{i=0}^{k-1} (x - x_i)$  sind die Newton-Basispolynome, Knotenpunktpolynome oder Newton-Terme,  $\delta^k f_0 = f[x_k, \dots, x_1, x_0]$  die  $k$ -te dividierte Differenz,  $\delta^0 f_i = f_i$ .
(2)  $N_n(x)$  erfüllt die Interpolationsforderungen (1.11) und stellt das eindeutig bestimmte Interpolationspolynom der Interpolationsaufgabe (IA1) dar.

(3) Der Interpolationsfehler (Restglied der Interpolation) lautet

$$\begin{aligned}
R_n(x) &= f(x) - N_n(x) \\
&= f[x, x_n, x_{n-1}, \dots, x_1, x_0](x - x_0)(x - x_1) \cdot \dots \cdot (x - x_n) \\
&= \frac{f^{(n+1)}(\xi)}{(n+1)!} (x - x_0)(x - x_1) \cdot \dots \cdot (x - x_n),
\end{aligned} \tag{1.29}$$

mit  $\xi \in \text{int}(x_0, x_1, \dots, x_n)$ , vorausgesetzt  $f \in \mathcal{C}^{n+1}[a, b]$ . Natürlich gilt  $R_n(x_i) = 0$ .**Folgerung 1.7** Die Fehlerabschätzung für das Interpolationspolynom im Intervall der Stützstellen ist

$$|R_n(x)| = |f(x) - N_n(x)| \leq \frac{\max_{x \in [a, b]} |f^{(n+1)}(x)|}{(n+1)!} |(x - x_0)(x - x_1) \cdot \dots \cdot (x - x_n)|. \tag{1.30}$$

**Beispiel 1.5** Newton-InterpolationspolynomAus der Referenz mit 4 Punkten ( $n = 3$ )

$x_i$	1	3	4	6
$y_i$	4	6	4	12

bildet man das Schema der dividierten Differenzen

			1		4		
			2		1		
		3	3		6	-1	
5		1			-2		$\frac{3}{5}$
	3		4		4	2	
		2			4		
			6		12		

Daraus ergibt sich die Interpolationsformel

$$\begin{aligned}
 N_3(x) &= f(x_0) + f[x_1, x_0](x - x_0) \\
 &\quad + f[x_2, x_1, x_0](x - x_0)(x - x_1) + f[x_3, x_2, x_1, x_0](x - x_0)(x - x_1)(x - x_2) \\
 &= 4 + 1(x - 1) - 1(x - 1)(x - 3) + \frac{3}{5}(x - 1)(x - 3)(x - 4) \quad (\text{obere Zeile}) \\
 &= 12 + 4(x - 6) + 2(x - 6)(x - 4) + \frac{3}{5}(x - 6)(x - 4)(x - 3) \quad (\text{untere Zeile}) \\
 &= \frac{3}{5}x^3 - \frac{29}{5}x^2 + \frac{82}{5}x - \frac{36}{5} \quad (\text{mit inversem Horner-Schema}).
 \end{aligned}$$

### Beispiel 1.6 Genauigkeit der Interpolationsformel

Gegeben seien die Logarithmuswerte  $\ln(100)$ ,  $\ln(101)$ ,  $\ln(102)$ ,  $\ln(103)$ .

Mit welcher Genauigkeit kann der Wert  $\ln(100.5)$  mit dem Newton-Interpolationspolynom berechnet werden?

Die Problemparameter sind  $n = 3$ ,  $a = 100$ ,  $b = 103$  und von der Funktion  $f(x) = \ln(x)$ ,  $f \in \mathcal{C}^\infty(\mathbb{R}_+)$  benötigt man die 4. Ableitung  $f^{(4)}(x) = -6/x^4$ . Es gilt

$$\begin{aligned}
 |R_3(100.5)| &= |f(100.5) - N_3(100.5)| \\
 &\leq \frac{\max_{[100,103]} |f^{(4)}(x)|}{4!} |(+0.5)(-0.5)(-1.5)(-2.5)| \\
 &\leq \frac{\max_{[100,103]} 6/x^4}{4!} \cdot 0.5 \cdot 0.5 \cdot 1.5 \cdot 2.5 \\
 &= 2.344 \cdot 10^{-9}.
 \end{aligned}$$

Die Genauigkeit der Berechnung ist bis auf 8 Nachkommastellen möglich.

Für eine einfache Berechnung von Polynomwerten definiert man  $c_0 = f(x_0) = f[x_0]$  und  $c_j = f[x_j, x_{j-1}, \dots, x_1, x_0]$ ,  $j \geq 1$ . Dann erhält man

$$\begin{aligned}
 N_n(x) &= c_0 + \sum_{j=1}^n c_j \prod_{i=0}^{j-1} (x - x_i) \\
 &= c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \dots + c_n(x - x_0)(x - x_1) \cdot \dots \cdot (x - x_{n-1}) \\
 &= c_0 + (x - x_0)\{c_1 + (x - x_1)[c_2 + \dots + (x - x_{n-2})(c_{n-1} + (x - x_{n-1})c_n)\dots]\}.
 \end{aligned}$$

### Algorithmische Darstellung der Polynomwertberechnung

```

Q := c[n];
for j := n - 1 downto 0 do Q := Q * (x - x[j]) + c[j];
Polynomwert := Q;

```

Später werden die effizienteren Schemata zur Polynomwertberechnung von Aitken und Aitken-Neville vorgestellt.

**Nachweis der allgemeinen Form des Newton-Interpolationspolynoms**

Für die Interpolationspolynome vom Grad 1 und 2 haben wir in der Folgerung 1.5 die Darstellung (1.28) schon gezeigt. Zum allgemeinen Nachweis verwenden wir rekursive Beziehungen aus den folgenden beiden Sätzen.

**Satz 1.8** *Das Newton-Interpolationspolynom  $N_n(x)$  aus (1.28) kann man rekursiv berechnen gemäß*

$$\begin{aligned} N_0(x) &= f_0, \\ N_k(x) &= N_{k-1}(x) + \frac{f_k - N_{k-1}(x_k)}{\omega_k(x_k)} \omega_k(x), \quad k = 1, 2, \dots, n. \end{aligned} \quad (1.31)$$

**Beweis.** Anwendung der vollständigen Induktion.

Der Induktionsanfang für  $k = 1$  ist mit der Geradengleichung

$$N_1(x) = N_0(x) + \frac{f_1 - N_0(x_0)}{\omega_1(x_1)} \omega_1(x) = f_0 + \frac{f_1 - f_0}{x_1 - x_0} (x - x_0)$$

schon durchgeführt worden.

Der Induktionsschritt wird bez. des Grades des Polynoms gemacht.

Sei  $N_{k-1}(x)$  das Interpolationspolynom durch die Punkte  $(x_j, f_j)$ ,  $j = 0, 1, \dots, k-1$ . Wegen  $N_{k-1}(x_j) = f_j$  sowie  $\omega_k(x)$  als Polynom  $k$ -ten Grades mit  $\omega_k(x_j) = 0$ ,  $j = 0, 1, \dots, k-1$ , ist  $N_k(x)$  das eindeutig bestimmte Polynom  $k$ -ten Grades, das an den Stellen  $x_j$ ,  $j = 0, 1, \dots, k-1, k$ , die Funktionswerte  $f_j$  hat. Letzteres ist durch Ausrechnen der Werte  $N_k(x_j)$  einfach zu überprüfen.  $\square$

**Satz 1.9** *Wenn  $p_{n_1, n_2, \dots, n_k}(x)$  das Interpolationspolynom zu den (nicht notwendig äquidistanten) Stützstellen  $x_{n_1}, x_{n_2}, \dots, x_{n_k}$  ist, dann gilt mit  $p_i(x) = f_i$  die Beziehung*

$$p_{0,1,\dots,n}(x) = \frac{1}{x_n - x_{n-1}} \begin{vmatrix} p_{0,1,\dots,n-2,n-1}(x) & x_{n-1} - x \\ p_{0,1,\dots,n-2,n}(x) & x_n - x \end{vmatrix}. \quad (1.32)$$

Die Formel (1.32) wird auch als **Aitken-Interpolation** bezeichnet.

**Beweis.** (Skizze)

Wir zeigen nur den Induktionsschritt bez. der Anzahl der Indizes des Polynoms.

Das Polynom  $p_{0,1,\dots,n-2,n-1}(x)$ , Grad  $\leq n-1$ , verläuft durch die Punkte  $(x_j, f_j)$ ,  $j = 0, 1, \dots, n-1$ .

Ähnlich geht das Polynom  $p_{0,1,\dots,n-2,n}(x)$ , auch vom Grad  $\leq n-1$ , durch die Punkte  $(x_j, f_j)$ ,  $j = 0, 1, \dots, n-2, n$ .

Die Frage ist nun, ob das Polynom  $p_{0,1,\dots,n}(x)$ , das vom Grad  $\leq n$  ist, durch die Punkte  $(x_j, f_j)$ ,  $j = 0, 1, \dots, n$ , geht?

Das Nachrechnen für die Argumente  $x_0, x_1, \dots, x_{n-1}, x_n$  ergibt

$$\begin{aligned} p_{0,1,\dots,n}(x_n) &= p_{0,1,\dots,n-2,n}(x_n) = f_n, \\ p_{0,1,\dots,n}(x_{n-1}) &= p_{0,1,\dots,n-2,n-1}(x_{n-1}) = f_{n-1}, \\ p_{0,1,\dots,n}(x_k) &= \frac{1}{x_n - x_{n-1}} [(x_n - x_k)p_{0,1,\dots,n-2,n-1}(x_k) \\ &\quad - (x_{n-1} - x_k)p_{0,1,\dots,n-2,n}(x_k)], \quad k \neq n-1, n \\ &= \frac{1}{x_n - x_{n-1}} [(x_n - x_k)f_k - (x_{n-1} - x_k)f_k] \\ &= f_k. \end{aligned}$$

$p_{0,1,\dots,n}(x)$  als ein Polynom höchstens  $n$ -ten Grades zur Referenz  $\{(x_j, f_j), j = 0(1)n\}$  ist somit eindeutig bestimmt.  $\square$

Damit kann man nunmehr die Newton-Interpolationsformel (1.28) wiederum mittels vollständiger Induktion zeigen.

Den Induktionsanfang haben wir schon überprüft. Wenn das Interpolationspolynom  $p_{0,1,\dots,n-1}(x)$  die Gestalt

$$f_0 + f[x_1, x_0](x - x_0) + \dots + f[x_{n-1}, \dots, x_1, x_0](x - x_0)(x - x_1) \cdot \dots \cdot (x - x_{n-2})$$

hat, besitzt diese auch das um einen Grad höhere Interpolationspolynom  $p_{0,1,\dots,n}(x)$ .

**Beweis.** Satz 6.6 Teil (1) und (2) mit Durchführung des Induktionsschritts.

Gemäß Induktionsvoraussetzung sind

$$N_k(x) = f_0 + f[x_1, x_0](x - x_0) + \dots + f[x_k, \dots, x_1, x_0](x - x_0)(x - x_1) \cdot \dots \cdot (x - x_{k-1}),$$

$k = 0, 1, \dots, n-1$ , jeweils die Interpolationspolynome zu den aufeinander folgenden Referenzen  $\{(x_j, f_j), j = 0, 1, \dots, k\}$ .

Wir formen die rekursive Formel (1.32) um und zeigen die Übereinstimmung mit der Darstellung (1.28).

$$\begin{aligned} p_{0,1,\dots,n}(x) &= \\ &= \frac{1}{x_n - x_{n-1}} \{p_{0,1,\dots,n-1}(x)(x_n - x) - p_{0,1,\dots,n-2,n}(x)(x_{n-1} - x)\} \\ &= \frac{1}{x_n - x_{n-1}} \{[N_{n-2}(x) + f[x_{n-1}, x_{n-2}, \dots, x_1, x_0](x - x_0) \cdot \dots \cdot (x - x_{n-2})](x_n - x) \\ &\quad - [N_{n-2}(x) + f[x_n, x_{n-2}, \dots, x_1, x_0](x - x_0) \cdot \dots \cdot (x - x_{n-2})](x_{n-1} - x)\} \end{aligned}$$

$$\begin{aligned}
p_{0,1,\dots,n}(x) &= \\
&= N_{n-2}(x) + \frac{1}{x_{n-1} - x_n} \{f[x_{n-1}, x_{n-2}, \dots, x_1, x_0](x - x_0) \cdot \dots \cdot (x - x_{n-2})(x - x_n) \\
&\quad - f[x_n, x_{n-2}, \dots, x_1, x_0](x - x_0) \cdot \dots \cdot (x - x_{n-2})(x - x_{n-1})\} \\
&= N_{n-2}(x) + \frac{1}{x_{n-1} - x_n} \cdot \\
&\quad \{f[x_{n-1}, x_{n-2}, \dots, x_1, x_0](x - x_0) \cdot \dots \cdot (x - x_{n-2})(x - x_{n-1} + x_{n-1} - x_n) \\
&\quad - f[x_n, x_{n-2}, \dots, x_1, x_0](x - x_0) \cdot \dots \cdot (x - x_{n-2})(x - x_{n-1})\} \\
&= N_{n-2}(x) + f[x_{n-1}, x_{n-2}, \dots, x_1, x_0](x - x_0) \cdot \dots \cdot (x - x_{n-2}) + \\
&\quad \frac{1}{x_{n-1} - x_n} \{f[x_{n-1}, x_{n-2}, \dots, x_1, x_0](x - x_0) \cdot \dots \cdot (x - x_{n-2})(x - x_{n-1}) \\
&\quad - f[x_n, x_{n-2}, \dots, x_1, x_0](x - x_0) \cdot \dots \cdot (x - x_{n-2})(x - x_{n-1})\} \\
&= N_{n-1}(x) + \frac{f[x_{n-1}, x_{n-2}, \dots, x_1, x_0] - f[x_n, x_{n-2}, \dots, x_1, x_0]}{x_{n-1} - x_n} \cdot \\
&\quad (x - x_0) \cdot \dots \cdot (x - x_{n-2})(x - x_{n-1}) \\
&= N_{n-1}(x) + f[x_n, x_{n-1}, x_{n-2}, \dots, x_1, x_0](x - x_0) \cdot \dots \cdot (x - x_{n-2})(x - x_{n-1}) \\
&= N_n(x). \quad \square
\end{aligned}$$

### Interpolationspolynom von Newton-Gregory

Im Fall äquidistanter Stützstellen  $x_i = x_0 + ih$ ,  $i = 0, 1, \dots, n$ , erhält man die Formel von Newton-Gregory.

Für die Newton-Interpolation gelten dann die Beziehungen

$$\begin{aligned}
\omega_k(x) &= \prod_{i=0}^{k-1} (x - x_i) = h^k \prod_{i=0}^{k-1} (s - i), \quad x = x_0 + sh, \quad \omega_0(x) = 1, \\
c_k &= \delta^k f_0 = f[x_0, x_1, \dots, x_k] = \frac{\Delta^k f_0}{k! h^k}, \\
N_n(x) &= \sum_{k=0}^n c_k \omega_k(x), \quad x = x_0 + sh \\
&= \sum_{k=0}^n \binom{s}{k} \Delta^k f_0.
\end{aligned}$$

Abschließend notieren wir noch eine Beziehung zwischen den Basispolynomen von Lagrange und Newton, die sich einfach nachrechnen lässt.

**Satz 1.10** *Gegeben seien an den Stützstellen  $x_i$ ,  $i = 0, 1, \dots, n$ , die Basispolynome von Lagrange*

$$\varphi_k(x) = \prod_{\substack{i=0 \\ i \neq k}}^n \frac{x - x_i}{x_k - x_i}, \quad \text{Grad}(\varphi_k(x)) = n,$$

und von Newton

$$\omega_{n+1}(x) = \prod_{i=0}^n (x - x_i) = x^{n+1} - x^n \sum_{i=0}^n x_i + \dots + (-1)^{n+1} \prod_{i=0}^n x_i.$$

Dann gilt die Beziehung

$$\varphi_k(x) = \frac{\omega_{n+1}(x)}{\omega'_{n+1}(x_k)(x - x_k)}. \quad (1.33)$$

### Berechnung von Darstellungen von Interpolationspolynomen in Maple

Für die Ermittlung der Normalform des Interpolationspolynoms kann die Maple-Funktion `interp(ststellen, stwerte, variable)` verwendet werden. Sie wird auch in der folgenden nutzerdefinierten Prozedur `interpol` als sonstige Variante (*default-Zweig*) einbezogen.

```
# nm6ip2.mws
#
# Prozedur zur Erzeugung des Interpolationspolynoms in verschiedenen
# Darstellungen:
# Newtonsche, Lagrangesche, baryzentrische Darstellung
#
> restart:
#
> interpol:=proc(stst::list, stw::list, var::name, typ::string)

> description 'Polynominterpolation in verschiedenen Darstellungen';

>   local i,k,n,omega,rho,ityp,x;

>   n:=nops(stst);
>   if nargs=3 then ityp:="Normal" else ityp:=typ; fi;
```

```

>   if ityp="Lagrange" then
>     omega:=[1];
>     for k from 1 to n-1 do
>       for i from 1 to nops(omega) do
>         omega:=subsop(i=omega[i]/(stst[i]-stst[k+1]),omega);
>       od;
>       omega:=[op(omega),-sum(omega[j],j=1..nops(omega))];
>     od;
>     i:='i': k:='k':
> [sum(omega[i]/(var-stst[i])*stw[i],i=1..n)/
>  sum(omega[i]/(var-stst[i]),i=1..n),
> sum(stw[i]*omega[i]*product(var-stst[k],k=1..i-1)*
>  product(var-stst[k],k=i+1..n),i=1..n)];
>   elif
>     ityp="Newton" then
>     omega:=stw;
>     for k from 0 to n-1 do rho:=stw[k+1];
>       for i from 0 to k-1 do
>         rho:=(rho-omega[i+1])/(stst[k+1]-stst[i+1]);
>       od;
>       omega:=subsop(k+1=rho,omega);
>     od;
>     i:='i': k:='k':
>     sum(omega[k]*product(var-stst[i],i=1..k-1),k=1..n);
>   else
>     interp(stst,stw,var);
> fi;
> end:
#
# Aufruf:
# Anzahl der Parameter: 3 oder 4
#
# Bedeutung:
# 1. Parameter: Stuetzstellen (Liste)
# 2. Parameter: Stuetzwerte (Liste)
# 3. Parameter: Name der zu verwendenden Variablen
# 4. Parameter: Newton oder Lagrange; wenn dieser Parameter fehlt, wird
# Maples interne Interpolationsroutine interp(stst,stw,var) verwendet.
#
# Besonderheiten:
# Das Lagrangesche Interpolationspolynom kann auf zwei verschiedene
# Arten ausgegeben werden; dies wird durch einen zusaetzlichen Index [1]
# bzw. [2] erreicht.

```

```

# Beispiel:
#
# Referenz
> n := 3;
> xv := [1, 2, 5, 6];
> yv := [2, 4, 1, 1];

      n := 3
      xv := [1, 2, 5, 6]
      yv := [2, 4, 1, 1]

#
> interpol(xv,yv,x,"Newton");
      2 x - 3/4 (x - 1) (x - 2) + 1/5 (x - 1) (x - 2) (x - 5)

#
> N3 := 2+2*(x-1)-3/4*(x-1)*(x-2)+1/5*(x-1)*(x-2)*(x-5);
      N3 := 2 x - 3/4 (x - 1) (x - 2) + 1/5 (x - 1) (x - 2) (x - 5)

#
> interpol(xv,yv,x,"Lagrange")[2]; # die konventionelle Darstellung
      - 1/10 (x - 2) (x - 5) (x - 6) + 1/3 (x - 1) (x - 5) (x - 6)
      - 1/12 (x - 1) (x - 2) (x - 6) + 1/20 (x - 1) (x - 2) (x - 5)

#
> interpol(xv,yv,x,"Lagrange")[1]; # die baryzentrische Darstellung
      1          1          1          1
      - 1/10 ----- + 1/3 ----- - 1/12 ----- + 1/20 -----
      x - 1      x - 2      x - 5      x - 6
      -----
      1          1          1          1
      - 1/20 ----- + 1/12 ----- - 1/12 ----- + 1/20 -----
      x - 1      x - 2      x - 5      x - 6

#
> interpol(xv,yv,x);
      3   47   2   153
      1/5 x - -- x + --- x - 7/2
      20      20

> interp(xv,yv,x);
      3   47   2   153
      1/5 x - -- x + --- x - 7/2
      20      20

```



### 1.3.4 Polynomwertberechnung

An Stelle der Berechnung des Polynomwertes  $p(x)$  an einer Zwischenstelle über das Newton-Interpolationspolynom oder mittels seiner Normalform und Horner-Schema kann dies geschickter und schneller mittels der Aitken-Interpolation erfolgen.

Dazu nimmt man das Schema von Aitken bei gegebener Referenz.

Eine verbesserte Version ist das Schema von Aitken-Neville.

#### (1) Wiederholung

Wenn  $p_{n_1, n_2, \dots, n_k}(x)$  das Interpolationspolynom zu den Stützstellen  $x_{n_1}, x_{n_2}, \dots, x_{n_k}$  ist, dann gilt mit  $p_i(x) = y_i$  die Beziehung

$$p_{0,1,\dots,n}(x) = \frac{1}{x_n - x_{n-1}} \begin{vmatrix} p_{0,1,\dots,n-2,n-1}(x) & x_{n-1} - x \\ p_{0,1,\dots,n-2,n}(x) & x_n - x \end{vmatrix}. \quad (1.34)$$

Bisher haben wir uns für die Koeffizienten des Interpolationspolynoms  $n$ -ten Grades interessiert. Andererseits kann diese rekursive Beziehung natürlich auch zur numerischen Berechnung des Polynomwertes für ein spezielles Argument verwendet werden.

#### (2) Aitken-Schema

Gegeben ist  $(x_i, y_i)$ ,  $i = 0, 1, \dots, n$ , und ein Zwischenwert  $x$ .

Weiter sei  $p_i(x) = y_i$ ,  $i = 0, 1, \dots, n$ ,  $p(x) = p_{0,1,\dots,n}(x) \equiv p_{01\dots n}(x)$ .

Im folgenden Schema sind die ersten drei Spalten durch die vorgegebenen Größen der Referenz und des Arguments definiert. Jetzt berechnen wir mittels (1.34) die weiteren Spalten.

$x_i$	$x_i - x$	$p_i(x) = y_i$ $k = 0$	$k = 1$	$k = 2$	.	.	$k = n$
$x_0$	$x_0 - x$	$p_0(x) = y_0$					
$x_1$	$x_1 - x$	$p_1(x) = y_1$	$p_{01}(x)$				
$x_2$	$x_2 - x$	$p_2(x) = y_2$	$p_{02}(x)$	$p_{012}(x)$			
.	.	.	.	.	.		
.	.	.	.	.	.	.	
$x_n$	$x_n - x$	$p_n(x) = y_n$	$p_{0n}(x)$	$p_{01n}(x)$	.	.	$p_{01\dots n}(x) = p(x)$

Der rechte untere Wert ist der gesuchte Funktionswert des Polynoms bei  $x$ .

#### (3) Schema von Aitken-Neville

In der vierten Spalte des Aitken-Schemas mit den Werten  $p_{0k}(x)$ ,  $k = 1, 2, \dots, n$ , setzen wir nun die Größen  $p_{01}(x), p_{12}(x), p_{23}(x), \dots$  ein. Dazu wählt man noch eine günstigere Darstellung mit Doppelindizierung wie bei Matrizen.

Anfangsbelegung und ein zeilenweise Berechnung im Aitken-Neville-Schema führen somit auf die Formeln

$$\begin{aligned}
p_{i0} &= y_i, \quad i = 0, 1, \dots, n, \\
p_{ik} &= \frac{1}{x_{i-k} - x_i} \begin{vmatrix} p_{i,k-1} & x_i - x \\ p_{i-1,k-1} & x_{i-k} - x \end{vmatrix} \\
&= p_{i,k-1} - \frac{x_i - x}{x_i - x_{i-k}} (p_{i,k-1} - p_{i-1,k-1}), \quad 1 \leq k \leq i, \\
p_{nn} &= p(x),
\end{aligned} \tag{1.35}$$

und ihre schematische Darstellung

$x_i$	$x_i - x$	$p_{i0} = y_i$ $k = 0$	$k = 1$	$k = 2$	.	.	$k = n$
$x_0$	$x_0 - x$	$p_{00} = y_0$					
$x_1$	$x_1 - x$	$p_{10} = y_1$	$p_{11}$				
$x_2$	$x_2 - x$	$p_{20} = y_2$	$p_{21}$	$p_{22}$			
.	.	.	.	.	.		
.	.	.	.	.	.	.	
$x_n$	$x_n - x$	$p_{n0} = y_n$	$p_{n1}$	$p_{n2}$	.	.	$p_{nn} = p(x)$

**Beispiel 1.7** Schemata zur Polynomwertberechnung mit der Aitken-Interpolation  
Gegeben seien das Argument  $x = 3$  und die Referenz mit 4 Punkten ( $n = 3$ )

$x_i$	0	1	2	5
$y_i$	0	-1	4	115

Man kann beide Schemata auf der linken Seite noch um die Differenzen der Stützstellen ergänzen.

Schema von Aitken mit  $N_3(3) = 21$

$x_i - x_{i-k}$	$x_i$	$x_i - x$	$y_i$	$k = 0$	$k = 1$	$k = 2$	$k = 3$	
	0	-3	0					
1	1	-2	-1	-3				
2	1	2	4	6	15			
5	4	3	5	2	115	69	33	21

Schema von Aitken-Neville mit dem gleichen Polynomwert

$x_{i-k} - x_i$	$x_i$	$x_i - x$	$y_i$ $k = 0$	$k = 1$	$k = 2$	$k = 3$
	0	-3	0			
-1	1	-2	-1	-3		
-2 -1	2	-1	4	9	15	
-5 -4 -3	5	2	115	41	25	21

## 1.4 Hermite-Interpolation

Dabei handelt es sich um eine **verallgemeinerte Polynominterpolation** mit der Einbeziehung von Funktions- und Ableitungswerten.

- Grundintervall  $I = [a, b] \subset \mathbb{R}$  mit  $-\infty < a < b < \infty$  und (bekannte oder unbekannte) reelle Funktion  $f \in \mathcal{C}^\alpha(I)$ ,  $\alpha \geq 0$ .
- Referenz  $R$  gemäß (1.9) mit den Stützstellen  $x_i$  und Stützwerten  $y_i$  sowie weiteren Größen in Form von Ableitungswerten.
- Das Interpolationspolynom vom Grad  $\leq m$  entspricht dem Ansatz

$$p_m(x) = a_0 x^m + a_1 x^{m-1} + \dots + a_{m-1} x + a_m, \quad a_i \in \mathbb{R}. \quad (1.36)$$

- Die Interpolationsforderung ist das System von Bedingungen

$$\begin{aligned} p_m(x_0) &= f(x_0), \quad p'_m(x_0) = f'(x_0), \quad \dots, \quad p_m^{(\alpha_0-1)}(x_0) = f^{(\alpha_0-1)}(x_0), \\ p_m(x_1) &= f(x_1), \quad p'_m(x_1) = f'(x_1), \quad \dots, \quad p_m^{(\alpha_1-1)}(x_1) = f^{(\alpha_1-1)}(x_1), \\ &\dots \\ p_m(x_n) &= f(x_n), \quad p'_m(x_n) = f'(x_n), \quad \dots, \quad p_m^{(\alpha_n-1)}(x_n) = f^{(\alpha_n-1)}(x_n). \end{aligned} \quad (1.37)$$

Dabei sind  $\alpha_i \geq 1$  Vielfachheiten und

$$m = -1 + \sum_{i=0}^n \alpha_i, \quad \alpha = -1 + \max_{i=0(1)n} \alpha_i.$$

Somit stimmt die Anzahl der unbekannten Koeffizienten im Ansatz (1.36) mit der Anzahl der Bedingungen überein.

- **Interpolationsaufgabe (IA2)**

Gesucht sind die Koeffizienten  $a_0, a_1, \dots, a_m$  in (1.36), so dass die Interpolationsforderung (1.37) erfüllt ist.

Mit der Vorgabe von Ableitungswerten werden diese als Steigungen im Schema der dividierten Differenzen berücksichtigt. Das führt dann dort auf Initialisierungsanteile und die entsprechenden Stützstellen müssen dann mehrfach aufgelistet werden. Deshalb ist die Definition der dividierten Differenzen zu verallgemeinern.

**Definition 1.5** Dividierte Differenzen mit mehrfachen Argumenten

Sei  $f$  hinreichend oft stetig differenzierbar auf  $I$ . Dann gilt

$$\begin{aligned} f[\underbrace{x_0, \dots, x_0}_{k_0\text{-mal}}, \underbrace{x_1, \dots, x_1}_{k_1\text{-mal}}, \dots, \underbrace{x_n, \dots, x_n}_{k_n\text{-mal}}] &= \\ &= \lim_{x_i^{(j)} \rightarrow x_i} f[x_0, x_0^{(1)}, \dots, x_0^{(k_0-1)}, x_1, x_1^{(1)}, \dots, x_1^{(k_1-1)}, \dots, x_n, x_n^{(1)}, \dots, x_n^{(k_n-1)}], \end{aligned} \quad (1.38)$$

wobei alle  $x_i^{(j)}$  voneinander verschieden sind.

**Satz 1.11** Sei  $f(x)$   $k$ -mal stetig differenzierbar ( $k \geq 1$ ) in einer Umgebung von  $x$ . Dann gilt

$$f[\underbrace{x, x, \dots, x}_{(k+1)\text{-mal}}] = \frac{f^{(k)}(x)}{k!}.$$

**Hermite-Interpolationspolynom**

- Die Ausgangsreferenz  $R$  der Stützstellen wird entsprechend dem Auftreten von Ableitungswerten vergrößert. Somit sind die hier benutzten Werte  $x_j$  die entsprechende Erweiterung verbunden mit einer Neunummerierung.
- Das Polynom höchstens  $m$ -ten Grades

$$H_m(x) = f(x_0) + \sum_{j=1}^m f[x_j, x_{j-1}, \dots, x_0](x-x_0)(x-x_1) \cdot \dots \cdot (x-x_{j-1}) \quad (1.39)$$

heißt Hermite-Interpolationspolynom.

$H_m(x)$  löst die Interpolationsaufgabe (IA2) eindeutig.

- Der **Interpolationsfehler** (Restglied der Interpolation) lautet

$$R_m(x) = f(x) - H_m(x) = \frac{f^{(m+1)}(\xi)}{(m+1)!}(x-x_0)(x-x_1) \cdot \dots \cdot (x-x_m) \quad (1.40)$$

mit  $\xi \in \text{int}(x_0, x_1, \dots, x_m)$ , vorausgesetzt  $f \in \mathcal{C}^{m+1}(I)$ .

**1.4.1 Bestimmung des Hermite-Interpolationspolynoms**

Wir zeigen ein Beispiel. Gegeben sei die Ausgangsreferenz

$$\begin{array}{c|l} x_0 & f(x_0), \quad f'(x_0), \quad f''(x_0) \\ x_1 & f(x_1), \quad f'(x_1) \end{array}$$

Die Vielfachheiten sind  $\alpha_0 = 3$ ,  $\alpha_1 = 2$ , somit ist  $m = 4$ .

Damit ermitteln wir in mehreren Schritten  $H_4(x)$ .

Entsprechend den Vielfachheiten ist das Ausgangsschema der dividierten Differenzen so zu ergänzen, dass die Paare  $(x_i, f_i)$   $\alpha_i$ -mal aufzunehmen sind und dazu die entsprechenden Steigungen

$$f[x_i, x_i] = f'(x_i), \quad f[x_i, x_i, x_i] = \frac{f''(x_i)}{2!}, \dots, \quad f[\underbrace{x_i, \dots, x_i}_{\alpha_i\text{-mal}}] = \frac{f^{(\alpha_i-1)}(x_i)}{(\alpha_i - 1)!}. \quad (1.41)$$

### (1) Ausgangsschema der dividierten Differenzen

Die Initialisierung heißt

$$\begin{array}{rcccl} & & x_0 & | & f_0 \\ & -- & & & f[x_0, x_0] \\ -- & & x_0 & | & f_0 \quad f[x_0, x_0, x_0] \\ & -- & & & f[x_0, x_0] \\ & & x_0 & | & f_0 \\ & & x_1 & | & f_1 \\ -- & & & & f[x_1, x_1] \\ & & x_1 & | & f_1 \end{array}$$

mit den speziellen dividierten Differenzen

$$f[x_0, x_0] = \frac{f'(x_0)}{1!}, \quad f[x_1, x_1] = \frac{f'(x_1)}{1!}, \quad f[x_0, x_0, x_0] = \frac{f''(x_0)}{2!}.$$

Die erweiterte Stützstellenfolge ist  $x_j$ ,  $j = 0, 1, \dots, 4$  (bei gleicher Bezeichnung der Größen).

### (2) Vervollständigtes Schema der dividierten Differenzen

$$\begin{array}{rcccccl} & & & & x_0 & | & f_0 \\ & & & & & & f[x_0, x_0] \\ & & -- & & & & \\ & & & -- & x_0 & | & f_0 \quad f[x_0, x_0, x_0] \\ & & & & & & f[x_0, x_0] \\ x_1 - x_0 & & & -- & & & f[x_1, x_0, x_0, x_0] \\ x_1 - x_0 & & x_1 - x_0 & & x_0 & | & f_0 \quad f[x_1, x_0, x_0] \quad f[x_1, x_1, x_0, x_0, x_0] \\ & & x_1 - x_0 & & & & f[x_1, x_0] \quad f[x_1, x_1, x_0, x_0] \\ & & & & x_1 - x_0 & & \\ & & & & & -- & \\ & & & & x_1 & | & f_1 \quad f[x_1, x_1, x_0] \\ & & & & & & f[x_1, x_1] \\ & & & & x_1 & | & f_1 \end{array}$$

### (3) Hermite-Interpolationspolynom

$$\begin{aligned} H_4(x) = & f_0 + f[x_0, x_0](x - x_0) + f[x_0, x_0, x_0](x - x_0)^2 \\ & + f[x_1, x_0, x_0, x_0](x - x_0)^3 + f[x_1, x_1, x_0, x_0, x_0](x - x_0)^3(x - x_1). \end{aligned}$$



**Bemerkung 1.2** Bei einer Stützstelle mit Funktionswert und Ableitungen bis zum Grad  $n$  erhält man das Interpolationspolynom

$$p(x) = \sum_{k=0}^n \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k,$$

was nichts anderes als die Taylor-Reihe der Funktion ist.

Man spricht deshalb in diesem Fall auch von der **Taylor-Interpolation**.

### 1.4.2 Allgemeine Referenz

Das Aufstellen des Schemas der dividierten Differenzen im Fall vorhandener Ableitungswerte verlangt die entsprechende Wiederholung der Stützstellen. Oben wurde dies demonstriert.

Günstiger ist es, eine allgemeine Stützstellenfolge zu konstruieren, die diese Wiederholungen enthält. Die Bezeichnung  $x_i$  für die Stützstellen behalten wir bei. Mit der Neunummerierung bzw. Neuindizierung vergrößert sich aber der maximale Index  $n$ .

**Definition 1.6 Allgemeine Referenz für  $f(x)$**

*Es ist die Darstellung der erweiterten Referenz in Form von Tripeln gemäß*

$$\begin{aligned} R_H &= \{(x_i, d_i, \mu_i(f)), i = 0, 1, \dots, n, \quad x_0 \leq x_1 \leq \dots \leq x_n\}, \\ \text{mit } d_i &= \max(j : x_i = x_{i-j}), \quad i = 0, 1, \dots, n, \text{ und der} \\ \text{Abbildung } \mu_i &: \mathcal{C}^n[a, b] \rightarrow \mathbb{R}, \quad \mu_i(f) = f^{(d_i)}(x_i). \end{aligned} \quad (1.43)$$

**Beispiel 1.9** Die Konstruktion der Referenz  $R_H$  zu den Größen

$$\left. \begin{array}{l} f(1) = 2, \quad f'(1) = 6, \quad f''(1) = 0 \\ f(0) = 1, \quad f'(0) = -10 \end{array} \right| \begin{array}{l} x_0 = x_1 = x_2 = 1, \\ x_3 = x_4 = 0, \quad n = 4, \end{array}$$

ergibt die folgende Übersicht.

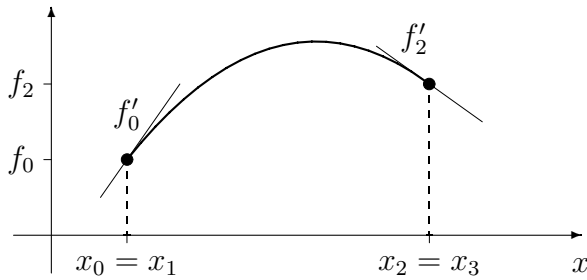
$x_i, i = 0, 1, 2, 3, 4 = n$	1	1	1	0	0
$d_i$	0	1	2	0	1
$\mu_i(f)$	2	6	0	1	-10
	$f(x_0)$	$f'(x_1)$	$f''(x_2)$	$f(x_3)$	$f'(x_4)$

Die Interpolationsaufgabe besteht im Auffinden eines Polynoms  $p(x) \in \mathcal{P}_n$  mit  $\mu_i(p) = \mu_i(f)$ ,  $i = 0, 1, \dots, n$ .

Die bisher betrachteten Fälle der Interpolation finden sich in dieser allgemeinen Version wieder.

### Spezialfälle

1. Lagrange- bzw. Newton-Interpolation mit  $x_0 < x_1 < \dots < x_n$ ,  $d_i = 0$ .
2. Taylor-Interpolation mit  $x_0 = x_1 = \dots = x_n$ ,  $d_i = i$ .
3. Hermite-Interpolation mit  $x_0 = x_1 < x_2 = x_3 < \dots < x_{n-1} = x_n$ ,  $d_{2k} = 0$ ,  $d_{2k+1} = 1$ .
4. Kubische Hermite-Interpolation mit  $x_0 = x_1 < x_2 = x_3$ ,  $n = 3$ .



**Abb. 1.2** Kubische Hermite-Interpolation bei einem Intervall

Mit dem letzten Fall machen wir den Übergang zum nächsten Abschnitt.

### 1.4.3 Kubische Hermite-Basis

Wir legen die allgemeine Referenz  $R_H$  im Fall  $n = 3$ ,  $x_0 = x_1 < x_2 = x_3$ , zu Grunde. Mit der Auswahl von entsprechenden Funktions- und Ableitungswerten kann man sogenannte Standardpolynome 3. Grades definieren.

#### Definition 1.7 Kubische Hermite-Polynome

$$H_3^{(i)}(x) \in \mathcal{P}_3, \quad i = 0, 1, 2, 3, \quad [H_3^{(i)}(x_k)]' = [H_3^{(i)}(x)]'|_{x=x_k}, \quad \text{wobei}$$

$$H_3^{(0)}(x_0) = 1, \quad [H_3^{(0)}(x_0)]' = 0, \quad H_3^{(0)}(x_2) = 0, \quad [H_3^{(0)}(x_2)]' = 0,$$

$$H_3^{(1)}(x_0) = 0, \quad [H_3^{(1)}(x_0)]' = 1, \quad H_3^{(1)}(x_2) = 0, \quad [H_3^{(1)}(x_2)]' = 0,$$

$$H_3^{(2)}(x_0) = 0, \quad [H_3^{(2)}(x_0)]' = 0, \quad H_3^{(2)}(x_2) = 1, \quad [H_3^{(2)}(x_2)]' = 0,$$

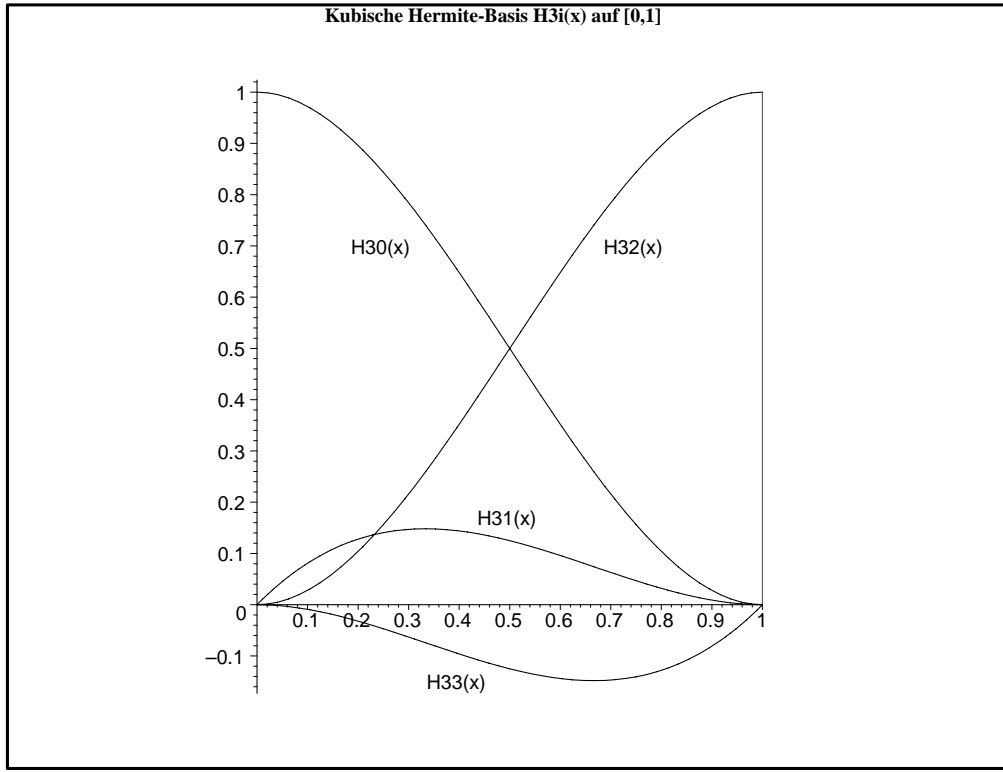
$$H_3^{(3)}(x_0) = 0, \quad [H_3^{(3)}(x_0)]' = 0, \quad H_3^{(3)}(x_2) = 0, \quad [H_3^{(3)}(x_2)]' = 1.$$

**Beispiel 1.10** Die kubischen Hermite-Polynome auf dem Intervall  $[0,1]$  sind

$$\begin{aligned} x_0 = x_1 &= 0, & x_2 = x_3 &= 1, \\ H_3^{(0)}(x) &= 2x^3 - 3x^2 + 1, & H_3^{(2)}(x) &= -2x^3 + 3x^2, \\ H_3^{(1)}(x) &= x^3 - 2x^2 + x, & H_3^{(3)}(x) &= x^3 - x^2, \end{aligned}$$

wobei  $H_3^{(0)}(x) + H_3^{(2)}(x) = 1$  und  $H_3^{(1)}(x) + H_3^{(3)}(x) = 2x(x - \frac{1}{2})(x - 1)$  gelten.





**Abb. 1.3** Die 4 kubischen Hermite-Basispolynome auf dem Intervall  $[0,1]$

**Beispiel 1.11** Wir notieren die kubischen Hermite-Polynome auf  $[a, b]$ .

$$x_0 = x_1 = a,$$

$$x_2 = x_3 = b,$$

$$H_3^{(0)}(x) = \left(\frac{x-b}{b-a}\right)^2 \left(1 + 2\frac{x-a}{b-a}\right), \quad H_3^{(2)}(x) = \left(\frac{x-a}{b-a}\right)^2 \left(1 + 2\frac{x-b}{a-b}\right), \quad (1.44)$$

$$H_3^{(1)}(x) = \frac{1}{(b-a)^2}(x-a)(x-b)^2, \quad H_3^{(3)}(x) = \frac{1}{(b-a)^2}(x-a)^2(x-b).$$

Die Polynome  $H_3^{(i)}(x)$ ,  $i = 0, 1, 2, 3$ , sind linear unabhängig und bilden somit eine Basis im Raum  $\mathcal{P}_3$ , die sogenannte **kubische Hermite-Basis**. Damit gilt, dass jedes kubische Polynom zur Referenz  $R_H = \{(a, 0, f(a)), (a, 1, f'(a)), (b, 0, f(b)), (b, 1, f'(b))\}$  die Darstellung

$$p(x) = f(a)H_3^{(0)}(x) + f'(a)H_3^{(1)}(x) + f(b)H_3^{(2)}(x) + f'(b)H_3^{(3)}(x). \quad (1.45)$$

besitzt.

Die Basis  $\{H_3^{(i)}\}$  wird bei der intervallweisen Hermite-Interpolation, also bei der Splineinterpolation angewandt. Das Zusammensetzen von solchen Teilstücken gestattet die Konstruktion einer stetigen und differenzierbaren Gesamtfunktion.

## 1.5 Fehler und Konvergenz

### 1.5.1 Fehlerfortpflanzung in Schemata

Rechnungen mit den verschiedenen Horner-Schemata für Polynome, die Auswertung von Interpolationsformeln oder die Polynomwertberechnung mit Schemata vom Aitken-Typ sind vom numerischen Standpunkt aus schon problematisch. Das betrifft sowohl die Fehlerakkumulation bei fortlaufender Multiplikation und Addition, als auch die Division durch kleine Zahlen. Es sei denn, man verwendet bessere Algorithmen. Dazu gehören zum Beispiel beim Interpolationsproblem die Splinetechnik.

Die Erkennung von zufälligen oder systematischen Fehlern im Schema oder Algorithmus ist nur bedingt möglich. Als Demonstration dient hier die Fehlerfortpflanzung in einem Differenzenschema, die durch Vorzeichenwechsel oder Störungen im monotonen Verhalten der Werte angezeigt wird.

$x_0$	0				
		0			
$x_1$	0		$\varepsilon$		
		$\varepsilon$		$-3\varepsilon$	
$x_2$	$\varepsilon$		$-2\varepsilon$		$6\varepsilon$
		$-\varepsilon$		$3\varepsilon$	
$x_3$	0		$\varepsilon$		
		0			
$x_4$	0				

**Tab. 1.2** Schema der Differenzen mit Ausbreitung eines lokalen Fehlers  $\varepsilon$

### 1.5.2 Konvergenz der Interpolation

Für die Grundtypen der Interpolation einer Funktion ist der **Interpolationsfehler**

$$R_n(x) = f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x - x_0)(x - x_1) \cdot \dots \cdot (x - x_n), \quad (1.46)$$

wobei  $\xi \in I = [a, b]$ , vorausgesetzt  $f \in \mathcal{C}^{n+1}(I)$ .

Man hofft, dass mit wachsendem  $n$  bei hinreichend glatter Funktion das Interpolationspolynom  $p_n(x)$  immer genauer wird, d. h. die Bedingung

$$\|f - p_n\|_\infty = \max_x |f(x) - p_n(x)| \rightarrow 0$$

für  $n \rightarrow \infty$  erfüllt ist.

Dabei sind natürlich einige Aspekte zu berücksichtigen:

- Glattheit der zu interpolierenden Funktion und Beschränktheit ihrer Ableitungen,
- Endlichkeit des Intervalls  $[a, b]$  mit den Stützstellen  $x_i$ ,
- Verteilung der Stützstellen  $x_i$  verbunden mit der Kondition der Interpolation.

**Beispiel 1.12** Sei  $f(x) = |x|$ ,  $x \in [-1, 1]$ ,  $x_0 = -1$ ,  $x_n = 1$ ,  $x_i$  äquidistant. Die Interpolationspolynome bei ausgewählten Knotenanzahlen sind:

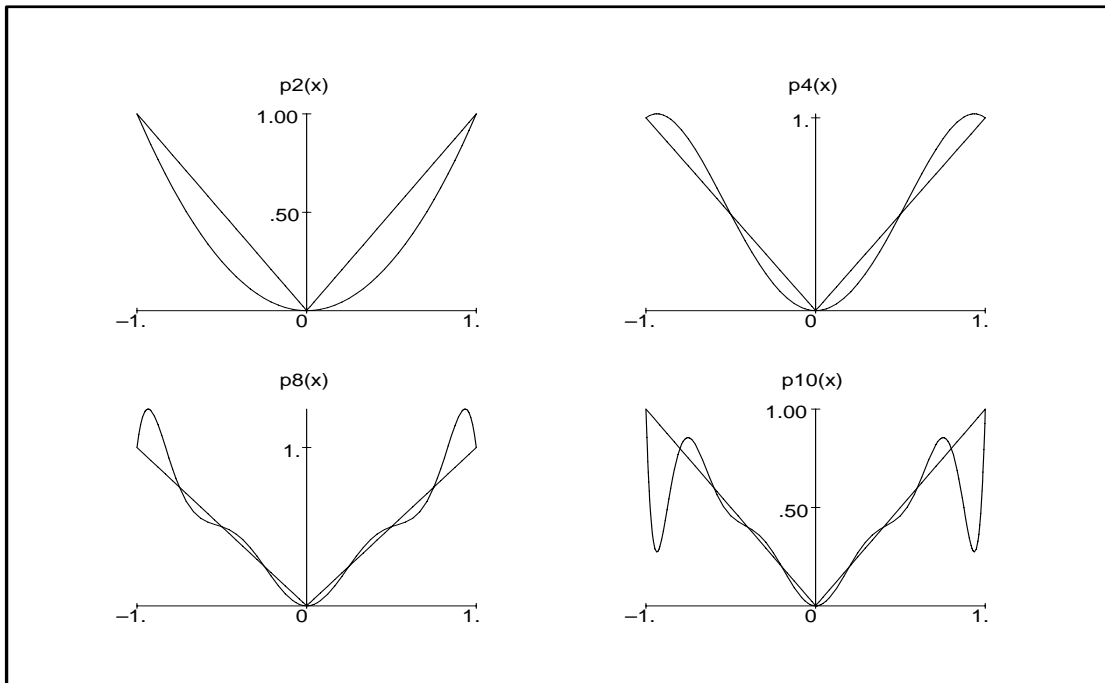
$$n = 2, \quad p_2(x) = x^2,$$

$$n = 4, \quad p_4(x) = -\frac{4}{3}x^4 + \frac{7}{3}x^2,$$

$$n = 8, \quad p_8(x) = -\frac{1024}{63}x^8 + \frac{1408}{45}x^6 - \frac{172}{9}x^4 + \frac{533}{105}x^2,$$

$$n = 10, \quad p_{10}(x) = \frac{390625}{5184}x^{10} - \frac{1015625}{6048}x^8 + \frac{221875}{1728}x^6 - \frac{13375}{324}x^4 + \frac{1627}{252}x^2.$$

Weiter gilt  $|f(x) - p_{20}(x)| \approx 100$  für  $x$  nahe 1.



**Abb. 1.4**  $f(x) = |x|$  und ausgewählte Interpolationspolynome auf  $[-1, 1]$

Ein Grund für das wachsende oszillierende Verhalten der Interpolationspolynome ist, dass die Funktion im Punkt Null keine Ableitung besitzt. Das ist aber nicht die alleinige Ursache.

Man bemerke, dass in der Mitte des Intervalls eine gute Näherung vorliegt, während zu den Rändern hin die Approximation immer schlechter wird. Diese Eigenschaft wird **Gibbscher** oder **Rungescher Effekt** bzw. **Phänomen** genannt.

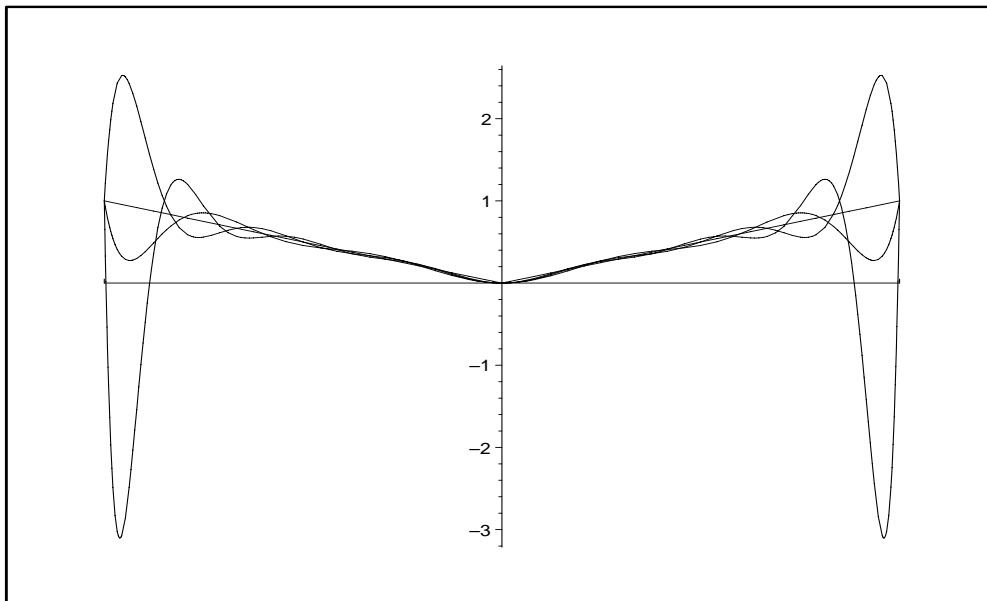
**Beispiel 1.13** Sei  $f(x) = \frac{1}{1 + 25x^2}$ ,  $x \in [-1, 1]$ ,  $x_0 = -1$ ,  $x_n = 1$ ,  $x_i$  äquidistant.

Trotz Glattheit der Funktion tritt auch hier bei äquidistanten Stützstellen der Gibbsche Effekt auf. Also ist eine andere Stützstellenverteilung zu wählen.

**Beispiel 1.14****(1) Das Beispiel von Bernstein**

Schon Anfang des 20. Jahrhunderts wurde die Frage der punktweisen und gleichmäßigen Konvergenz von Interpolationspolynomen gegen die interpolierte Funktion für den Fall, dass die Anzahl der Stützstellen anwächst, ausführlich untersucht.

Für die stetige Funktion  $f(x) = |x|$  konvergiert die Folge der äquidistanten Interpolationspolynome auf  $[-1, 1]$  in keinem Punkt außer den Randpunkten  $\pm 1$ .



**Abb. 1.5** Interpolation von  $f(x) = |x|$ ,  $x \in [-1, 1]$ ,  
durch Polynome mit äquidistanten Stützstellen (3 Fälle)

Maple-Anweisungen dazu

```
> restart;
> with(plots):

> ffejer:=x->abs(x);  Intervall:=-1..1:
  r:=3;
> a:=op(1,Intervall): b:=op(2,Intervall):
> for i from 1 to r do
>   n:=8+2*i;
>   x:=seq(a+(b-a)/n*k,k=0..n);
>   y:=map(ffejer,x);
>   pol.i:=unapply(interp(x,y,t),t);
> end do;
> plot([ffejer,seq(pol.i,i=1..r)],-1..1,thickness=[2,1$r],
      xtickmarks=[a,b],color=black);
```

## (2) Bernstein-Polynome

Im Jahre 1912 bewies Sergei Natanowitsch Bernstein, dass es zu jeder stetigen Funktion  $f$  auf einem abgeschlossenen, beschränkten Intervall eine Folge von Polynomen gibt, die gleichmäßig gegen  $f$  konvergiert.

Für das Intervall  $[0,1]$  lässt sich eine solche Folge explizit angeben.

Die Folge der Bernstein-Polynome  $\{B_n(x)\}_{n \in \mathbb{N}}$

$$B_n f(x) = B_n(x) = \sum_{k=0}^n f\left(\frac{k}{n}\right) \binom{n}{k} x^k (1-x)^{n-k}$$

hat für die stetige Funktion  $f$  die Eigenschaft

$$\lim_{n \rightarrow \infty} \|f - B_n\|_{\infty} = \lim_{n \rightarrow \infty} \max_{x \in [0,1]} |f(x) - B_n(x)| = 0.$$

Die Bernstein-Polynome interpolieren **nicht** in den Stützstellen  $\frac{k}{n}$ ,  $k = 0, 1, \dots, n$ , sie approximieren die Funktion  $f$  nur. Wegen ihrer guten Formerhaltungseigenschaften (Invarianz unter affinen Abbildungen) spielt diese polynomiale Approximation stetiger Funktionen eine große praktische Rolle. Naheliegend war daher die Fragestellung, ob auch die Folge der Interpolationspolynome zu äquidistanten Stützstellen punktweise oder gar gleichmäßig konvergiert. Mit dem Beispiel aus Abschnitt (1) beantwortete Bernstein selbst diese Frage negativ.

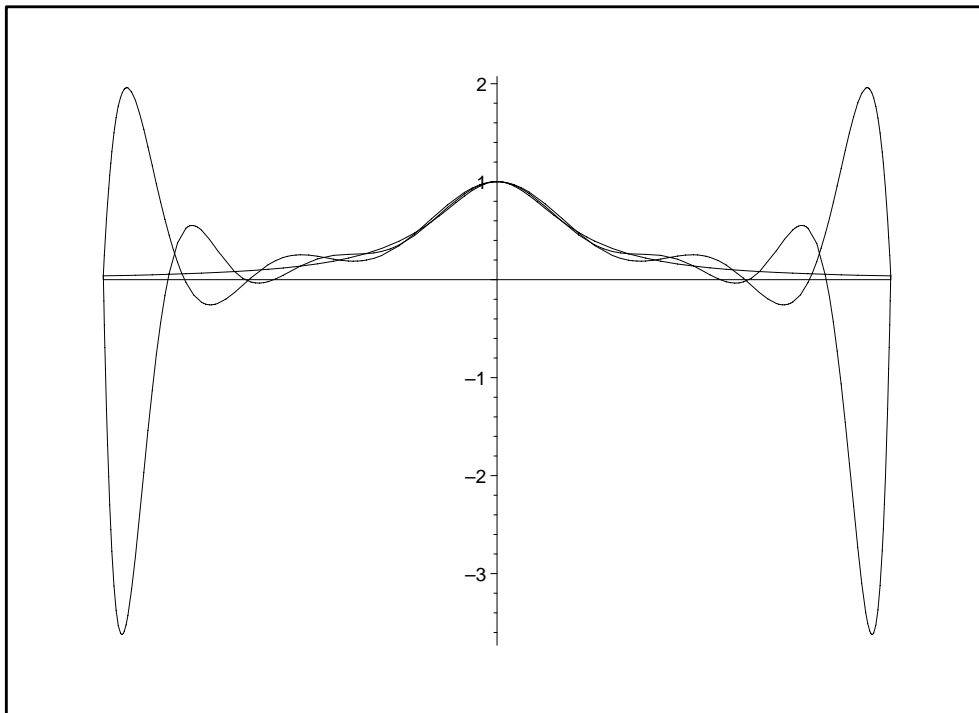
## (3) Das Beispiel von Runge

Von Carl D.T. Runge (1856-1927) stammt der Beweis, dass die Interpolationspolynome  $p_n$  zu  $n+1$  äquidistant verteilten Stützstellen für die unendlich oft differenzierbare Funktion

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

auf dem Intervall  $[-5, 5]$  nur auf dem Teilstück  $I = [-3.63\dots, 3.63\dots]$  punktweise gegen  $f$  konvergieren und außerhalb von  $I$  in jedem Punkt, außer den Randpunkten  $-5$  und  $5$ , divergieren.

Dieses Beispiel wird auch heute noch oft benutzt, um das mögliche ungünstige Verhalten von Interpolationspolynomen hohen Grades zu demonstrieren.



**Abb. 1.6** Beispiel von Runge mit  $f(x) = 1/(1+x^2)$ ,  $x \in [-5, 5]$ ,  
mit  $n = 10$  bzw. 12 Interpolationsstützstellen

Maple-Anweisungen dazu

```
> restart:
> with(plots):

> r:=3:
> frunge:=x->1/(1+x^2);
> for i from 2 to r do
>   n:=6+2*i;
>   x:=seq(-5+10/n*k,k=0..n);
>   y:=map(frunge,x);
>   pol.i:=unapply(interp(x,y,t),t);
> end do:
> plot([frunge,pol.2,pol.3],-5..5,thickness=[2,1,1],
      xtickmarks=[-5,5],color=black);
```

Tatsächlich wurde schon 1914 von Bernstein und Faber bewiesen, dass es zu jeder Stützstellenfolge eine stetige Funktion gibt, für welche die Folge der Interpolationspolynome nicht punktweise konvergiert.

### 1.5.3 Konvergenzsätze

Entscheidend bei der Abschätzung des Interpolationsfehlers und damit bei der Untersuchung der Konvergenz sind die Beschränktheit der Ableitungen der zu interpolierenden Funktion sowie die Wahl der Stützstellen.

#### Satz 1.13 Konvergenz

Ist die Funktion  $f$  aus  $\mathcal{C}^\infty(I)$ ,  $I = [a, b]$ , und genügen ihre sämtlichen Ableitungen der Bedingung  $|f^{(n)}(x)| \leq M$ ,  $x \in I$ ,  $n = 1, 2, \dots$ , so gilt für die Interpolationspolynome  $p_n(x)$  mit zugehöriger Referenz

$$\forall x \in I \quad \lim_{n \rightarrow \infty} |f(x) - p_n(x)| = 0. \quad (1.47)$$

**Beweis.** Man betrachtet den Interpolationsfehler  $R_n(x)$  mit  $|x - x_i| \leq b - a$  und berücksichtigt, dass  $\lim_{n \rightarrow \infty} (b - a)^{n+1}/(n+1)! = 0$  ist.  $\square$

**Bemerkung 1.3** (1) Die obige Konvergenz ist auch gleichmäßig in der Norm.

(2) Die Funktionen  $\sin(x), \cos(x), e^x, \dots$  haben auf  $[a, b]$  gleichmäßig beschränkte Ableitungen beliebig hoher Ordnung, so dass der Satz 1.13 angewendet werden kann.

(3)  $f \in \mathcal{C}^\infty[a, b]$  ist oft nicht erfüllt.

(4)  $f \in \mathcal{C}^\infty$  kann ersetzt werden durch die Voraussetzung, dass  $f$  eine ganze Funktion ist, d. h., sie ist darstellbar für jedes  $x$  als konvergente Potenzreihe

$$f(x) = \sum_{k=0}^{\infty} a_k (x - x_0)^k.$$

(5) Der Weierstraßsche Approximationssatz [15] gilt für eine stetige Funktion.

Warum werden hier solche scharfen Bedingungen an die Glattheit gestellt?

Die ungünstige Variante, die an zwei Beispielen schon demonstriert wurde, beschreibt folgender Satz.

#### Satz 1.14 G. FABER

Zu jeder Stützstellenfolge  $\{R_n\}$  lässt sich eine auf  $[a, b]$  stetige Funktion  $f$  angeben, für die das Folgende nicht gilt.

$$\max_{x \in [a, b]} |f(x) - p_n(x)| \rightarrow 0 \quad \text{für } n \rightarrow \infty$$

[G. Faber: Über die interpolatorische Darstellung stetiger Funktionen. Jahresbericht D. Math. Vereinigung 23, 1914]

**Beispiel 1.15** Sei  $[a, b] = [-1, 1]$  und

$$R_n = \{x_i\}_{i=0}^n = \{-1, -1 + \frac{2}{n}, -1 + \frac{4}{n}, \dots, 1 - \frac{2}{n}, 1\}, \quad x_i \text{ äquidistant.}$$

Für diese Referenzen  $R_n$  ist  $f(x) = |x|$  eine solche stetige Funktion im Satz 1.14.

Mehr noch, nach S.N. Bernstein gilt sogar die Aussage:

Für **kein**  $x \in (-1, 1) \setminus \{0\}$  ist  $||x| - p_n(x)| \rightarrow 0$  für  $n \rightarrow \infty$ .

Nun gibt es aber auch eine günstige Situation für stetige Funktionen.

Aus dem Weierstraßschen Approximationssatz und dem Satz von Tschebyscheff über Alternanten (Punktfolge mit einem Polynom, das "sich um die Funktion legt") folgt eine positive Antwort.

**Satz 1.15** J. MARCINKIEWICZ, J.M. JANKOWSCY

*Zu jeder stetigen Funktion  $f$  auf  $[a, b]$  gibt es mindestens eine Folge von Stützstellen  $\{R_n\}$ , so dass die zugehörigen Interpolationspolynome  $p_n$  für  $n \rightarrow \infty$  gleichmäßig gegen  $f$  streben.*

Das Problem ist somit die Angabe solcher Folgen, zumal für andere Stützstellenfolgen genau das Gegenteil eintreten kann.

**Satz 1.16** *Ist  $f \in C^1[-1, 1]$ , so konvergiert die Folge der Interpolationspolynome  $p_n$  mit der **Tschebyscheff-Referenz**, das sind die Stützstellen*

$$x_i = -\cos\left(\frac{2i+1}{2n+2}\pi\right) \in (-1, 1), \quad i = 0, 1, \dots, n, \quad (1.48)$$

*und zugehörigen Stützwerte für  $n \rightarrow \infty$  gleichmäßig gegen  $f$  auf  $[-1, 1]$ .*

**Bemerkung 1.4** (1) Im Produktpolynom

$$\omega_{n+1}(x) = \Pi_n(x) = \prod_{i=0}^n (x - x_i),$$

das in den Randbereichen des Intervalls zu starker Oszillation neigt, ist dort somit eine Verdichtung der Stützstellen vorzunehmen. Das gelingt mit der Tschebyscheff-Referenz.

(2) Für das Intervall  $[a, b]$  sind die Stützstellen nach Transformation

$$x_i = \frac{a+b}{2} - \frac{b-a}{2} \cos\left(\frac{2i+1}{2n+2}\pi\right).$$

(3) Die Tschebyscheff-Stützstellen  $-\cos((2i+1)\pi/(2n+2))$ ,  $i = 0, 1, \dots, n$ , sind die Nullstellen des Tschebyscheff-Polynoms  $T_{n+1}(x)$ .

(4) Die Tschebyscheff-Stützstellenfolge  $\{x_i\}$  auf  $[a, b]$  erfüllt für das Produktpolynom die Bedingung

$$\max_{x \in [a, b]} |\omega_{n+1}(x)| \rightarrow \min.$$

Man nennt diese auch **Minimax-Eigenschaft** der Tschebyscheff-Polynome.



### 1.5.4 Bemerkungen zur Wahl der Tschebyscheff-Referenz

Eine ausführliche Beschreibung der Tschebyscheff-Polynome  $T_n(x)$  und ihrer Eigenschaften findet man in [28].

Hier betrachten wir speziell die genannte Minimax-Eigenschaft. Wir verwenden dazu die folgenden Informationen über die Polynome  $T_n(x)$ .

$$T_n(x) = \cos(n \arccos(x)), \quad x \in [-1, 1], \quad n = 0, 1, \dots,$$

$$|T_n(x)| \leq 1, \quad \|T_n\|_\infty = 1,$$

$$T_0(x) = 1,$$

$$T_1(x) = x,$$

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \quad n = 2, 3, \dots \quad (\text{Rekursion}),$$

$$T_n(x) = 2^{n-1}x^n - a_2x^{n-2} + \dots = 2^{n-1}(x - x_0)(x - x_1) \cdot \dots \cdot (x - x_{n-1}),$$

Hauptkoeffizient  $a_0$  von  $T_n(x)$  bei  $x^n$  ist  $2^{n-1}$ .

$$T_n(x) \text{ hat die Nullstellen } x_i = -\cos\left(\frac{2i+1}{2n}\pi\right) \in (-1, 1), \quad i = 0, 1, \dots, n-1.$$

$$T_n(x) \text{ hat die Extremalstellen } \bar{x}_i = -\cos\left(\frac{i}{n}\pi\right) \in [-1, 1], \quad i = 0, 1, \dots, n,$$

$$T_n(\bar{x}_i) = (-1)^{n-i} = \pm 1, \quad i = 0, 1, \dots, n.$$

Das modifizierte Polynom  $\bar{T}_n(x) = 2^{-n+1}T_n(x)$  hat den Hauptkoeffizient Eins und genügt der Beziehung  $\|\bar{T}_n\|_\infty = 2^{-n+1}$ .

$\bar{T}_n(x)$  besitzt die Form eines Knotenpunktpolynoms. Aber im Vergleich dazu hat im Interpolationsfehler  $R_{n-1}(x)$  das dort auftretende Knotenpunktpolynom  $\omega_n(x) = (x - x_0)(x - x_1) \cdot \dots \cdot (x - x_{n-1})$  mit dem Hauptkoeffizienten Eins an irgendeiner Zwischenstelle einen Wert vom Betrag  $\geq 1/2^{n-1}$ , was im nachfolgenden Satz gezeigt wird.

**Satz 1.17** Jedes Polynom  $p_n \in \mathcal{P}_n$  mit führendem Koeffizient  $a_0 \neq 0$  nimmt in  $[-1, 1]$  einen Wert vom Betrag  $\geq |a_0|/2^{n-1}$  an.

Insbesondere sind die Tschebyscheff-Polynome  $T_n(x)$  unter allen Polynomen in  $\mathcal{P}_n$  mit führendem Koeffizienten  $2^{n-1}$  minimal bezüglich der Maximumnorm der Funktion  $\|f\|_\infty = \max_{x \in [-1, 1]} |f(x)|$ .

**Beweis.** (indirekt)

(1) Sei  $p_n \in \mathcal{P}_n$  ein anderes Polynom mit dem führenden Koeffizienten  $a_0 = 2^{n-1}$  und angenommen, dass  $\|p_n\|_\infty < 1$  ist.

Wir untersuchen das Polynom  $T_n(x) - p_n(x) \in \mathcal{P}_{n-1}$ .

An den  $n+1$  Extremalstellen  $\bar{x}_i = -\cos(i\pi/n)$ ,  $i = 0, 1, \dots, n$ , gilt bei Unterscheidung von geradem und ungeradem Index das Folgende.

$$\begin{aligned} T_n(\bar{x}_{2k}) &= 1, & p_n(\bar{x}_{2k}) &< 1, & T_n(\bar{x}_{2k}) - p_n(\bar{x}_{2k}) &> 0, \\ T_n(\bar{x}_{2k+1}) &= -1, & p_n(\bar{x}_{2k+1}) &> -1, & T_n(\bar{x}_{2k+1}) - p_n(\bar{x}_{2k+1}) &< 0. \end{aligned}$$

Damit hat  $T_n(x) - p_n(x)$  an  $n+1$  Stellen alternierendes Vorzeichen, somit  $n$  verschiedene Nullstellen in  $[-1, 1]$ . Das ist aber ein Widerspruch dazu, dass das Polynom  $T_n(x) - p_n(x) \in \mathcal{P}_{n-1}$  verschieden von Null ist.

(2) Betrachten wir nun ein beliebiges Polynom  $p_n \in \mathcal{P}_n$ .

Dann definiert man das neue Polynom  $\tilde{p}_n = \frac{2^{n-1}}{a_0} p_n \in \mathcal{P}_n$  mit dem führenden Koeffizienten  $2^{n-1}$  und gelangt so zum Teil (1) des Beweises.  $\square$

**Folgerung 1.18** Das Knotenpunktpolynom  $\omega_{n+1}(x) = (x-x_0)(x-x_1)\cdots(x-x_n)$  wird minimal im Sinne von  $\max_{x \in [-1,1]} |\omega_{n+1}(x)| \rightarrow \min$  mit den  $n+1$  Nullstellen  $x_i = -\cos((2i+1)\pi/(2n+2))$ ,  $i = 0, 1, \dots, n$ , des Tschebyscheff-Polynoms  $T_{n+1}(x)$ .

**Beispiel 1.16** Darstellung der Knotenpunktpolynome

$\omega_{n+1}(x) = (x-x_0)(x-x_1)\cdots(x-x_n)$  für 2 Arten von Stützstellen

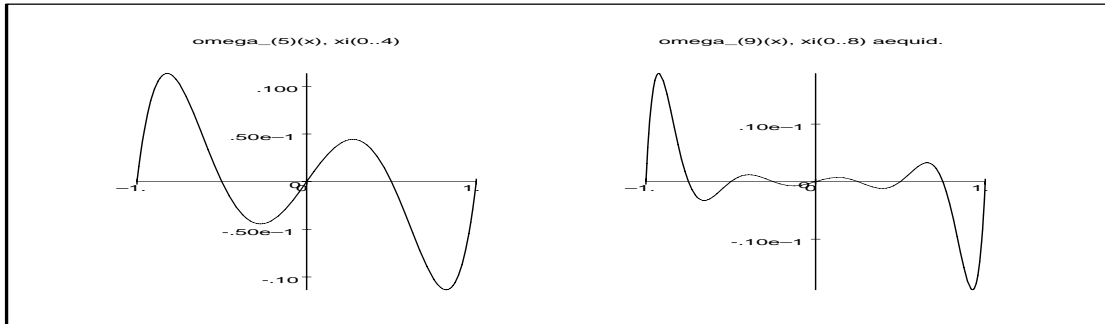


Abb. 1.7 Äquidistante St.  $x_i = -1 + 2\frac{i}{n} \in [-1, 1]$ ,  $i = 0, 1, \dots, n$ ,  $n = 4, 8$

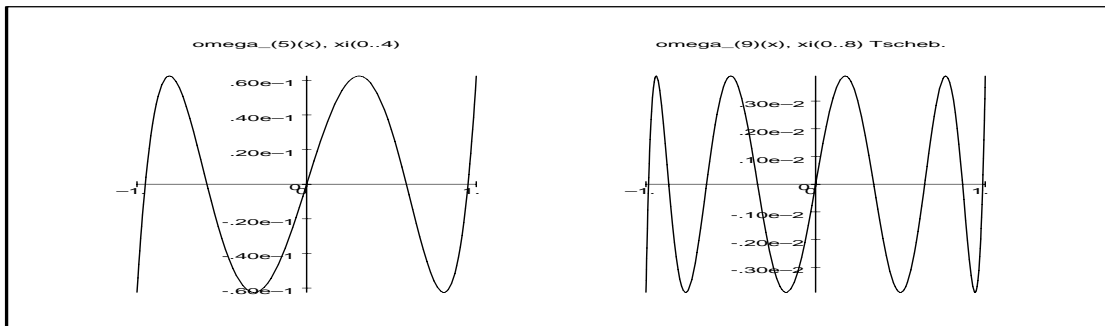
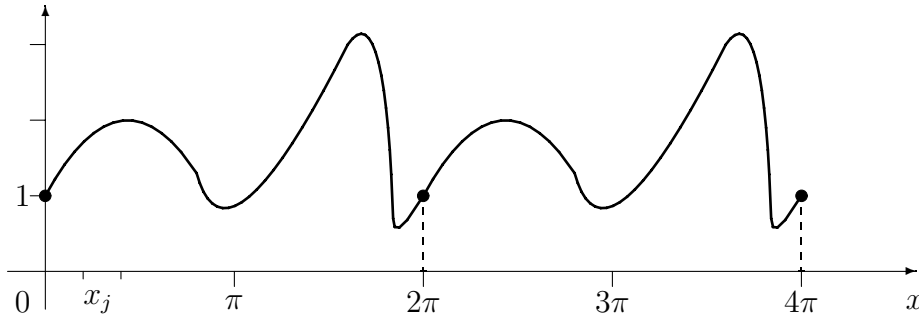


Abb. 1.8 Tschebyscheff-St.  $x_i = -\cos(\frac{2i+1}{2n+2}\pi) \in (-1, 1)$ ,  $i = 0, 1, \dots, n$ ,  $n = 4, 8$

## 1.6 Trigonometrische Interpolation

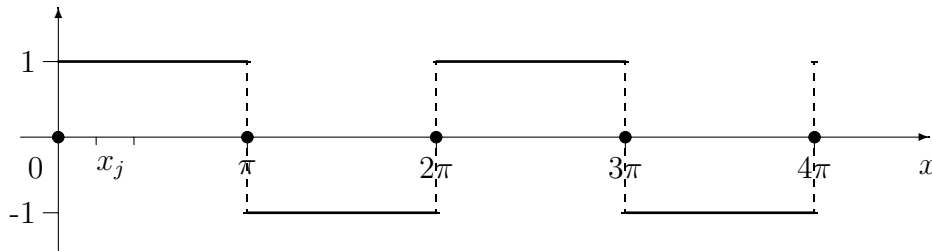
### 1.6.1 Übersicht und Merkmale

(1) Die Anwendung der trigonometrischen Interpolation erfolgt auf periodische Funktionen  $f(x)$  mit der Periode  $T > 0$ , speziell  $T = 2\pi$ .



**Abb. 1.9** Stetige  $2\pi$ -periodische Funktion

Bei unstetigen Funktionen wie der Signalfunktion wird an den Sprungstellen als Funktionswert das arithmetische Mittel zwischen linken und rechten Grenzwert genommen.



**Abb. 1.10**  $2\pi$ -periodische Funktion mit Unstetigkeitsstellen

(2) Die Referenz basiert auf dem Periodizitätsintervall  $[0, 2\pi]$  und den  $N$  äquidistanten Stützstellen

$$x_j = \frac{2\pi j}{N}, \quad j = 0, 1, \dots, N-1, \quad N = \begin{cases} \text{ungerade} = 2n+1, \\ \text{gerade} = 2n, \\ 2^p, \quad p = 1, 2, \dots \end{cases} \quad (1.49)$$

(3) Die Betrachtung der Interpolationspolynome wird im Komplexen und Reellen durchgeführt. Dabei verwendet man die folgenden Ansätze.

$$\begin{aligned} \mathbb{C}: \quad \Phi_N(x) &= \sum_{j=0}^{N-1} c_j e^{ijx}, \\ \mathbb{R}: \quad \Phi_N(x) &= \frac{a_0}{2} + \sum_{j=1}^{N/2-1} [a_j \cos(jx) + b_j \sin(jx)] + \frac{a_{N/2}}{2} \cos(N/2 x), \end{aligned} \quad (1.50)$$

wobei meistens  $N = 2^p$  gesetzt wird und  $i = \sqrt{-1}$  die imaginäre Einheit ist.

(4) Dazu braucht man die  $N$ -ten Einheitswurzeln  $\omega_j$  mit einigen ihrer Eigenschaften.

$$\begin{aligned}\omega_j &= e^{i2\pi j/N}, \quad \omega_j^0 = \omega_j^N = 1, \quad \omega_j^k = \omega_k^j, \\ \omega_j^l &= \omega_j^{l \bmod N},\end{aligned}$$

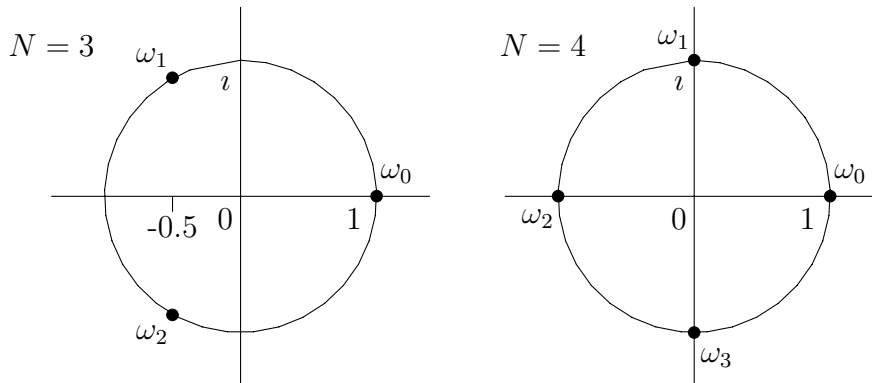
$$\sum_{j=0}^{N-1} \omega_j^k \omega_j^{-l} = N \delta_{kl},$$

$$\sum_{j=0}^{N-1} \omega_j^k = \begin{cases} N & \text{für } k = 0, \\ 0 & \text{für } k = 1, 2, \dots, N-1. \end{cases}$$

Die Funktionen

$$\Psi_k(x) = e^{ikx}$$

bilden ein ONS mit dem komplexen Skalarprodukt  $(f, g) = \frac{1}{N} \sum_{j=0}^{N-1} f(x_j) \overline{g(x_j)}$ ,



**Abb. 1.11** Komplexe Wurzeln  $\omega_j$  auf dem Einheitskreis

(5) Eine effiziente Berechnung der Terme  $\cos(jx)$  und  $\sin(jx)$  erfolgt unter Verwendung der Additionstheoreme ausgehend von den Werten  $\cos(x)$  und  $\sin(x)$ .

$$\cos(2x) = \cos(x) \cos(x) - \sin(x) \sin(x) = \cos^2(x) - \sin^2(x) = 2 \cos^2(x) - 1,$$

$$\sin(2x) = \sin(x) \cos(x) + \cos(x) \sin(x) = 2 \sin(x) \cos(x),$$

$$\cos(3x) = \cos(2x + x) = 4 \cos^3(x) - 3 \cos(x),$$

$$\sin(3x) = \sin(2x + x) = 3 \sin(x) - 4 \sin^3(x),$$

$$\cos(4x) = \cos(3x + x) = 8 \cos^4(x) - 8 \cos^2(x) + 1,$$

$$\sin(4x) = \sin(3x + x) = 4 \cos(x) \sin(x) (1 - 2 \sin^2(x)).$$

(6) Bei der Berechnung der Polynomkoeffizienten benutzt man die Idee der Faltung.

$$\begin{aligned}
 \mathbb{C}: \quad c_j &= \frac{1}{N} \sum_{k=0}^{N-1} f_k \omega_k^{-j}, \quad j = 0, 1, \dots, N-1, \quad f_k = f(x_k), \\
 \mathbb{R}: \quad a_j &= c_j + c_{N-j}, \quad a_0 = 2c_0, \\
 b_j &= \imath(c_j - c_{N-j}), \\
 \omega_k^{-j} &= e^{-\imath 2\pi jk/N} = \cos\left(\frac{2\pi jk}{N}\right) - \imath \sin\left(\frac{2\pi jk}{N}\right).
 \end{aligned} \tag{1.51}$$

(7) Für einen Fall wollen wir die Darstellung detaillierter machen.

$$N = 8, \quad \omega_k^{-j} = \cos\left(\frac{\pi jk}{4}\right) - \imath \sin\left(\frac{\pi jk}{4}\right), \quad k = 0, 1, \dots, 7.$$

Wir berechnen die Größe  $c_1$  mit  $\omega_k^{-1}$ ,  $k = 0, 1, \dots, 7$ , analog geschieht es für die anderen  $c_j$ .

$$\begin{aligned}
 \omega_0^{-1} &= 1, \\
 \omega_1^{-1} &= \cos\left(\frac{\pi}{4}\right) - \imath \sin\left(\frac{\pi}{4}\right), \dots \\
 \omega_7^{-1} &= \cos\left(\frac{7\pi}{4}\right) - \imath \sin\left(\frac{7\pi}{4}\right).
 \end{aligned}$$

Faltung bedeutet das paarweise Zusammenfassen von Summanden mit gleichen Werten der trigonometrischen Funktionen.

$k$	$\dots$	$=$	$\dots$	Faltung
1, 7	$\cos(\frac{\pi}{4})$		$\cos(\frac{7\pi}{4})$	$f_1 \cos(\frac{\pi}{4}) + f_7 \cos(\frac{7\pi}{4}) = (f_1 + f_7) \cos(\frac{\pi}{4})$
2, 6	$\cos(\frac{2\pi}{4})$		$\cos(\frac{6\pi}{4})$	$f_2 \cos(\frac{2\pi}{4}) + f_6 \cos(\frac{6\pi}{4}) = (f_2 + f_6) \cos(\frac{2\pi}{4})$
3, 5	$\cos(\frac{3\pi}{4})$		$\cos(\frac{5\pi}{4})$	$f_3 \cos(\frac{3\pi}{4}) + f_5 \cos(\frac{5\pi}{4}) = (f_3 + f_5) \cos(\frac{3\pi}{4})$
4	$\cos(\frac{\pi}{4})$			$f_4 \cos(\frac{4\pi}{4})$
	$\dots$	$=$	$\dots$	
1, 7	$\sin(\frac{\pi}{4})$		$-\sin(\frac{7\pi}{4})$	$f_1 \sin(\frac{\pi}{4}) + f_7 \sin(\frac{7\pi}{4}) = (f_1 - f_7) \sin(\frac{\pi}{4})$
2, 6	$\sin(\frac{2\pi}{4})$		$-\sin(\frac{6\pi}{4})$	$f_2 \sin(\frac{2\pi}{4}) + f_6 \sin(\frac{6\pi}{4}) = (f_2 - f_6) \sin(\frac{2\pi}{4})$
3, 5	$\sin(\frac{3\pi}{4})$		$-\sin(\frac{5\pi}{4})$	$f_3 \sin(\frac{3\pi}{4}) + f_5 \sin(\frac{5\pi}{4}) = (f_3 - f_5) \sin(\frac{3\pi}{4})$
4	$\sin(\frac{\pi}{4})$			$f_4 \sin(\frac{4\pi}{4})$

**Tab. 1.3** Algorithmus zur 1. Faltung für  $c_1$

Natürlich ist die Faltung der cos- und sin-Glieder auch im Reellen möglich. Ist  $N$  eine Zweierpotenz, so wird eine Folge von Faltungen realisiert.

### 1.6.2 Komplexe und reelle trigonometrische Interpolation

Wir betrachten periodische Funktionen sowie Referenzen mit Stützstellen in  $[0, T]$  mit der Periode  $T = 2\pi$ .

Die Stützstellen seien äquidistant und  $x_j = 2\pi j/N \in [0, 2\pi)$ ,  $j = 0, 1, \dots, N-1$ .

**Definition 1.8** Sei  $\mathcal{T}_{\mathbb{C}}^N$  der  $N$ -dimensionale Raum der komplexen trigonometrischen Polynome

$$\Phi_N(x) = \sum_{j=0}^{N-1} c_j e^{ijx}, \quad c_j \in \mathbb{C}, \quad i = \sqrt{-1}. \quad (1.52)$$

$\mathcal{T}_{\mathbb{R}}^N$  sei der Raum der reellen trigonometrischen Polynome.

Für  $N = 2n + 1$  ist

$$\Phi_{2n+1}(x) = \frac{a_0}{2} + \sum_{j=1}^n [a_j \cos(jx) + b_j \sin(jx)], \quad a_j, b_j \in \mathbb{R}, \quad (1.53)$$

für  $N = 2n$  ist

$$\Phi_{2n}(x) = \frac{a_0}{2} + \sum_{j=1}^{n-1} [a_j \cos(jx) + b_j \sin(jx)] + \frac{a_n}{2} \cos(nx). \quad (1.54)$$

Die Basisfunktionen sind

$$\{e^{ijx}\} \text{ bzw. } \{1, \cos(jx), \sin(jx)\}$$

und linear unabhängig.

#### Satz 1.19 Existenz und Eindeutigkeit

Zur Referenz, d. h. zu den  $N$  Stützpunkten  $(x_k, f_k)$ ,  $k = 0, 1, \dots, N-1$ , existiert genau ein interpolierendes trigonometrisches Polynom  $\Phi_N(x) \in \mathcal{T}_{\mathbb{C}}^N$  bzw.  $\Phi_N(x) \in \mathcal{T}_{\mathbb{R}}^N$ .

#### Komplexe Interpolation

Wir fassen die bei der Interpolation auftretenden Aspekte in mehreren Anstrichen zusammen.

- Interpolationsforderung (Interpolationsbedingung)

$$\Phi_N(x_k) = f_k, \quad k = 0, 1, \dots, N-1. \quad (1.55)$$

- Interpolationsaufgabe (IA3)

Gesucht sind die Koeffizienten  $c_0, c_1, \dots, c_{N-1}$  in (1.52), so dass die Interpolationsforderung (1.55) erfüllt ist.



Für die Koeffizienten  $a_j, b_j$  macht man eine Fallunterscheidung.

(1)  $N$  ungerade,  $N = 2n + 1$

$$\begin{aligned}
 \Phi_N(x_k) &= c_0 + \sum_{j=1}^{2n} c_j e^{ijx_k} \\
 &= c_0 + \sum_{j=1}^n (c_j e^{ijx_k} + c_{N-j} e^{i(N-j)x_k}) \\
 &= c_0 + \sum_{j=1}^n (c_j e^{ijx_k} + \bar{c}_j e^{-ijx_k}) \\
 &= c_0 + \sum_{j=1}^n \left( c_j e^{ijx_k} + \overline{c_j e^{ijx_k}} \right), \quad z + \bar{z} = 2\Re(z) \\
 &= c_0 + \sum_{j=1}^n 2\Re(c_j e^{ijx_k}) \\
 &= c_0 + \sum_{j=1}^n 2\Re((\Re(c_j) + i\Im(c_j))(\cos(jx_k) + i\sin(jx_k))) \\
 &= c_0 + \sum_{j=1}^n 2[\Re(c_j) \cos(jx_k) - \Im(c_j) \sin(jx_k)].
 \end{aligned}$$

(2)  $N$  gerade,  $N = 2n$

Wegen  $c_j = \overline{c_{N-j}}$  folgt insbesondere  $c_{N/2} = c_n = \overline{c_n}$  und somit ist  $c_n$  reell.

Weiter gilt

$$\begin{aligned}
 \Phi_N(x_k) &= c_0 + \sum_{j=1}^{2n-1} c_j e^{ijx_k} \\
 &= c_0 + \sum_{j=1}^{n-1} (c_j e^{ijx_k} + c_{N-j} e^{i(N-j)x_k}) + c_n e^{inx_k} \\
 &= c_0 + \sum_{j=1}^{n-1} (c_j e^{ijx_k} + \bar{c}_j e^{-ijx_k}) + c_n (\cos(nx_k) + i\sin(nx_k)), \\
 &\qquad\qquad\qquad \sin(nx_k) = 0 \\
 &= c_0 + \sum_{j=1}^{n-1} \left( c_j e^{ijx_k} + \overline{c_j e^{ijx_k}} \right) + c_n \cos(nx_k) \\
 &= c_0 + \sum_{j=1}^{n-1} 2\Re(c_j e^{ijx_k}) + c_n \cos(nx_k) \\
 &= c_0 + \sum_{j=1}^{n-1} 2[\Re(c_j) \cos(jx_k) - \Im(c_j) \sin(jx_k)] + c_n \cos(nx_k).
 \end{aligned}$$

□



**Lemma 1.21** Orthonormalität der Basisfunktionen  $\Psi_k(x) = e^{ikx}$

Für die  $N$ -ten Einheitswurzeln  $\omega_j = e^{ix_j} = e^{i2\pi j/N}$  gilt

$$\sum_{j=0}^{N-1} \omega_j^k \omega_j^{-l} = N \delta_{kl}. \quad (1.58)$$

Insbesondere sind die  $\Psi_k(x)$  bezüglich des (komplexen) Skalarprodukts

$$(f, g) = \frac{1}{N} \sum_{j=0}^{N-1} f(x_j) \overline{g(x_j)} \text{ orthonormal, d. h. } (\Psi_k, \Psi_l) = \delta_{kl}.$$

**Beweis.**

(1) Wegen  $\omega_j^k \omega_j^{-l} = \omega_j^{k-l}$ ,  $\omega_j^k = \omega_k^j$ ,  $\omega_j^0 = \omega_j^N = 1$  ist nur zu zeigen, dass

$$\sum_{j=0}^{N-1} \omega_k^j = N \delta_{0k}.$$

Es gilt  $0 = \omega_k^N - 1 = (\omega_k - 1)(\omega_k^{N-1} + \omega_k^{N-2} + \dots + 1)$ .

Eine Fallunterscheidung ergibt

$$(a) \quad k \neq 0: \quad \omega_k \neq 1 \Rightarrow \sum_{j=0}^{N-1} \omega_k^j = 0 = N \delta_{0k},$$

$$(b) \quad k = 0: \quad \sum_{j=0}^{N-1} \omega_0^j = \sum_{j=0}^{N-1} 1 = N = N \delta_{00}.$$

(2) Zum Skalarprodukt folgt schließlich

$$(\Psi_k, \Psi_l) = (e^{ikx}, e^{ilx}) = \frac{1}{N} \sum_{j=0}^{N-1} e^{ikx_j} \overline{e^{ilx_j}} = \frac{1}{N} \sum_{j=0}^{N-1} \omega_j^k \omega_j^{-l} = \delta_{kl}.$$

□

**Satz 1.22** Bestimmung der Koeffizienten  $c_j$

Die Koeffizienten  $c_j$  zur Referenz  $(x_k, f_k)$ ,  $k = 0, 1, \dots, N-1$ , mit äquidistanten Stützstellen  $x_k$  gemäß (1.49) sind gegeben durch

$$c_j = \frac{1}{N} \sum_{k=0}^{N-1} f_k \omega_k^{-j}, \quad j = 0, 1, \dots, N-1. \quad (1.59)$$

**Beweis.**

Wegen der Eindeutigkeit der Darstellung des Interpolationspolynoms genügt es zu zeigen, dass die Funktion  $\Phi_N(x)$  genau mit diesen  $c_j$  die Interpolationsbedingung (1.55) erfüllt.

Man erhält für  $l = 0, 1, \dots, N-1$

$$\begin{aligned}
 \Phi_N(x_l) &= \sum_{j=0}^{N-1} c_j e^{ijx_l}, \quad x_l = 2\pi l/N \\
 &= \sum_{j=0}^{N-1} \left( \frac{1}{N} \sum_{k=0}^{N-1} f_k \omega_k^{-j} \right) \omega_j^l, \quad \omega_k^{-j} = \omega_j^{-k} \\
 &= \frac{1}{N} \sum_{k=0}^{N-1} \left( f_k \sum_{j=0}^{N-1} \omega_j^{-k} \omega_j^l \right) \\
 &= \frac{1}{N} \sum_{k=0}^{N-1} (f_k N \delta_{kl}) \\
 &= f_l.
 \end{aligned}$$

□

Nunmehr ist es nicht schwer, auch die Umrechnung der komplexen Koeffizienten  $c_j$  auf den reellen Fall gemäß (1.57) vorzunehmen.

Wir hatten zuvor die Koeffizienten  $c_j$  als Lösung des LGS (1.56) notiert.

Es zeigt sich, dass seine Koeffizientenmatrix einfach zu invertieren ist, denn die Berechnungsvorschrift für die  $c_j$  enthält explizit die Elemente der inversen Matrix, also

$$c = A_N f$$

mit

$$A_N = \frac{1}{N} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 & 1 \\ \omega_0^{-1} & \omega_1^{-1} & \omega_2^{-1} & \cdots & \omega_{N-2}^{-1} & \omega_{N-1}^{-1} \\ \omega_0^{-2} & \omega_1^{-2} & \omega_2^{-2} & \cdots & \omega_{N-2}^{-2} & \omega_{N-1}^{-2} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \omega_0^{-(N-2)} & \omega_1^{-(N-2)} & \omega_2^{-(N-2)} & \cdots & \omega_{N-2}^{-(N-2)} & \omega_{N-1}^{-(N-2)} \\ \omega_0^{-(N-1)} & \omega_1^{-(N-1)} & \omega_2^{-(N-1)} & \cdots & \omega_{N-2}^{-(N-1)} & \omega_{N-1}^{-(N-1)} \end{pmatrix}. \quad (1.60)$$

### 1.6.3 Diskrete Fourier-Transformation

Die Formel für die Koeffizienten

$$c_j = \frac{1}{N} \sum_{k=0}^{N-1} f_k \omega_k^{-j}, \quad j = 0, 1, \dots, N-1,$$

des trigonometrischen Interpolationspolynoms schreiben wir wiederum in Matrixform. Seien  $c = (c_0, c_1, \dots, c_{N-1})^T$  und  $f = (f_0, f_1, \dots, f_{N-1})^T$ . Dann haben wir

$$c = A_N f, \quad (A_N)_{jk} = \frac{1}{N} \omega_k^{-j}. \quad (1.61)$$

Die ausführliche Notation der Matrizen erfolgt mittels der Beziehung

$$I(N, N) = A_N A_N^{-1} = A_N^{-1} A_N =$$

$$\begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_1 & \omega_1^2 & \cdots & \omega_1^{N-1} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & \omega_{N-2} & \omega_{N-2}^2 & \cdots & \omega_{N-2}^{N-1} \\ 1 & \omega_{N-1} & \omega_{N-1}^2 & \cdots & \omega_{N-1}^{N-1} \end{pmatrix} \begin{pmatrix} \frac{1}{N} & \frac{1}{N} & \frac{1}{N} & \cdots & \frac{1}{N} \\ \frac{1}{N} & \frac{1}{N} \omega_1^{-1} & \frac{1}{N} \omega_2^{-1} & \cdots & \frac{1}{N} \omega_{N-1}^{-1} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{1}{N} & \frac{1}{N} \omega_1^{-(N-2)} & \frac{1}{N} \omega_2^{-(N-2)} & \cdots & \frac{1}{N} \omega_{N-1}^{-(N-2)} \\ \frac{1}{N} & \frac{1}{N} \omega_1^{-(N-1)} & \frac{1}{N} \omega_2^{-(N-1)} & \cdots & \frac{1}{N} \omega_{N-1}^{-(N-1)} \end{pmatrix}$$

$$(\omega_0^k = \omega_j^0 = 1).$$

Die Koeffizienten  $c_j$  nennt man auch **Fourier-Koeffizienten**.

#### Definition 1.9 Diskrete Fourier-Transformation

Die Abbildung  $\mathcal{F}_N: \mathbb{C}^N \rightarrow \mathbb{C}^N$ ,  $(f_j) \rightarrow (c_j)$ , mit  $c_j = \frac{1}{N} \sum_{k=0}^{N-1} f_k e^{-i2\pi jk/N}$

heißt *diskrete Fourier-Transformation (DFT)*.

Die Umkehrabbildung  $\mathcal{F}_N^{-1}$  mit  $f_j = \sum_{k=0}^{N-1} c_k e^{i2\pi jk/N}$  ist die **inverse DFT**.

**Bemerkung 1.5** Der Aufwand der DFT beträgt bei einer naiven Auswertung der Ausdrücke  $\mathcal{O}(N^2)$  komplexe Multiplikationen.

Ein schnellere Variante mit geschickten Faltungen und Butterfly-Algorithmus ist die schnelle Fourier-Transformation (Fast FT, FFT), welche die Komplexität von  $\mathcal{O}(N \log_2(N))$  erreicht (siehe [24]).

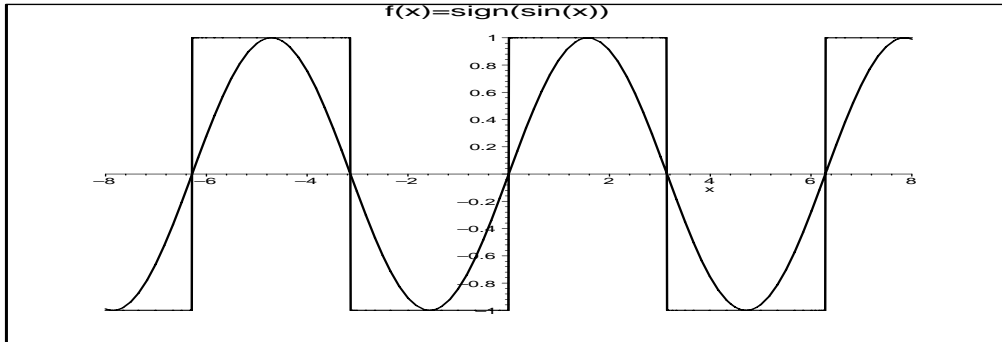
#### Beispiel 1.17 Durchführung der DFT für die Signalfunktion

Funktion und Parameter in der Aufgabe sind  $f(x) = \text{sign}(\sin(x))$ ,  $x \in [0, T]$ ,

$$T = 2\pi, \quad N = 2n = 8, \quad x_k = \frac{2\pi k}{8}, \quad k = 0, 1, \dots, 7(, 8).$$

Die Referenz ist

$x_k$	0	$\frac{\pi}{4}$	$\frac{2\pi}{4}$	$\frac{3\pi}{4}$	$\pi$	$\frac{5\pi}{4}$	$\frac{6\pi}{4}$	$\frac{7\pi}{4}$	$2\pi$
$f_k$	0	1	1	1	0	-1	-1	-1	0



**Abb. 1.12** Funktionen  $\sin(x)$  und  $f(x) = \text{sign}(\sin(x))$

Die Einheitswurzeln

$$\omega_j = e^{i2\pi j/N} = e^{i\pi j/4} = \cos(\pi j/4) + i \sin(\pi j/4), \quad j = 0, 1, \dots, N-1 = 7,$$

sowie  $\{\cos(\pi j/4), \sin(\pi j/4)\}$  sind die Ausgangswerte für die Berechnung aller weiteren Winkelfunktionen. Damit ermittelt man

$$c_j = \frac{1}{N} \sum_{k=0}^{N-1} f_k \omega_k^{-j}, \quad j = 0, 1, \dots, N-1 = 7.$$

Da  $f$  reell ist, gilt  $c_{N-j} = \overline{c_j}$ . Somit sind nur die Koeffizienten  $c_0, c_1, c_2, c_3, c_4$  zu berechnen.

Wegen der Besonderheit der Funktionswerte  $f_k$  haben wir folgende Vereinfachungen.

$$\begin{aligned} c_j &= \frac{1}{N} \sum_{k=0}^{N-1} f_k \omega_k^{-j} \\ &= \frac{1}{8} [\omega_1^{-j} + \omega_2^{-j} + \omega_3^{-j} - (\omega_5^{-j} + \omega_6^{-j} + \omega_7^{-j})], \\ \omega_k^{-j} &= e^{-i2\pi jk/8} = e^{-i\pi jk/4} = \cos(\pi jk/4) - i \sin(\pi jk/4). \end{aligned}$$

Die  $\{\cos, \sin\}$ -Werte wiederholen sich bei aufeinander folgenden  $k$  ( $\Rightarrow$  Faltungen).

$$\begin{aligned} c_0 &= \frac{1}{N} \sum_{k=0}^{N-1} f_k = 0, \\ c_1 &= \frac{1}{N} \sum_{k=0}^{N-1} f_k \omega_k^{-1} = \frac{1}{8} [\omega_1^{-1} + \omega_2^{-1} + \omega_3^{-1} - (\omega_5^{-1} + \omega_6^{-1} + \omega_7^{-1})], \end{aligned}$$

die cos-Glieder heben sich dabei weg.

$$c_1 = -\frac{i}{8} \left[ \frac{\sqrt{2}}{2} + 1 + \frac{\sqrt{2}}{2} - \left( -\frac{\sqrt{2}}{2} - 1 - \frac{\sqrt{2}}{2} \right) \right] = -i \frac{1 + \sqrt{2}}{4},$$

$$c_7 = \bar{c}_1 = i \frac{1 + \sqrt{2}}{4},$$

$$c_2 = \frac{1}{8} [\omega_1^{-2} + \omega_2^{-2} + \omega_3^{-2} - (\omega_5^{-2} + \omega_6^{-2} + \omega_7^{-2})] = 0,$$

$$\omega_k^{-2} = \cos\left(\frac{\pi k}{2}\right) - i \sin\left(\frac{\pi k}{2}\right),$$

cos / sin -Glieder sind Null oder heben sich weg,

$$c_6 = \bar{c}_2 = 0,$$

$$c_3 = \frac{1}{8} [\omega_1^{-3} + \omega_2^{-3} + \omega_3^{-3} - (\omega_5^{-3} + \omega_6^{-3} + \omega_7^{-3})] = -i \frac{-1 + \sqrt{2}}{4},$$

$$\omega_k^{-3} = \cos\left(\frac{3\pi k}{4}\right) - i \sin\left(\frac{3\pi k}{4}\right), \quad \text{cos-Glieder heben sich weg,}$$

$$c_5 = \bar{c}_3 = i \frac{-1 + \sqrt{2}}{4},$$

$$c_4 = 0.$$

Wir fassen die Ergebnisse zusammen.

- Die cos-Glieder entfallen, da die Funktion  $f(x)$  ungerade ist.
- Von den sin-Gliedern bleiben nur die mit ungeradem Index.

Die DFT ergibt das Polynom

$$\Phi_N(x) = \sum_{j=0}^{N-1} c_j e^{ijx},$$

$$\Phi_8(x) = -i \frac{1 + \sqrt{2}}{4} e^{ix} - i \frac{-1 + \sqrt{2}}{4} e^{3ix} + i \frac{-1 + \sqrt{2}}{4} e^{5ix} + i \frac{1 + \sqrt{2}}{4} e^{7ix}.$$

Die Umrechnung auf den reellen Fall führt zu

$$a_j = 2\Re(c_j) = 0, \quad a_0 = 2c_0 = 0,$$

$$b_j = -2\Im(c_j),$$

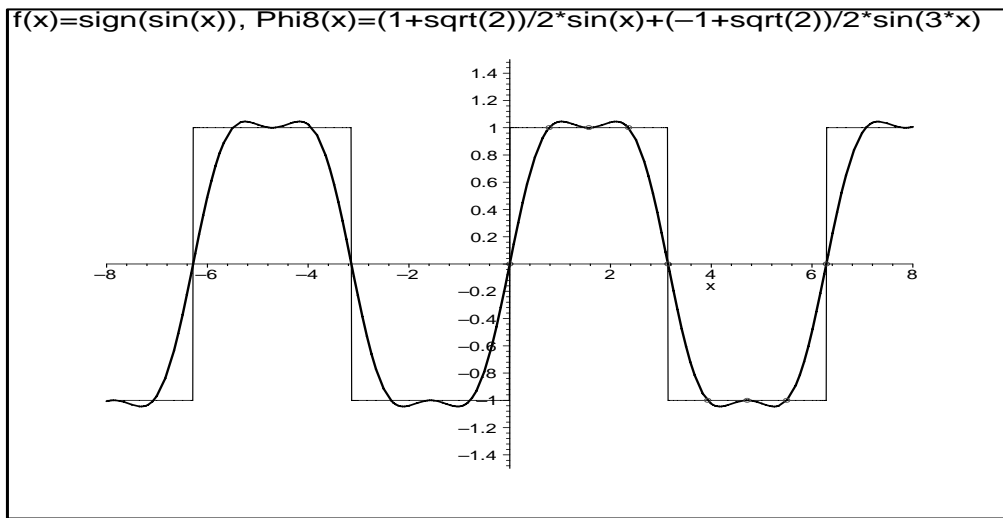
$$b_1 = \frac{1 + \sqrt{2}}{2}, \quad b_2 = 0,$$

$$b_3 = \frac{-1 + \sqrt{2}}{2}, \quad b_4 = 0,$$

und das reelle Interpolationspolynom ist

$$\Phi_{2n}(x) = \frac{a_0}{2} + \sum_{j=1}^{n-1} [a_j \cos(jx) + b_j \sin(jx)] + \frac{a_n}{2} \cos(nx),$$

$$\Phi_8(x) = \frac{1 + \sqrt{2}}{2} \sin(x) + \frac{-1 + \sqrt{2}}{2} \sin(3x).$$



**Abb. 1.13** Signalfunktion mit Referenz und Lösung der DFT

An dieser Stelle sei ein Vorgriff auf die stetige Approximation im quadratischen Mittel der Signalfunktion mit trigonometrischen Polynomen im Sinne eine möglichst guten Übereinstimmung der Flächen unter den Kurven gestattet.

Dabei wird keine Referenz vorgegeben und die exakten Fourier-Koeffizienten berechnen sich aus Integralen, die aber eventuell numerisch auszuwerten sind. Man beachte auch, dass der Begriff des Fourier-Koeffizienten sowohl im Zusammenhang mit der trigonometrischen Interpolation (DFT) als auch mit der Approximation im Mittel verwendet wird.

Das bei der Approximation entstehende reelle Fourier-Polynom  $n$ -ten Grades ist

$$p_n(x) = \frac{4}{\pi} \left( \sin(x) + \frac{1}{3} \sin(3x) + \frac{1}{5} \sin(5x) + \dots + \frac{1}{2n+1} \sin((2n+1)x) \right).$$

Ein Vergleich der ersten Koeffizienten von  $p_n(x)$  und  $\Phi_8(x)$  ergibt das Folgende.

$$\begin{aligned} \text{Koeffizient bei } \sin(x): \quad 1.273\,239 &= \frac{4}{\pi} \approx \frac{1 + \sqrt{2}}{2} = 1.207\,107, \\ \text{Koeffizient bei } \sin(3x): \quad 0.424\,413 &= \frac{4}{3\pi} \approx \frac{-1 + \sqrt{2}}{2} = 0.207\,107, \\ \text{Koeffizient bei } \sin(5x): \quad 0.254\,648 &= \frac{4}{5\pi}. \end{aligned}$$

Nimmt man entsprechend mehr Stützstellen, so liegt die Lösung  $\Phi_N(x)$  immer näher bei  $p_n(x)$ , was für  $N = 16, 32, 64, 128$  zumindest bei den ersten drei Koeffizienten gut zu erkennen ist. Die genannten Lösungen der DFT sind

$$\Phi_{16}(x) = 1.256835 \sin(x) + 0.374151 \sin(3x) + 0.167045 \sin(5x) + 0.049728 \sin(7x),$$

$$\Phi_{32}(x) = 1.269146 \sin(x) + 0.412070 \sin(3x) + 0.233859 \sin(5x) + \dots + b_{15} \sin(15x),$$

$$\Phi_{64}(x) = 1.272217 \sin(x) + 0.421341 \sin(3x) + 0.249514 \sin(5x) + \dots + b_{31} \sin(31x),$$

$$\Phi_{128}(x) = 1.272984 \sin(x) + 0.423646 \sin(3x) + 0.253368 \sin(5x) + \dots + b_{63} \sin(63x).$$

Dazu geben wir noch einige vergleichende grafische Darstellungen und anschließend die zugehörigen Anweisungen in Maple an.

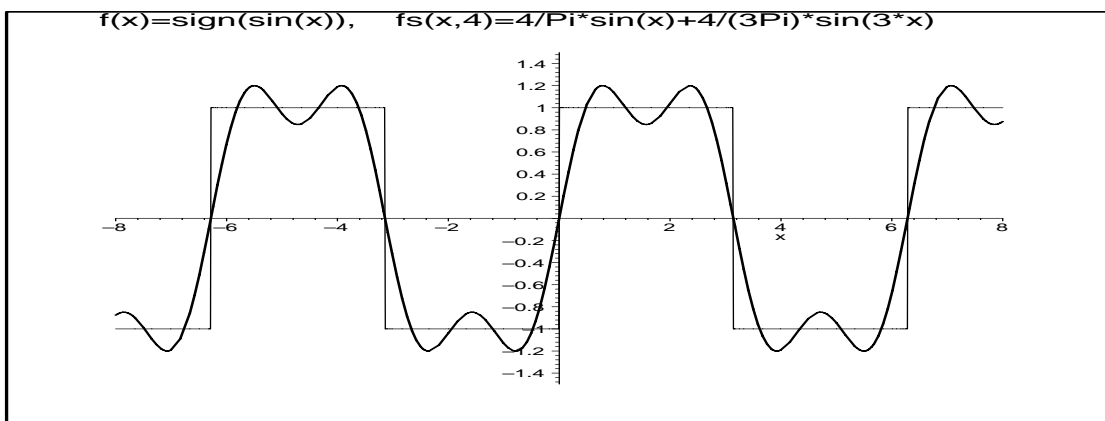


Abb. 1.14 Signalfunktion  $f(x)$  mit Fourier-Polynom  $p_1(x) = fs(x, 4)$

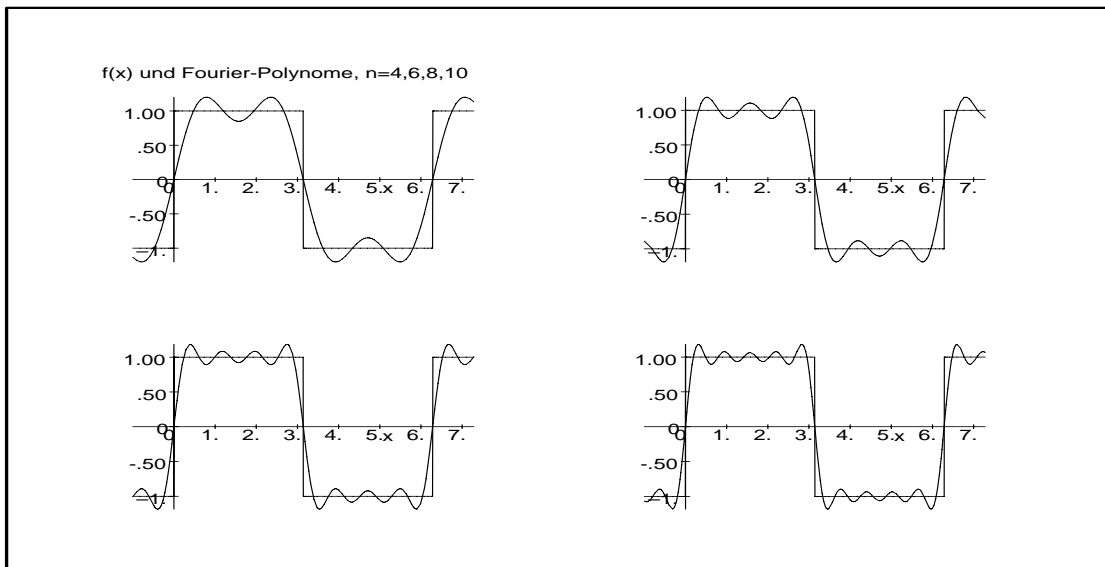


Abb. 1.15 Signalfunktion  $f(x)$  mit Fourier-Polynomen  $p_n(x)$ ,  $n = 1, 2, 3, 4$

Man beachte, dass in den Abbildungen  $p_n(x) = fs(x, 2n + 2)$  ist.

Wiederum erkennt man das Überspringen der Fourier-Polynome an den Sprungstellen der Signalfunktion, d. h. den Gibbs'schen Effekt.

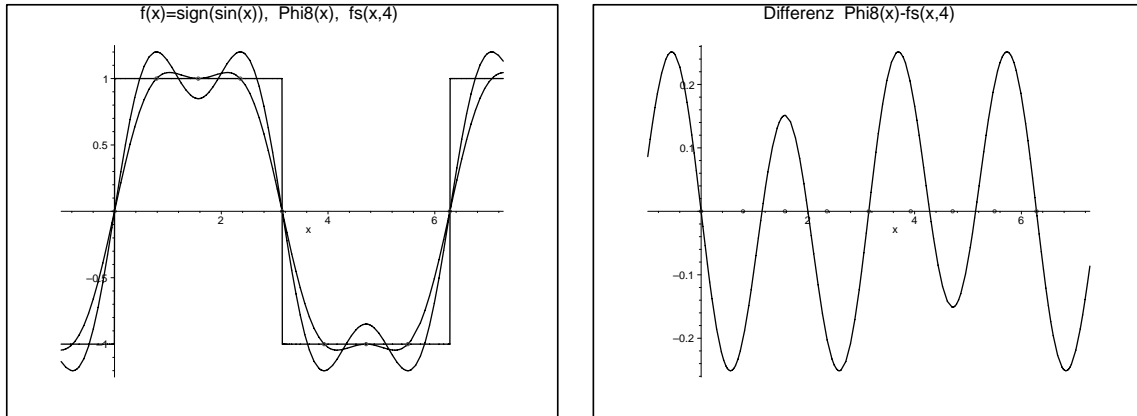


Abb. 1.16  $f(x)$  mit  $\Phi_8(x)$  und  $p_1(x)$

Differenz  $\Phi_8(x) - p_1(x)$

### Maple-Anweisungen

```
# Polynome, Interpolation, Splines, Differentiation
# nm6ip6.mws
#
# Trigonometrische Interpolation mit aequidistanten Stuetzstellen
# (DFT) und Fourier-Entwicklung (Fourier-Polynome, Approximation im
# quadratischen Mittel)
# fuer Signalfunktion f(x) mit Grafik
#
> restart:
> with(plots):
> with(plottools):
> # Arbeitsverzeichnis
> pfad := 'D:/Neundorf/Nwptexte/Math93/':
>
> # Graphik output files
> name1 := 'nm6ip11.ps':
> date1 := cat(pfad,name1):
> .....

# m-File fuer Fourier-Reihe
> read 'D:/Neundorf/Nwptexte/Math93/cfourier.m';

# Parameter
> a:=-Pi: b:=Pi: T:=b-a: N:=8:
```



```
> u1:=x->piecewise(x=-Pi,0,x<0,-1,x>0,1,0):
> u1(x);
```

```

{ 0          x = -Pi
{
{ -1         x < 0
{
{ 1          0 < x
{
{ 0          otherwise
```

```
> f:=x->u1(x-T*floor((x-a)/T)):
> f(x);
> Q1:=plot([f(x),sin(x)],x=-8..8,-1..1,thickness=2,
>         title='f(x)=sign(sin(x))'):
> Q2:=plot(f(x),x=-8..8,-1.5..1.5,thickness=1):

> plots[display](Q1); # -> datei1,
```

```

{
{ 0          x - 2 Pi floor(1/2 -----) = -Pi
{
{
{
{
{ -1         x - 2 Pi floor(1/2 -----) < 0
{
{
{
{
{ 1          0 < x - 2 Pi floor(1/2 -----)
{
{
{
{ 0          otherwise
```

```
> xp:=[seq(k*T/N,k=0..N)];
> yp:=[seq(f(k*T/N),k=0..N)];
> ref:=[seq([xp[k],yp[k]],k=1..N+1)];
> ref0:=[seq([xp[k],0],k=1..N+1)];
> Q3:=plot(ref,style=point,symbol=circle):
> Q30:=plot(ref0,style=point,symbol=circle):
```

```

xp := [0, 1/4 Pi, 1/2 Pi, 3/4 Pi, Pi, 5/4 Pi, 3/2 Pi, 7/4 Pi, 2 Pi]
yp := [0, 1, 1, 1, 0, -1, -1, -1, 0]

ref := [[0, 0], [1/4 Pi, 1], [1/2 Pi, 1], [3/4 Pi, 1], [Pi, 0],
        [5/4 Pi, -1], [3/2 Pi, -1], [7/4 Pi, -1], [2 Pi, 0]]
ref0 := [[0, 0], [1/4 Pi, 0], [1/2 Pi, 0], [3/4 Pi, 0], [Pi, 0],
        [5/4 Pi, 0], [3/2 Pi, 0], [7/4 Pi, 0], [2 Pi, 0]]

> PhiN:= x -> (1+sqrt(2))/2*sin(x)+(-1+sqrt(2))/2*sin(3*x):
> Q:=plot(PhiN(x),x=-8..8,-1.5..1.5,thickness=1,
>         title='f(x)=sign(sin(x)),
>         PhiN(x)=(1+sqrt(2))/2*sin(x)+(-1+sqrt(2))/2*sin(3*x),N=8'):
> plots[display]([Q,Q2,Q3]); # -> datei2,

# Fourier-Polynom
# Approximation im quadratischen Mittel

> cfourier(u1,a..b,N,'reell','Plot');

Reelle Fourierkoeffizienten:
                                {      0      n = 0
                                {
a[n] = {      n = 0      , b[n] = {      n
                                {      (-1)  - 1
                                {      -2 ----- otherwise
                                {      n Pi

Reelle Fourierreihe:
sin(t)      sin(3 t)      sin(5 t)      sin(7 t)
4 ----- + 4/3 ----- + 4/5 ----- + 4/7 -----, +...
Pi          Pi          Pi          Pi

```

```

> bk:= k -> 2*(1-(-1)^k)/(Pi*k):

# Fourier-Polynom, explizit
> n:='n':
> fs:=(x,n)-> sum(bk(k)*sin(k*x),k=1..n):

> fs(x,n);
> fs(x,4);

```

```
> Q:=plot(fs(x,4),x=-8..8,-1.5..1.5,thickness=2,
> title='f(x)=sign(sin(x)), fs(x,4)=4/Pi*sin(x)+4/(3Pi)*sin(3*x)':
> plots[display]([Q,Q2]); # -> datei3
```

$$\frac{n}{k=1} \frac{(1 - (-1)^k) \sin(kx)}{2 \pi^k}$$

$$\frac{\sin(x)}{4 \pi} + \frac{\sin(3x)}{4/3 \pi}$$

```
> P:=array(1..2,1..2,[]):
> n:=2:
> for i from 1 by 1 to 2 do
>   for j from 1 by 1 to 2 do
>     n:=n+2:
>     if i*j=1 then
>       P[i,j]:=plot([f(x),fs(x,n)],x=-1..T+1,color=black,
>         title='f(x) und Fourier-Polynome, n=4,6,8,10')
>     else
>       P[i,j]:=plot([f(x),fs(x,n)],x=-1..T+1,color=black)
>     fi:
>   od:
> od:

> plots[display](P); # -> datei4

> Q:=plot([f(x),PhiN(x),fs(x,4)],x=-1..T+1,color=black,
>   title='f(x)=sign(sin(x)), Phi8(x), fs(x,4)':
> plots[display]([Q,Q3]); # -> datei5

> Q:=plot((PhiN(x)-fs(x,4)),x=-1..T+1,color=black,
>   title='Differenz Phi8(x)-fs(x,4)':
> plots[display]([Q,Q30]); # -> datei6
```

## 1.7 Splineinterpolation im $\mathbb{R}^1$

Die Konstruktion von Splines beruht auf einer intervallweisen Interpolation, wo man stückweise Polynome niedrigen Grades zu einer glatten Gesamtfunktion zusammensetzt. Die Ausgangssituation wird durch die folgenden Aspekte beschrieben.

- Grundintervall  $I = [a, b] \subset \mathbb{R}$  mit  $-\infty < a < b < \infty$  und (bekannte oder unbekannte) reelle Funktion  $f : I \rightarrow \mathbb{R}$ .
- Gegeben sind Stützstellen  $x_k$ , Stützwerte  $y_k$  und Schrittweiten  $h_k = x_{k+1} - x_k > 0$  sowie  $R_0$  als **Referenz** mit  $n + 1$  paarweise verschiedenen **Stützstellen** und den  $n + 1$  zugehörigen **Stützwerten**. Falls eine Funktion  $f(x)$  zu Grunde liegt, definiert man die Stützwerte i. Allg. gemäß  $y_k = f(x_k)$ ,  $k = 0, 1, \dots, n$ .
- Man konstruiert die Splinefunktion als zusammengesetztes Polynom  $s(x) = s(x, R_0)$  vom Grad  $m$  ( $m \geq 1$ ) mit den folgenden Eigenschaften.
  - (a)  $s(x)$  ist ein Polynom vom Grad  $\leq m$  auf jedem der Teilintervalle, d. h.  $s(x) \in \mathcal{S}_m(R_0)$  und

$$s(x) = s^{(k)}(x) = \alpha_{k0} + \alpha_{k1}x + \dots + \alpha_{km}x^m, \quad x \in [x_k, x_{k+1}], \quad (1.62)$$

- (b) Bezüglich der Glattheit fordert man  $s(x) \in \mathcal{C}^{m-1}(I)$ .

- Interpolationsforderung (Interpolationsbedingung)

$$(1) \quad s(x_k) = y_k = f(x_k), \quad k = 0, 1, \dots, n. \quad (1.63)$$

- (2) An den inneren Punkten  $x_1, x_2, \dots, x_{n-1}$  ist  $s(x)$  stetig differenzierbar bis zur Ordnung  $m - 1$ , d. h., für  $k = 1, 2, \dots, n - 1$  gelten

$$\begin{aligned} s^{(k-1)}(x_k) &= s^{(k)}(x_k), \\ s^{(k-1)}(x_k)' &= s^{(k)}(x_k)', \\ &\vdots \\ s^{(k-1)}(x_k)^{(m-1)} &= s^{(k)}(x_k)^{(m-1)}. \end{aligned} \quad (1.64)$$

- **Interpolationsaufgabe (IA4)**

Gesucht sind auf den  $n$  Teilintervallen insgesamt  $n(m + 1)$  Koeffizienten  $\alpha_{kl}$ ,  $k = 0, 1, \dots, n - 1$ ,  $l = 0, 1, \dots, m$ , so dass die Interpolationsforderung erfüllt ist.

Zunächst stellt man fest, dass man für die  $n(m + 1) = n + 1 + mn - 1$  Unbekannten nur  $n + 1 + m(n - 1) = n + 1 + mn - m$  Interpolationsbedingungen (1)+(2) zur Verfügung hat. Damit besitzt das Polynom  $m - 1$  freie Parameter. Zwecks Eindeutigkeit sind diese durch weitere Bedingungen zu binden. Im Einzelnen stellt sich die Situation so dar:

- $m = 1$ : kein freier Parameter, eindeutige Lösung als Polygonzug, linearer Spline,
- $m = 2$ : 1 freier Parameter, quadratische Splines,
- $m = 3$ : 2 freie Parameter, kubische Splines.

### 1.7.1 Einfache Typen von Splines

#### (1) Linearer Spline

Für  $x \in [x_k, x_{k+1}]$  macht man den Ansatz  $s^{(k)}(x) = a_k + b_k(x - x_k)$ ,  $k = 0, 1, \dots, n-1$ .

Die  $2n$  Bedingungen sind

$$\begin{aligned} s^{(k)}(x_k) &= f_k, \quad k = 0, 1, \dots, n-1, \quad s^{(n-1)}(x_n) = f_n, \\ s^{(k)}(x_k) &= s^{(k-1)}(x_k), \quad k = 1, 2, \dots, n-1. \end{aligned} \quad (1.65)$$

Das Ergebnis auf dem Teilintervall ist die Newtonsche bzw. Lagrangesche Form

$$\begin{aligned} s^{(k)}(x) &= f_k + \frac{f_{k+1} - f_k}{x_{k+1} - x_k}(x - x_k), \\ &= \frac{x_{k+1} - x}{x_{k+1} - x_k} f_k + \frac{x - x_k}{x_{k+1} - x_k} f_{k+1}, \quad k = 0, 1, \dots, n-1. \end{aligned} \quad (1.66)$$

#### (2) Quadratische Splines

Für  $x \in [x_k, x_{k+1}]$  sei  $s^{(k)}(x) = a_k + b_k(x - x_k) + c_k(x - x_k)^2$ ,  $k = 0, 1, \dots, n-1$ .

Die  $3n$  Bedingungen sind

$$\begin{aligned} s^{(k)}(x_k) &= f_k, \quad k = 0, 1, \dots, n-1, \quad s^{(n-1)}(x_n) = f_n, \\ s^{(k)}(x_k) &= s^{(k-1)}(x_k), \quad k = 1, 2, \dots, n-1, \\ s^{(k)}(x_k)' &= s^{(k-1)}(x_k)', \quad k = 1, 2, \dots, n-1, \\ s^{(0)}(x_0)' &= m_0 \quad (m_0 \text{ gegeben oder approximiert}). \end{aligned} \quad (1.67)$$

An Stelle der letzten Bedingung  $s^{(0)}(x_0)' = m_0$  sind auch andere möglich. Sie werden auch Endbedingungen genannt, falls sie am Ende des untersuchten Bereichs definiert werden. Andere Zusatzbedingungen sind

- $s^{(k)}(\bar{x}) = \bar{y}$ ,  $\bar{x}$  ist eine zusätzliche Stelle,
- $s^{(n-1)}(x_n)' = m_n$ ,
- $s^{(0)}(x_0)' = s^{(n-1)}(x_n)'$ , Periodizität verbunden mit  $f_0 = f_n$ ,
- $s^{(0)}(x_0)' = -s^{(n-1)}(x_n)'$ , Antiperiodizität verbunden mit  $f_0 = f_n$ ,
- $K(s) = \int_{x_0}^{x_n} \omega(x) [s''(x)]^2 dx \rightarrow \min$ ,  $\omega(x) > 0$  gegebene Gewichtsfunktion,

damit wird die Gesamtkrümmung der Kurve minimal.

Untersuchungen zur Approximationsgüte linearer und quadratischer Splines findet man in [30].

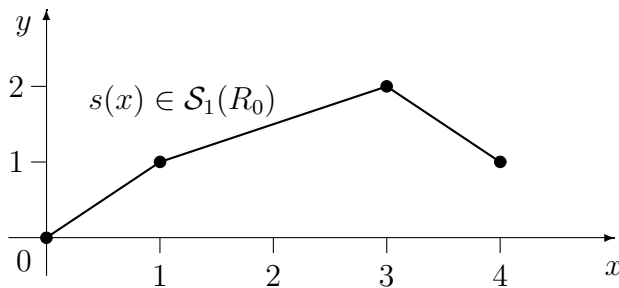
**Beispiel 1.18** Gegeben sei die Referenz  $R_0$  mit 4 Knoten ( $n = 3$ )

$x_k$	0	1	3	4
$y_k$	0	1	2	1

(a) Linearer Spline

Die stetige aus Geradenstücken zusammengesetzte Funktion ist

$$s(x) = \begin{cases} x & \text{für } x \in [0, 1], \\ 1 + \frac{1}{2}(x - 1) & \text{für } x \in [1, 3], \\ 2 - (x - 3) & \text{für } x \in [3, 4]. \end{cases}$$



**Abb. 1.17** Lineare Splinefunktion

(b) Quadratischer Spline

Wir suchen die aus Parabelstücken zusammengesetzte differenzierbare Funktion.

Als modifizierten Ansatz für das Teilintervall  $[x_k, x_{k+1}]$  nehmen wir

$$s^{(k)}(x) = a_k + b_k x + c_k x^2, \quad k = 0, 1, \dots, n-1. \quad (1.68)$$

Der Ansatz für das erste Parabelstück der Kurve ist  $s^{(0)}(x) = a_0 + b_0 x + c_0 x^2$ . Aus den Bedingungen  $s^{(0)}(0) = 0$  und  $s^{(0)}(1) = 1$  folgt  $a_0 = 0$  und  $b_0 = 1 - c_0$ . Damit ist

$$s^{(0)}(x) = (1 - c_0)x + c_0 x^2.$$

Außerdem gilt  $\frac{d}{dx}s^{(0)}(1) = 1 + c_0$ .

Der Ansatz für das zweite Parabelstück  $s^{(1)}(x) = a_1 + b_1 x + c_1 x^2$  führt mit den Bedingungen  $s^{(1)}(1) = 1$ ,  $s^{(1)}(3) = 2$  und  $\frac{d}{dx}s^{(1)}(1) = 1 + c_0$  auf die Teillösung

$$s^{(1)}(x) = -(0.25 + 1.5c_0) + (1.5 + 2c_0)x - (0.25 + 0.5c_0)x^2.$$

Es ist  $\frac{d}{dx}s^{(1)}(3) = -c_0$ .

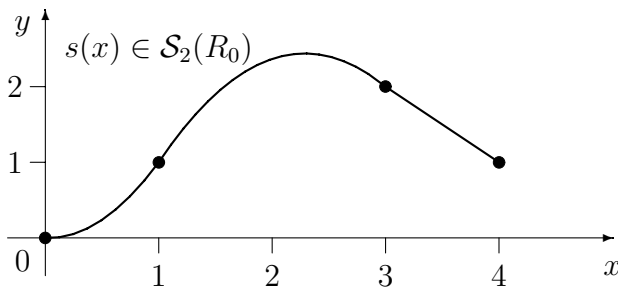
Mit dem Ansatz für das dritte Parabelstück  $s^{(2)}(x) = a_2 + b_2 x + c_2 x^2$  und den Bedingungen  $s^{(2)}(3) = 2$ ,  $s^{(2)}(4) = 1$  und  $\frac{d}{dx}s^{(2)}(3) = -c_0$  erhalten wir die Lösung

$$s^{(2)}(x) = (-7 + 12c_0) + (6 - 7c_0)x + (-1 + c_0)x^2.$$

Damit die Gesamtlösung eindeutig wird, muss eine weitere Bedingung gestellt werden, z. B.  $\frac{d}{dx}s^{(0)}(0) = 0$ . Dann ist  $c_0 = 1$  und

$$s(x) = \begin{cases} x^2 & \text{für } x \in [0, 1], \\ -\frac{7}{4} + \frac{7}{2}x - \frac{3}{4}x^2 & \text{für } x \in [1, 3], \\ 5 - x & \text{für } x \in [3, 4]. \end{cases}$$

Die einzelnen Teilpolynome  $s^{(k)}$ ,  $k = 0, 1, 2$ , können bei einer zusätzlichen Bedingung am linken Rand der Reihe nach vollständig berechnet werden.



**Abb. 1.18** Quadratische Splinefunktion

Zum selben Ergebnis gelangt man auch mit dem Ansatz

$$s^{(k)}(x) = a_k + b_k(x - x_k) + c_k(x - x_k)^2, \quad k = 0, 1, \dots, n-1, \quad (1.69)$$

für das Teilintervall  $[x_k, x_{k+1}]$ , der sofort die Lösung für den Koeffizienten  $a_k$  nach sich zieht, nämlich  $a_k = f_k$ .

Außerdem vereinfacht die Darstellung der Ableitung  $s^{(k)}(x)' = b_k + 2c_k(x - x_k)$  die Berechnung von  $s^{(k)}(x_k)' = b_k$ ,  $s^{(k)}(x_{k+1})' = b_k + 2c_k h_k$ . Wir erhalten somit die nicht ausmultiplizierte Form der Splinefunktion

$$s(x) = \begin{cases} 0 + 0(x - 0) + 1(x - 0)^2 & \text{für } x \in [0, 1], \\ 1 + 2(x - 1) - \frac{3}{4}(x - 1)^2 & \text{für } x \in [1, 3], \\ 2 - 1(x - 3) + 0(x - 3)^2 & \text{für } x \in [3, 4]. \end{cases}$$

### Zwei Varianten zur Bestimmung der quadratischen Splinefunktion

(1) Man bestimmt die Splinekoeffizienten als Lösung eines LGS.

Wir verwenden den Ansatz mit der Normalform des Polynoms 2. Grades

$$\begin{aligned} s^{(k)}(x) &= a_k + b_k x + c_k x^2, \quad x \in [x_k, x_{k+1}], \quad k = 0, 1, \dots, n-1, \\ s^{(k)}(x)' &= b_k + 2c_k x. \end{aligned}$$

Die Zusammenfassung der Interpolationsbedingungen ergibt die folgende Übersicht.

Bedingungen	Anzahl
$s^{(0)}(x_0)' = m_0$	1
$s^{(k)}(x_k) = f_k, \quad k = 0, 1, \dots, n-1$	$n$
$s^{(k)}(x_{k+1}) = f_{k+1}, \quad k = 0, 1, \dots, n-1$	$n$
$s^{(k-1)}(x_k)' = s^{(k)}(x_k)', \quad k = 1, 2, \dots, n-1$	$n-1$
	$\Sigma = 3n$

Somit entsteht ein reguläres LGS  $A\alpha = \beta$  mit Blockstruktur, im Fall  $n = 4$  ist das

$$\begin{pmatrix} 0 & 1 & 2x_0 & & & & & & & 0 \\ 1 & x_0 & x_0^2 & & & & & & & \\ 1 & x_1 & x_1^2 & & & & & & & \\ 0 & 1 & 2x_1 & 0 & -1 & -2x_1 & & & & \\ & & & 1 & x_1 & x_1^2 & & & & \\ & & & 1 & x_2 & x_2^2 & & & & \\ & & & 0 & 1 & 2x_2 & 0 & -1 & -2x_2 & \\ & & & & & & 1 & x_2 & x_2^2 & \\ & & & & & & 1 & x_3 & x_3^2 & \\ & & & & & & 0 & 1 & 2x_3 & 0 & -1 & -2x_3 \\ & & & & & & & & & 1 & x_3 & x_3^2 \\ 0 & & & & & & & & & 1 & x_4 & x_4^2 \end{pmatrix} \begin{pmatrix} a_0 \\ b_0 \\ c_0 \\ a_1 \\ b_1 \\ c_1 \\ a_2 \\ b_2 \\ c_2 \\ a_3 \\ b_3 \\ c_3 \end{pmatrix} = \begin{pmatrix} m_0 \\ f_0 \\ f_1 \\ 0 \\ f_1 \\ f_2 \\ 0 \\ f_2 \\ f_3 \\ 0 \\ f_3 \\ f_4 \end{pmatrix}.$$

(2) Die sukzessive Berechnung als zweite Möglichkeit wurde schon genannt.

Man führt die zusätzlichen Unbekannten  $s'(x_k) = d_k$  ein. Damit haben die Interpolationsbedingungen für das Teilpolynom  $s^{(k)}(x) = a_k + b_k x + c_k x^2$  die Gestalt

$$\begin{aligned} s^{(k)}(x_j) &= f_j, \quad j = k, k+1, \\ s^{(k)}(x_k)' &= f'_k = d_k. \end{aligned}$$

Die Transformation des Intervalls  $[x_k, x_{k+1}]$  auf das Standardbezugsintervall  $[0, 1]$  ergibt

$$\begin{aligned} h_k &= x_{k+1} - x_k, \quad x = x_k + th_k, \quad t \in [0, 1], \\ s^{(k)}(x) &= s^{(k)}(x_k + th_k) = q^{(k)}(t), \\ q^{(k)}(t) &= \tilde{a}_k + \tilde{b}_k t + \tilde{c}_k t^2, \quad q^{(k)}(t)' = \tilde{b}_k + 2\tilde{c}_k t, \\ s^{(k)}(x_k) &= q^{(k)}(0) = \tilde{a}_k, \\ s^{(k)}(x_{k+1}) &= q^{(k)}(1) = \tilde{a}_k + \tilde{b}_k + \tilde{c}_k, \end{aligned}$$



$$s^{(k)}(x_k)' = \left. \frac{ds^{(k)}(x)}{dx} \right|_{x=x_k} = \left. \frac{ds^{(k)}(x_k + th_k)}{dt} \right|_{t=0} \frac{dt}{dx} = \left. \frac{dq^{(k)}(t)}{dt} \right|_{t=0} \frac{1}{h_k} = \frac{q^{(k)}(0)'}{h_k} = \frac{\tilde{b}_k}{h_k}.$$

Die Koeffizienten von  $q^{(k)}(t)$  ergeben sich aus den Bedingungen

$$\begin{aligned} \tilde{a}_k &= f_k, \\ \tilde{b}_k &= h_k f'_k = h_k d_k, \\ \tilde{a}_k + \tilde{b}_k + \tilde{c}_k &= f_{k+1} \Rightarrow \tilde{c}_k = f_{k+1} - f_k - h_k d_k, \end{aligned}$$

die wir nun in der Darstellung von  $s^{(k)}(x)$  anwenden.

$$\begin{aligned} s^{(k)}(x) &= s^{(k)}(x_k + th_k) = q^{(k)}(t) \\ &= f_k + h_k d_k t + (f_{k+1} - f_k - h_k d_k) t^2 \\ &= f_k + h_k d_k t + h_k \left( \underbrace{\frac{f_{k+1} - f_k}{h_k}}_{g_k \text{ (Steigung)}} - d_k \right) t^2 \\ &= f_k + h_k d_k t + h_k (g_k - d_k) t^2. \end{aligned}$$

Die Anwendung der Stetigkeitsbedingungen für die 1. Ableitung

$$s^{(k-1)}(x_k)' = s^{(k)}(x_k)', \quad k = 1, 2, \dots, n-1,$$

liefert

$$\begin{aligned} s^{(k-1)}(x_{k-1} + 1 \cdot h_{k-1})' &= s^{(k)}(x_k + 0 \cdot h_k)', \\ \frac{q^{(k-1)}(1)'}{h_{k-1}} &= \frac{q^{(k)}(0)'}{h_k}, \\ \frac{\tilde{b}_{k-1} + 2\tilde{c}_{k-1}}{h_{k-1}} &= \frac{\tilde{b}_k}{h_k}. \end{aligned}$$

Das Einsetzen von  $\tilde{b}_k$ ,  $\tilde{b}_{k-1}$ ,  $\tilde{c}_{k-1}$  führt zu

$$\begin{aligned} \frac{1}{h_{k-1}} [h_{k-1} d_{k-1} + 2h_{k-1} (g_{k-1} - d_{k-1})] &= \frac{h_k d_k}{h_k}, \\ d_{k-1} + 2(g_{k-1} - d_{k-1}) &= d_k, \\ d_{k-1} + d_k &= 2g_{k-1}, \quad k = 1, 2, \dots, n-1. \end{aligned}$$

Jetzt brauchen wir die Zusatzbedingung, die gemäß  $s^{(0)}(x_0)' = f'_0 = d_0$  den Wert  $d_0$  bereitstellt. Damit können mit der aufsteigenden Rekursion

$$d_k = 2g_{k-1} - d_{k-1}, \quad k = 1, 2, \dots, n-1,$$

die fehlenden Größen  $d_k$  ermittelt werden. Endlich haben wir

$$\begin{aligned} s^{(k)}(x) &= f_k + h_k d_k t + (g_k - d_k) h_k t^2 \quad \text{mit } t = \frac{x - x_k}{h_k} \\ &= f_k + d_k(x - x_k) + \frac{g_k - d_k}{h_k} (x - x_k)^2. \end{aligned}$$

Das Ergebnis zeigt große Ähnlichkeit zur Newton-Interpolationsformel bei gleichen Stützstellen und mit der 1. und 2. Steigung, die vergleichbar mit  $d_k$  bzw.  $(g_k - d_k)/h_k$  sind.

Andere Zusatzbedingungen sind auf ähnliche Weise zu behandeln, wobei bei Periodizität und Antiperiodizität mit  $d_0 = \pm d_n$  für die Lösung der Interpolationsaufgabe eine Fallunterscheidung bez.  $n$  gerade/ungerade erforderlich ist, aber das eventuell zu lösende LGS auch eine einfache Struktur hat.

## 1.7.2 Kubische Splines

### Definition 1.10 Natürliche kubische Splinefunktion

Eine natürliche kubische Splinefunktion  $s(x)$  zur Referenz  $R_0$  mit  $y_k = f_k = f(x_k)$  ist eine reelle Funktion mit den folgenden drei Eigenschaften.

- (a)  $s(x)$  ist in jedem Teilintervall  $[x_k, x_{k+1}]$ ,  $k = 0, 1, \dots, n-1$ , ein Polynom höchstens 3. Grades.
- (b)  $s(x)$  ist in den Intervallen  $(-\infty, x_0)$  und  $(x_n, \infty)$  ein Polynom 1. Grades. Das heißt, dass die Krümmung von  $s(x)$  an den Stellen  $x_0$  und  $x_n$  Null ist.
- (c)  $s(x), s'(x), s''(x)$  sind stetig in  $\mathbb{R}$  und  $s(x)$  interpoliert  $f(x)$  an den  $n+1$  Stützstellen  $x_k$ .

### Darstellung des Splines $s(x)$

Sei  $s(x) = s^{(k)}(x)$  für  $x \in [x_k, x_{k+1}]$  mit

$$s^{(k)}(x) = a_k + b_k(x - x_k) + c_k(x - x_k)^2 + d_k(x - x_k)^3, \quad k = 0, 1, \dots, n-1. \quad (1.70)$$

Die Formulierung aller Bedingungen mit  $f_k = f(x_k)$  ergibt

$$\begin{aligned} (a) \quad s^{(k)}(x_k) &= f_k, & k &= 0, 1, \dots, n, \\ (b) \quad s^{(k)}(x_k) &= s^{(k-1)}(x_k), & k &= 1, 2, \dots, n, \\ s^{(k)}(x_k)' &= s^{(k-1)}(x_k)', & k &= 1, 2, \dots, n, \\ s^{(k)}(x_k)'' &= s^{(k-1)}(x_k)'', & k &= 1, 2, \dots, n, \\ (c) \quad s^{(0)}(x_0)'' &= 0 & (2 \text{ Zusatzbedingungen}), \\ s^{(n)}(x_n)'' &= 0, \end{aligned} \quad (1.71)$$

mit der zusätzlichen Funktion auf  $[x_n, \infty)$

$$s^{(n)}(x) = a_n + b_n(x - x_n) + c_n(x - x_n)^2. \quad (1.72)$$

Die Funktion  $s^{(n)}(x)$  wurde künstlich hinzugefügt, ohne die Aufgabenstellung zu verändern, so dass die Anzahl der unbekannten Koeffizienten gleich der Anzahl der Bedingungen  $4n + 3$  beträgt.

Weitere Typen kubischer Splines werden nur kurz charakterisiert.

- Eingespannter Spline (clamped spline)

$$s'(x_0) = m_0, \quad s'(x_n) = m_n, \quad \text{wobei } m_0, m_n \text{ gegeben sind.} \quad (1.73)$$

- Periodischer Spline

$$s'(x_0) = s'(x_n), \quad s''(x_0) = s''(x_n), \quad (1.74)$$

wobei für die Periodizität  $f_0 = f_n$  sinnvoll ist.

- Spline mit Not-a-knot-Bedingung

$s(x)$  ist auf  $[x_0, x_1]$  und  $[x_1, x_2]$  sowie  $[x_{n-2}, x_{n-1}]$  und  $[x_{n-1}, x_n]$  identisch.

Damit erweisen sich die Knoten  $x_1$  und  $x_{n-1}$  als überflüssig (keine eigentlichen Knoten).

### Berechnung der Koeffizienten des natürlichen kubischen Splines

Die einzelnen Berechnungsschritte werden zunächst kompakt zusammengestellt und nachfolgend begründet.

1. Rechte Seiten der Bestimmungsgleichungen

Seien  $h_k = x_{k+1} - x_k$  die Schrittweiten und

$$e_k = 3 \left( \frac{f_{k+1} - f_k}{h_k} - \frac{f_k - f_{k-1}}{h_{k-1}} \right), \quad k = 1, 2, \dots, n-1. \quad (1.75)$$

2. Das System der Bestimmungsgleichungen für die Größen  $c_k$  ist

$$h_{k-1}c_{k-1} + 2(h_{k-1} + h_k)c_k + h_k c_{k+1} = e_k, \quad k = 1, 2, \dots, n-1, \quad (1.76)$$

wobei  $c_0 = c_n = 0$ .

3. Die restlichen Splinekoeffizienten ergeben sich zu

$$\begin{aligned} a_k &= f_k, \quad k = 0, 1, \dots, n, \\ b_k &= \frac{1}{h_k}(f_{k+1} - f_k) - \frac{1}{3}(2c_k + c_{k+1})h_k, \quad k = 0, 1, \dots, n-1, \\ d_k &= \frac{1}{3h_k}(c_{k+1} - c_k), \quad k = 0, 1, \dots, n-1. \end{aligned} \quad (1.77)$$

## 4. Algorithmus zur Lösung der Bestimmungsgleichungen (1.76)

Dies ist ein LGS mit einer diagonaldominanten Tridiagonalmatrix, das mit einer speziellen Variante des Gauß-Algorithmus gelöst werden kann.

$$\begin{aligned}
 (1) \quad \gamma_0 &= 1, \quad \gamma_1 = 2(h_0 + h_1), \quad g_1 = e_1, \\
 (2) \quad \gamma_k &= 2(h_{k-1} + h_k) - \frac{h_{k-1}^2}{\gamma_{k-1}}, \\
 g_k &= e_k - \frac{h_{k-1}}{\gamma_{k-1}} g_{k-1}, \quad k = 2, 3, \dots, n-1, \\
 (3) \quad \gamma_n &= 1, \quad g_n = 0, \\
 (4) \quad c_n &= 0, \\
 c_k &= \frac{1}{\gamma_k} (g_k - h_k c_{k+1}), \quad k = n-1, n-2, \dots, 1, \\
 c_0 &= 0.
 \end{aligned} \tag{1.78}$$

**Herleitung der Beziehungen für die Koeffizienten  $a_k, b_k, c_k, d_k$** 

- Einsetzen von (1.70) in die Interpolationsbedingungen (1.71) (a) – (c)

$$\begin{aligned}
 (a) \quad a_k &= f_k, \quad k = 0, 1, \dots, n, \\
 (b1) \quad a_k &= a_{k-1} + b_{k-1} h_{k-1} + c_{k-1} h_{k-1}^2 + d_{k-1} h_{k-1}^3, \quad k = 1, 2, \dots, n, \\
 (b2) \quad b_k &= b_{k-1} + 2c_{k-1} h_{k-1} + 3d_{k-1} h_{k-1}^2, \\
 (b3) \quad 2c_k &= 2c_{k-1} + 6d_{k-1} h_{k-1}, \\
 (c) \quad c_0 &= c_n = 0.
 \end{aligned}$$

- Umstellung

$$\begin{aligned}
 (b3) \quad d_k &= \frac{1}{3h_k} (c_{k+1} - c_k), \quad k = 0, 1, \dots, n-1, \\
 &\quad d_{k-1} \text{ in (b2), (b1) einsetzen,} \\
 (b2) \quad b_k &= b_{k-1} + (c_k + c_{k-1}) h_{k-1}, \quad k = 1, 2, \dots, n, \\
 (b1) \quad b_k &= \frac{1}{h_k} (a_{k+1} - a_k) - \frac{1}{3} (2c_k + c_{k+1}) h_k, \quad k = 0, 1, \dots, n-1, \\
 (c) \quad c_0 &= c_n = 0.
 \end{aligned}$$

- (b1) in (b2) einsetzen

$$\begin{aligned}
 \frac{1}{h_k} (a_{k+1} - a_k) - \frac{1}{3} (2c_k + c_{k+1}) h_k &= \frac{1}{h_{k-1}} (a_k - a_{k-1}) - \frac{1}{3} (2c_{k-1} + c_k) h_{k-1} + (c_k + c_{k-1}) h_{k-1}, \\
 h_{k-1} (c_k + c_{k-1} - \frac{2}{3} c_{k-1} - \frac{1}{3} c_k) + \frac{h_k}{3} (2c_k + c_{k+1}) &= \frac{1}{h_k} (a_{k+1} - a_k) - \frac{1}{h_{k-1}} (a_k - a_{k-1}), \\
 h_{k-1} c_{k-1} + 2(h_{k-1} + h_k) c_k + h_k c_{k+1} &= \frac{3}{h_k} (f_{k+1} - f_k) - \frac{3}{h_{k-1}} (f_k - f_{k-1}) = e_k, \\
 k &= 1, 2, \dots, n-1, \quad c_0 = c_n = 0.
 \end{aligned}$$

Die Lösung des Systems mit Tridiagonalmatrix ergibt  $c_k$  und dann folgen  $d_k, b_k$ .

**Beispiel 1.19** Man berechne den natürlichen kubischen Spline mit

$n = 2$ ,  $h_k = h = 0.5$ ,

$x_k$	0	0.5	1
$f_k$	1	-0.5	2

Zunächst bestimmt man die Koeffizienten  $c_k$  aus dem Tridiagonalsystem, das sich hier auf eine Gleichung reduziert.

$$c_0 = c_2 = 0, \quad 2(h_0 + h_1)c_1 = e_1 = \frac{3}{h}(f_2 - 2f_1 + f_0), \quad a_k = f_k \Rightarrow c_1 = 12.$$

Dann erfolgt die Berechnung von  $b_k$ .

$$b_0 = 2(a_1 - a_0) - \frac{1}{3}(2c_0 + c_1)\frac{1}{2} = -5,$$

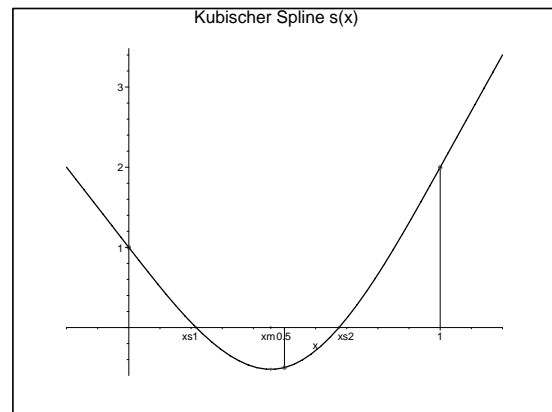
$$b_1 = b_0 + (c_1 + c_0)\frac{1}{2} = 1,$$

$$b_2 = b_1 + (c_2 + c_1)\frac{1}{2} = 7.$$

Die Bestimmung von  $d_k$  ergibt

$$d_0 = \frac{1}{3 \cdot \frac{1}{2}}(c_1 - c_0) = 8,$$

$$d_1 = \frac{1}{3 \cdot \frac{1}{2}}(c_2 - c_1) = -8.$$



**Abb. 1.19** Kubische Splinefunktion

Als Eigenschaften des resultierenden Splines

$$s(x) = \begin{cases} 1 - 5x, & x \in (-\infty, 0], \\ s^{(0)}(x) = 1 - 5x + 8x^3, & x \in [0, 0.5], \\ s^{(1)}(x) = -\frac{1}{2} + (x - \frac{1}{2}) + 12(x - \frac{1}{2})^2 - 8(x - \frac{1}{2})^3, & x \in [0.5, 1], \\ s^{(2)}(x) = 2 + 7(x - 1), & x \in [1, \infty), \end{cases}$$

notieren wir seine beiden Nullstellen  $x_1^* \approx 0.216$ ,  $x_2^* \approx 0.675$  sowie die Minimumstelle  $x_{min} \approx 0.456$  mit  $s(x_{min}) \approx -0.521$ .

### Satz 1.23 Existenz und Eindeutigkeit der kubischen Splinefunktion

*Zu paarweise verschiedenen Stützstellen  $x_k$  existiert stets genau eine natürliche kubische Splinefunktion.*

### Satz 1.24 HOLLADAY zur Optimalität kubischer Splines

*In der Klasse  $C^2[x_0, x_n]$  der Funktionen  $f(x)$ , die auf der Referenz  $R_0$  die Interpolationsbedingung  $f(x_k) = f_k$ ,  $k = 0, 1, \dots, n$ , erfüllen, minimiert die natürliche kubische Splinefunktion  $s(x)$  das Integral für die Gesamtkrümmung*

$$\int_{x_0}^{x_n} [f''(x)]^2 dx. \quad (1.79)$$

### 1.7.3 B-Splines

Wir betrachten nun einen etwas anderen Zugang zu den Splinekurven.

Er beruht auf der Konstruktion von entsprechend glatten Basispolynomen und ihrer Linearkombination.

**Definition 1.11** Gegeben seien das Intervall  $I = [a, b] \in \mathbb{R}$  und die Unterteilung (Folge der Stützstellen)  $\Delta = \{x_0, x_1, \dots, x_{l+1}\}$  mit  $a = x_0 < x_1 < \dots < x_{l+1} = b$ . Ein Spline vom Grad  $k - 1$  bzw. Spline der Ordnung  $k \geq 1$  über  $\Delta$  ist eine  $(k - 2)$ -mal stetig differenzierbare Funktion, welche in jedem Intervall  $[x_i, x_{i+1}]$  mit einem Polynom vom Grad  $\leq k - 1$  übereinstimmt. Der Fall  $k = 1$  nimmt jedoch eine Sonderstellung ein.

Der Raum aller Splines über  $\Delta$  vom Grad  $k - 1$  wird mit  $\mathcal{S}_{k,\Delta}$  bezeichnet.

**Bemerkung 1.6**  $\mathcal{S}_{k,\Delta}$  ist ein linearer Vektorraum der Dimension  $l + k$ . Insbesondere sind also  $l + k$  unabhängige Bedingungen nötig, um einen Spline vom Grad  $k - 1$  eindeutig zu bestimmen.

$\mathcal{S}_{2,\Delta}$  enthält den Polygonzug mit  $l + 2$  Bedingungen,

$\mathcal{S}_{3,\Delta}$  die quadratischen Splines mit  $l + 3$  Bedingungen und

$\mathcal{S}_{4,\Delta}$  die kubischen Splines mit  $l + 4$  Bedingungen.

Die Begründung der Dimension  $l + k$  des Vektorraums  $\mathcal{S}_{k,\Delta}$  und damit der Anzahl der unabhängigen Bedingungen erfolgt später.

Eine effiziente Möglichkeit, eine Menge von Datenpaaren durch Splinekurven zu interpolieren, bietet die Verwendung von **Basis-Splines**, auch B-Splines genannt.

Dazu nehmen wir eine vorläufige Umbenennung der Stützstellen vor.

#### Definition 1.12 B-Splines

Sei  $\xi_1 \leq \xi_2 \leq \dots \leq \xi_n \leq \xi_{n+1}$  eine Folge  $\Delta_1$  von Stützstellen.

B-Splines  $N_{ik}(x)$  der Ordnung  $k$  ( $k = 1, 2, \dots, n, n + 1$ ) sind für  $i = 1, 2, \dots, n + 1 - k$  wie folgt rekursiv definiert.

$k = 1$ :

$$N_{i,1}(x) = \chi_{[\xi_i, \xi_{i+1})}(x) = \begin{cases} 1, & \text{falls } \xi_i \leq x < \xi_{i+1}, \\ 0, & \text{sonst.} \end{cases} \quad (1.80)$$

Dieser B-Spline hat die Eigenschaft einer charakteristischen Funktion auf einer Menge und wird für den Anfang der rekursiven Berechnung gebraucht.

$k > 1$ :

$$N_{ik}(x) = \frac{x - \xi_i}{\xi_{i+k-1} - \xi_i} N_{i,k-1}(x) + \frac{\xi_{i+k} - x}{\xi_{i+k} - \xi_{i+1}} N_{i+1,k-1}(x). \quad (1.81)$$

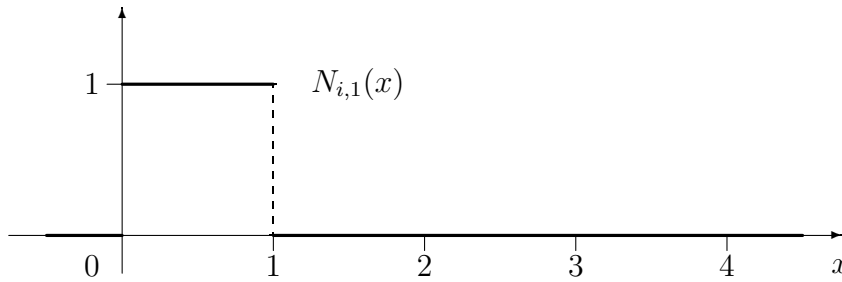
Die Vereinbarung für zusammenfallende Stützstellen  $\xi_i = \xi_j$  ist, dass dann Quotienten der Form "0/0" als Null interpretiert werden.

**Beispiel 1.20** Betrachten wir B-Splines bei paarweise verschiedenen Stellen  $\xi_i$ .

Zusätzlich geben wir auf einem inneren Standardintervall  $[\xi_i, \xi_{i+k}] = [0, k]$  der Unterteilung  $\Delta_1 = \{\xi_1, \xi_2, \dots, \xi_{n+1}\}$  den B-Spline mit diesem Träger sowie seine grafische Darstellung an.

$k = 1$ :  $N_{i,1}(x) = \chi_{[\xi_i, \xi_{i+1})}(x)$ ,  $i = 1, 2, \dots, n$ , ist stückweise konstant und  $\text{supp} N_{i,1} = [\xi_i, \xi_{i+1}]$  ist der Träger.

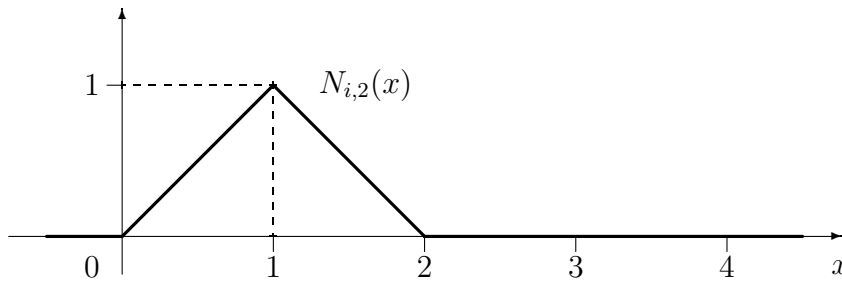
$$N_{i,1}(x) = \begin{cases} 1, & x \in [0, 1) \\ 0, & \text{sonst} \end{cases}, \quad \sum_{i=1}^n N_{i,1}(x) = 1 \text{ für } x \in [\xi_1, \xi_{n+1}).$$



**Abb. 1.20** B-Splines  $N_{i,1}(x)$  auf der Unterteilung  $\Delta_1$

$k = 2$ :  $N_{i,2}(x) = \frac{x - \xi_i}{\xi_{i+1} - \xi_i} N_{i,1}(x) + \frac{\xi_{i+2} - x}{\xi_{i+2} - \xi_{i+1}} N_{i+1,1}(x)$ ,  $i = 1, 2, \dots, n-1$ , ist stetig und stückweise linear,  $\text{supp} N_{i,2} = [\xi_i, \xi_{i+2}]$ .

$$N_{i,2}(x) = \begin{cases} x, & x \in [0, 1] \\ 1 - x, & x \in [1, 2] \\ 0, & \text{sonst} \end{cases}, \quad \sum_{i=1}^{n-1} N_{i,2}(x) = 1 \text{ für } x \in [\xi_1, \xi_{n+1}].$$



**Abb. 1.21** B-Splines  $N_{i,2}(x)$  im Inneren der Unterteilung  $\Delta_1$

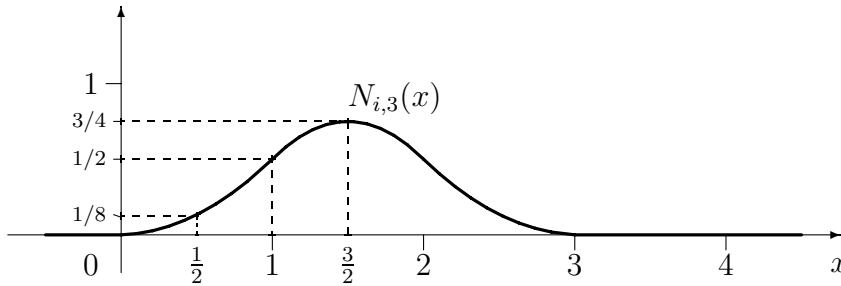
Die intervallweise Notation des B-Splines ist

$$N_{i,2}(x) = \begin{cases} \frac{x - \xi_i}{\xi_{i+1} - \xi_i}, & x \in [\xi_i, \xi_{i+1}], \\ \frac{\xi_{i+2} - x}{\xi_{i+2} - \xi_{i+1}}, & x \in [\xi_{i+1}, \xi_{i+2}], \\ 0, & \text{sonst.} \end{cases}$$

$$k = 3: N_{i,3}(x) = \frac{x - \xi_i}{\xi_{i+2} - \xi_i} N_{i,2}(x) + \frac{\xi_{i+3} - x}{\xi_{i+3} - \xi_{i+1}} N_{i+1,2}(x), \quad i = 1, 2, \dots, n-2,$$

ist differenzierbar und stückweise quadratisch,  $\text{supp } N_{i,3} = [\xi_i, \xi_{i+3}]$ .

$$N_{i,3}(x) = \begin{cases} \frac{1}{2}x^2, & x \in [0, 1] \\ -x^2 + 3x - \frac{3}{2}, & x \in [1, 2] \\ \frac{1}{2}(x-3)^2, & x \in [2, 3] \\ 0, & \text{sonst} \end{cases}, \quad \sum_{i=1}^{n-2} N_{i,3}(x) = 1.$$



**Abb. 1.22** B-Splines  $N_{i,3}(x)$  im Inneren der Unterteilung  $\Delta_1$

Allgemein erhält man

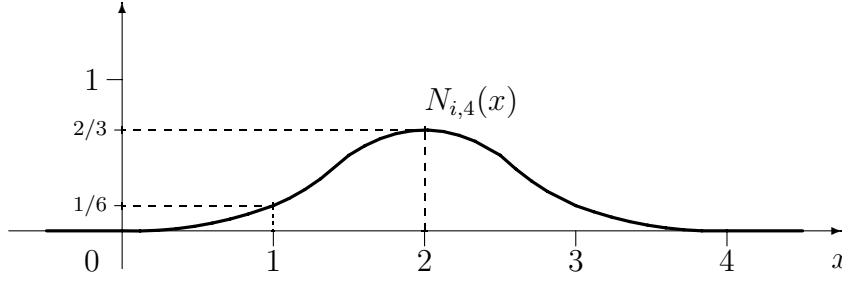
$$N_{i,3}(x) = \begin{cases} \frac{1}{\xi_{i+2} - \xi_i} \frac{(x - \xi_i)^2}{\xi_{i+1} - \xi_i}, & x \in [\xi_i, \xi_{i+1}], \\ \frac{1}{\xi_{i+2} - \xi_{i+1}} \left[ \frac{(x - \xi_i)(\xi_{i+1} - x)}{\xi_{i+2} - \xi_i} + \frac{(x - \xi_{i+1})(\xi_{i+3} - x)}{\xi_{i+3} - \xi_{i+1}} \right], & x \in [\xi_{i+1}, \xi_{i+2}], \\ \frac{1}{\xi_{i+3} - \xi_{i+1}} \frac{(\xi_{i+3} - x)^2}{\xi_{i+3} - \xi_{i+2}}, & x \in [\xi_{i+2}, \xi_{i+3}], \\ 0, & \text{sonst.} \end{cases}$$

$$k = 4: N_{i,4}(x) = \frac{x - \xi_i}{\xi_{i+3} - \xi_i} N_{i,3}(x) + \frac{\xi_{i+4} - x}{\xi_{i+4} - \xi_{i+1}} N_{i+1,3}(x), \quad i = 1, 2, \dots, n-3,$$

ist zweimal differenzierbar und stückweise kubisch,  $\text{supp } N_{i,4} = [\xi_i, \xi_{i+4}]$ .

$$N_{i,4}(x) = \begin{cases} \frac{1}{6}x^3, & x \in [0, 1] \\ \frac{1}{6} + \frac{1}{2}(x-1) + \frac{1}{2}(x-1)^2 - \frac{1}{2}(x-1)^3, & x \in [1, 2] \\ \frac{4}{6} - (x-2)^2 + \frac{1}{2}(x-2)^3, & x \in [2, 3] \\ \frac{1}{6} - \frac{1}{2}(x-3) + \frac{1}{2}(x-3)^2 - \frac{1}{6}(x-3)^3 \\ \quad = \frac{1}{6}(4-x)^3, & x \in [3, 4] \\ 0, & \text{sonst} \end{cases}, \quad \sum_{i=1}^{n-3} N_{i,4}(x) = 1.$$





**Abb. 1.23** B-Splines  $N_{i,4}(x)$  im Inneren der Unterteilung  $\Delta_1$

Allgemein folgt bei entsprechender Definition und Einsetzen von Zwischengrößen  $g_j$

$$N_{i,4}(x) = \begin{cases} \frac{(x - \xi_i)^3}{(\xi_{i+1} - \xi_i)(\xi_{i+2} - \xi_i)(\xi_{i+3} - \xi_i)} = g_1, & x \in [\xi_i, \xi_{i+1}], \\ g_1 - (x - \xi_{i+1})^3 \left[ \frac{1}{(\xi_{i+2} - \xi_{i+1})(\xi_{i+3} - \xi_{i+1})(\xi_{i+4} - \xi_{i+1})} \right. \\ \quad + \frac{1}{(\xi_{i+2} - \xi_{i+1})(\xi_{i+3} - \xi_{i+1})(\xi_{i+3} - \xi_i)} \\ \quad + \frac{1}{(\xi_{i+2} - \xi_{i+1})(\xi_{i+2} - \xi_i)(\xi_{i+3} - \xi_i)} \\ \quad \left. + \frac{1}{(\xi_{i+1} - \xi_i)(\xi_{i+2} - \xi_i)(\xi_{i+3} - \xi_i)} \right] = g_2, & x \in [\xi_{i+1}, \xi_{i+2}], \\ g_2 + (x - \xi_{i+2})^3 \left[ \frac{1}{(\xi_{i+3} - \xi_{i+2})(\xi_{i+4} - \xi_{i+2})(\xi_{i+4} - \xi_{i+1})} \right. \\ \quad + \frac{1}{(\xi_{i+3} - \xi_{i+2})(\xi_{i+3} - \xi_{i+1})(\xi_{i+4} - \xi_{i+1})} \\ \quad + \frac{1}{(\xi_{i+2} - \xi_{i+1})(\xi_{i+3} - \xi_{i+1})(\xi_{i+4} - \xi_i)} \\ \quad + \frac{1}{(\xi_{i+3} - \xi_{i+2})(\xi_{i+3} - \xi_{i+1})(\xi_{i+3} - \xi_i)} \\ \quad + \frac{1}{(\xi_{i+2} - \xi_{i+1})(\xi_{i+3} - \xi_{i+1})(\xi_{i+3} - \xi_i)} \\ \quad \left. + \frac{1}{(\xi_{i+2} - \xi_{i+1})(\xi_{i+2} - \xi_i)(\xi_{i+3} - \xi_i)} \right], & x \in [\xi_{i+2}, \xi_{i+3}], \\ \frac{(\xi_{i+4} - x)^3}{(\xi_{i+4} - \xi_{i+1})(\xi_{i+4} - \xi_{i+2})(\xi_{i+4} - \xi_{i+3})}, & x \in [\xi_{i+3}, \xi_{i+4}], \\ 0, & \text{sonst.} \end{cases}$$

Splinekurven aus  $\mathcal{S}_{k,\Delta_1}$  kann man damit als Linearkombinationen der zugehörigen B-Splines  $N_{ik}(x)$ ,  $i = 1, 2, \dots, n - (k - 1)$ , schreiben.

Im Folgenden betrachten wir die **erweiterte Stützstellenfolge**  $\Delta_2$ , in welcher der Anfangsknoten und der Endknoten  $k$ -mal vorkommen.

$$a = \xi_1 = \dots = \xi_k < \xi_{k+1} < \dots < \xi_n < \xi_{n+1} = \dots = \xi_{n+k} = b. \quad (1.82)$$

Man beachte hier die neue Nummerierung der Stützstellen.

Für die Eigenschaft der B-Splines, eine Funktionenbasis zu sein, werden wir zunächst Potenzfunktionen als Linearkombination dieser darstellen.

### Satz 1.25 Marsden-Identität

Für  $x \in [a, b]$ ,  $s \in \mathbb{R}$  und  $\Delta_2$  gilt

$$(x - s)^{k-1} = \sum_{i=1}^n \phi_{ik}(s) N_{ik}(x) \quad \text{mit} \quad \phi_{ik}(s) = \prod_{j=1}^{k-1} (\xi_{i+j} - s). \quad (1.83)$$

**Beweis.** Für  $k = 1$  ist die Behauptung richtig wegen  $\phi_{i,1}(s) = 1$ ,  $\sum_{i=1}^n N_{i,1}(x) = 1$ .

Es sei  $k > 1$ . Die Behauptung sei bewiesen für alle  $l \leq k - 1$ .

Durch Einsetzen der Rekursionsformel (1.81) folgt

$$\begin{aligned} & \sum_{i=1}^n \phi_{ik}(s) N_{ik}(x) = \\ &= \sum_{i=2}^n \left( \frac{x - \xi_i}{\xi_{i+k-1} - \xi_i} \phi_{ik}(s) + \frac{\xi_{i+k-1} - x}{\xi_{i+k-1} - \xi_i} \phi_{i-1,k}(s) \right) N_{i,k-1}(x) \\ &= \sum_{i=2}^n \prod_{j=1}^{k-2} (\xi_{i+j} - s) \underbrace{\left( \frac{x - \xi_i}{\xi_{i+k-1} - \xi_i} (\xi_{i+k-1} - s) + \frac{\xi_{i+k-1} - x}{\xi_{i+k-1} - \xi_i} (\xi_i - s) \right)}_{x-s} N_{i,k-1}(x) \\ &= (x - s) \sum_{i=2}^n \phi_{i,k-1}(s) N_{i,k-1}(x) \\ &= (x - s)(x - s)^{k-2} \\ &= (x - s)^{k-1}. \end{aligned}$$

□

**Folgerung 1.26** (1) Jedes Polynom vom Grad  $\leq k - 1$  lässt sich als Linearkombination der B-Splines  $N_{1,k}, N_{2,k}, \dots, N_{n,k}$  schreiben.

**Beweis.** Die  $l$ -te Ableitung von  $f(s) = (x - s)^{k-1}$  aus (1.83) an der Stelle Null ist

$$f^{(l)}(0) = (k-1)(k-2) \cdot \dots \cdot (k-l) \cdot (-1)^l x^{k-l-1} = \sum_{i=1}^n \phi_{ik}^{(l)}(0) N_{ik}(x).$$

Mit  $m = k - l - 1 \leq k - 1$  folgt bei Umstellung nach  $x^m$  daraus

$$x^m = \frac{(-1)^{k-m-1}}{(k-1)(k-2) \cdot \dots \cdot (m+1)} \sum_{i=1}^n \phi_{ik}^{(k-m-1)}(0) N_{ik}(x). \quad (1.84)$$

Somit ist eine beliebige Potenz  $x^m$ ,  $m = 0, 1, \dots, k-1$ , eine Linearkombination der B-Splines  $N_{ik}(x)$  und damit auch jedes Polynom vom Grad  $\leq k-1$ .  $\square$

(2) Wie im Beispiel 1.19 schon angedeutet wurde, bilden die B-Splines eine Zerlegung der Eins, also gilt

$$\sum_{i=1}^n N_{ik}(x) = 1 \quad \text{für } x \in [a, b]. \quad (1.85)$$

**Beweis.** Mit  $m = 0$ , d. h.  $l = k-1$ , folgt aus dem Beweis von Teil (1) der Folgerung

$$1 = x^0 = \frac{(-1)^{k-1}}{(k-1)!} \sum_{i=1}^n \phi_{ik}^{(k-1)}(0) N_{ik}(x).$$

Außerdem ist die  $(k-1)$ -ste Ableitung

$$\phi_{ik}^{(k-1)}(s) = \left[ \prod_{j=1}^{k-1} (\xi_{i+j} - s) \right]^{(k-1)} = [(-1)^{k-1} s^{k-1} + \dots]^{(k-1)} = (-1)^{k-1} (k-1)!$$

und somit auch  $\phi_{ik}^{(k-1)}(0) = (-1)^{k-1} (k-1)!$ . Dies eingesetzt in die Summenformel ergibt die Bedingung für die Zerlegung der Eins.  $\square$

(3) Die B-Splines sind lokal linear unabhängig.

Ist  $(c, d) \subset [a, b]$  ein Intervall mit  $c < d$ , und gilt für alle  $x \in (c, d)$

$$\sum_{i=1}^n c_i N_{ik}(x) = 0,$$

so ist  $c_i = 0$ , falls  $(c, d) \cap (\xi_i, \xi_{i+k}) \neq \emptyset$ .

Nur die  $N_{ik}(x)$ , die in  $(c, d)$  einen Beitrag liefern, haben  $c_i = 0$ .

**Beweis.** Man wählt ein Teilintervall  $(c_0, d_0) \subseteq (c, d)$  so, dass keine Stützstelle  $\xi_j$  in  $(c_0, d_0)$  enthalten ist. Der Raum  $\mathcal{P}_{k-1}(c_0, d_0)$  der Polynome vom Grad  $\leq k-1$  auf  $(c_0, d_0)$  hat die Dimension  $k$  und wird gemäß Punkt (1) der Folgerung von den B-Splines  $N_{ik}(x)$  aufgespannt. Auf  $(c_0, d_0)$  sind aber nur  $k$  B-Splines von Null verschieden. Diese müssen daher linear unabhängig sein, also ist  $c_i = 0$  für die entsprechenden Koeffizienten.  $\square$

(4) Jeder Spline  $s(x) \in \mathcal{S}_{k, \Delta_2}$  besitzt eine eindeutige Darstellung als Linearkombination

$$s(x) = \sum_{i=1}^n d_i N_{ik}(x). \quad (1.86)$$

Hier ist der Index  $n$  aus der Stützstellenfolge  $\Delta_2$  die Dimension des Vektorraums und gleichzeitig die Anzahl der unabhängigen Bedingungen.

Die Größe  $n$  soll nun in Bezug auf die anfängliche Unterteilung  $\Delta = \{x_0, x_1, \dots, x_{l+1}\}$  mit  $a = x_0 < x_1 < \dots < x_{l+1} = b$  sowie deren Modifikationen  $\Delta_1$  und Erweiterungen  $\Delta_2, \Delta_{erw}$  bestimmt werden. Dabei werden wir in der Unterteilung  $\Delta_{erw}$  die  $k$ -malige Wiederholung (Notation) der 1. Stützstelle links sowie der letzten Stützstelle rechts ausgehend von  $\Delta$  mit fortlaufenden Indizes versehen.

Das folgende Schema demonstriert die Zuordnung der Stützstellen.

$$\begin{array}{rcl}
 \Delta_1 : & a = \bar{\xi}_1 < \bar{\xi}_2 < \dots < \bar{\xi}_{\bar{n}} < \bar{\xi}_{\bar{n}+1} = b \\
 & \downarrow \quad \downarrow \dots \downarrow \quad \downarrow \\
 \Delta_2 : & \xi_1 = \dots = \xi_{k-1} = \xi_k < \xi_{k+1} < \dots < \xi_n < \xi_{n+1} = \xi_{n+2} = \dots = \xi_{n+k} \\
 & \downarrow \quad \downarrow \dots \downarrow \quad \downarrow \\
 \Delta_{erw} : & \underbrace{x_{-(k-1)} = \dots = x_{-1} = x_0}_{k\text{-mal}} < x_1 < \dots < x_l < \underbrace{x_{l+1} = x_{l+2} = \dots = x_{l+k}}_{k\text{-mal}}
 \end{array}$$

Die **Dimension** des Vektorraums ergibt sich auf zweierlei Weise als

- Anzahl der Komponenten  $\xi_1, \xi_2, \dots, \xi_n$  aus  $\Delta_2$ :  $n = \bar{n} + k - 1$ , oder
- Anzahl der Komponenten  $x_{-(k-1)}, x_{-(k-2)}, \dots, x_l$  aus  $\Delta_{erw}$ :  $l + k$ .

Somit erhalten wir die eingangs genannte Dimensionsgröße  $\dim = l + k$ .

In Beispielrechnungen hat man somit zwei Möglichkeiten der Bezeichnung der erweiterten Stützstellenfolge. Entweder nimmt man eine Umindizierung wie bei  $\Delta_2$  vor, oder man lässt ausgehend von  $\Delta$  bzw.  $\Delta_1$  die Nummerierung nach “links“ und “rechts“ fortlaufen. Der 1. Index im B-Spline  $k$ -ter Ordnung  $N_{ik}(x)$  bewegt sich dann entsprechend der Indizes der erweiterten Stützstellenfolge.

Var.	Unterteilung	Indizierung $i$ in $N_{ik}(x)$	Dimension
(V1)	$\Delta_2$	$1, 2, \dots, n$	$n$
(V2)	$\Delta = \{\xi_1, \xi_2, \dots, \xi_n\}$	$-(k-1) + 1, -(k-1) + 2, \dots, n-1$	$n + k - 2$
(V3)	$\Delta = \{\xi_1, \xi_2, \dots, \xi_{n+1}\}$	$-(k-1) + 1, -(k-1) + 2, \dots, n$	$n + k - 1$
(V4)	$\Delta = \{\xi_0, \xi_1, \dots, \xi_n\}$	$-(k-1), -(k-1) + 1, \dots, n-1$	$n + k - 1$
(V5)	$\Delta = \{x_0, x_1, \dots, x_{l+1}\}$	$-(k-1), -(k-1) + 1, \dots, l$	$l + k$

**Tab. 1.4** Varianten der Unterteilung und Stützstellenindizierung bei B-Splines

Bevorzugte Anwendung findet die beidseitige Indexerweiterung in den Varianten (V2) und (V5).

**Beispiel 1.21** Wir bestimmen Splines zur Referenz mit 4 Knoten ( $n = 4, l = 2$ )

$\xi_j = x_{j-1}, j = 1, 2, 3, 4$	0	1	2	3
$y_j$	2	3	3	5

unter Anwendung von Variante (V2).

1. Linearer Spline, Polygonzug der Ordnung  $k = 2$ 

Die erweiterte Stützstellenfolge ist  $\{\xi_0, \xi_1, \dots, \xi_4, \xi_5\}$ ,  $\xi_0 = \xi_1$ ,  $\xi_4 = \xi_5$ ,  $\dim = 4$ .

$$s(x) = \sum_{i=0}^3 d_i N_{i,2}(x), \text{ wobei}$$

$$N_{0,2}(x) = \begin{cases} 1-x, & x \in [0, 1] \\ 0, & \text{sonst} \end{cases}$$

$$N_{1,2}(x) = \begin{cases} x, & x \in [0, 1] \\ 2-x, & x \in [1, 2] \\ 0, & \text{sonst} \end{cases}$$

$$N_{2,2}(x) = \begin{cases} 0, & \text{sonst} \\ x-1, & x \in [1, 2] \\ 3-x, & x \in [2, 3] \end{cases}$$

$$N_{3,2}(x) = \begin{cases} 0, & \text{sonst} \\ x-2, & x \in [2, 3]. \end{cases}$$

Aus der Forderung  $y_j = s(\xi_j) = \sum_{i=0}^3 d_i N_{i,2}(\xi_j)$ ,  $j = 1, 2, 3, 4$ , folgt  $d_i = y_{i+1}$  und als

linearer Spline  $s(x) = 2N_{0,2}(x) + 3N_{1,2}(x) + 3N_{2,2}(x) + 5N_{3,2}(x)$ .

2. Quadratischer Spline der Ordnung  $k = 3$ 

Die erweiterte Stützstellenfolge ist  $\{\xi_{-1}, \xi_0, \xi_1, \dots, \xi_4, \xi_5, \xi_6\}$  mit  $\xi_{-1} = \xi_0 = \xi_1$ ,  $\xi_4 = \xi_5 = \xi_6$ ,  $\dim = 5$ . Der Splineansatz ist

$$s(x) = \sum_{i=-1}^3 d_i N_{i,3}(x), \text{ wobei}$$

$$N_{-1,3}(x) = \begin{cases} (1-x)^2, & x \in [0, 1] \\ 0, & \text{sonst} \end{cases}$$

$$N_{0,3}(x) = \begin{cases} \frac{x}{2}(4-3x), & x \in [0, 1] \\ \frac{1}{2}(2-x)^2, & x \in [1, 2] \\ 0, & \text{sonst} \end{cases}$$

$$N_{1,3}(x) = \begin{cases} \frac{x^2}{2}, & x \in [0, 1] \\ \frac{x^2}{2} - \frac{3}{2}(x-1)^2, & x \in [1, 2] \\ \frac{1}{2}(3-x)^2, & x \in [2, 3] \end{cases}$$

$$N_{2,3}(x) = \begin{cases} 0, & \text{sonst} \\ \frac{1}{2}(x-1)^2, & x \in [1, 2] \\ \frac{1}{2}(3-x)(3x-5), & x \in [2, 3] \end{cases}$$

$$N_{3,3}(x) = \begin{cases} 0, & \text{sonst} \\ (x-2)^2, & x \in [2, 3]. \end{cases}$$

Die Interpolationsforderung lautet

$$y_j = s(\xi_j) = \sum_{i=-1}^3 d_i N_{i,3}(\xi_j), \quad j = 1, 2, 3, 4.$$

Daraus folgt unter Berücksichtigung der Funktionswerte der B-Splines  $N_{i,3}(x)$  an den Stützstellen das unterbestimmte LGS

$$\begin{aligned} y_1 &= d_{-1}, \\ y_2 &= \frac{1}{2}(d_0 + d_1), \\ y_3 &= \frac{1}{2}(d_1 + d_2), \\ y_4 &= d_3, \end{aligned}$$

bzw. in Matrixform

$$\begin{pmatrix} 1 & & & 0 \\ & 1 & 1 & \\ & & 1 & 1 \\ 0 & & & 1 \end{pmatrix} \begin{pmatrix} d_{-1} \\ d_0 \\ d_1 \\ d_2 \\ d_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 6 \\ 6 \\ 5 \end{pmatrix}.$$

Seine parameterabhängige Lösung ist

$$d_{-1} = 2, \quad d_3 = 5, \quad d_0 = d, \quad d_1 = 6 - d, \quad d_2 = d.$$

Der eine freie Parameter  $d$  muss, wie bekannt, durch eine zusätzliche Bedingung gebunden werden. Dazu verwenden wir die Endbedingung “links“ mit der verschwindenden Ableitung  $s'(\xi_1) = s'(0) = 0$ .

Die benötigten Ableitungen der B-Splines auf dem ersten Teilintervall  $[0,1]$  sind

$$N'_{-1,3}(x) = -2(1-x), \quad N'_{0,3}(x) = 2-3x, \quad N'_{1,3}(x) = x, \quad N'_{2,3}(x) = N'_{3,3}(x) = 0.$$

Somit erhält man

$$0 = s'(0) = 2N'_{-1,3}(0) + dN'_{0,3}(0) + (6-d)N'_{1,3}(0) + 0 = -4 + 2d,$$

damit  $d = 2$  und als endgültige Lösung den quadratischen Spline

$$s(x) = 2N_{-1,3}(x) + 2N_{0,3}(x) + 4N_{1,3}(x) + 2N_{2,3}(x) + 5N_{3,3}(x).$$

Da die B-Splines  $N_{ik}(x)$  vom Grad  $k-1 = 1, 2, 3$  des öfteren verwendet werden, verschaffen wir uns einen Überblick mit der Standardunterteilung

$$\begin{aligned} n &= 5, \quad \Delta = \{\xi_1, \xi_2, \xi_3, \xi_4, \xi_5\} = \{x_0, x_1, x_2, x_3, x_4\} = \{0, 1, 2, 3, 4\}, \\ \Delta_{erw} &= \{\xi_{-(k-1)+1}, \xi_{-(k-1)+2}, \dots, \xi_{5+(k-1)}\}. \end{aligned}$$

Maple bietet dafür die hilfreiche Funktion

$$\text{bspline}(k-1, x, [x(1), x(2), \dots, x(k+1)]) ,$$

wobei  $k-1$  der Grad ist, die Ordnung dann  $k$ ,  $x$  das Argument in der Funktionsdarstellung sowie die  $x(1), \dots, x(k+1)$  die  $k+1$  Stützstellenwerte. Damit ist das Intervall  $[x(1), x(k+1)]$  der aus  $k$  Teilintervallen bestehende Träger des B-Splines  $N_{ik}(x)$  mit  $\xi_i = x(1)$ . Die Berechnung erfolgt mit den Rekursionsbeziehungen (1.80), (1.81).

### Einige Berechnungen und Grafik in Maple

Man beachte die Symmetrie der Splines bez. der Mitte des Intervalls der Stützstellen.

```
> restart:
> with(plots):
> readlib(spline): readlib(bspline):

# Splineinterpolation mit B-Splines
# ausgewaehlte B-Splines
> n := 5:
> xi := [0, 1, 2, 3, 4]:

> bspline(1,x,[0,0,1]);
> plot(bspline(1,x,[0,0,1]),x=-1..5,title='N02(x)');
```

$$\begin{cases} 0 & x < 0 \\ \{ & \\ \{ 1-x & x < 1 \\ \{ & \\ \{ 0 & \text{otherwise} \end{cases}$$

```
> bspline(2,x,[0,0,1,2]);
> plot(bspline(2,x,[0,0,1,2]),x=-1..5,title='N03(x)');
```

$$\begin{cases} 0 & x < 0 \\ \{ & \\ \{ & \\ \{ -\frac{3}{2}x^2 + 2x & x < 1 \\ \{ & \\ \{ & \\ \{ \frac{3}{2} - x + \frac{1}{2}(x-1)^2 & x < 2 \\ \{ & \\ \{ 0 & \text{otherwise} \end{cases}$$

```
> bspline(3,x,[0,0,1,2,3]);
> plot(bspline(3,x,[0,0,1,2,3]),x=-1..5,title='N04(x)');
```

```
{ 0                                     x < 0
{
{      3      2
{ - 11/12 x  + 3/2 x                     x < 1
{
{      2      3
{ 1/3 + 1/4 x - 5/4 (x - 1)  + 7/12 (x-1)   x < 2
{
{      2      3
{ 7/6 - 1/2 x + 1/2 (x - 2)  - 1/6 (x-2)   x < 3
{
{ 0                                     otherwise
```

$k=1$ :  $\Delta_{erw} = \{0, 1, 2, 3, 4\}$ .

$k=2$ :  $\Delta_{erw} = \{0, 0, 1, 2, 3, 4, 4\}$ ,  $dim = 5$ .

$N_{i,2}(x)$	Träger	3. Parameter in bspline()
$N_{0,2}(x)$	$[\xi_0, \xi_2] = [0, 1], \xi_0 = \xi_1$	$[0, 0, 1]$
$N_{1,2}(x)$	$[\xi_1, \xi_3] = [0, 2]$	$[0, 1, 2]$
$N_{2,2}(x)$	$[\xi_2, \xi_4] = [1, 3]$	$[1, 2, 3]$
$N_{3,2}(x)$	$[\xi_3, \xi_5] = [2, 4]$	$[2, 3, 4]$
$N_{4,2}(x)$	$[\xi_4, \xi_6] = [3, 4], \xi_6 = \xi_5$	$[3, 4, 4]$

Die “Hütchenfunktionen“  $N_{i,2}(x)$  sind im Beispiel 6.19 schon einmal gezeigt worden.

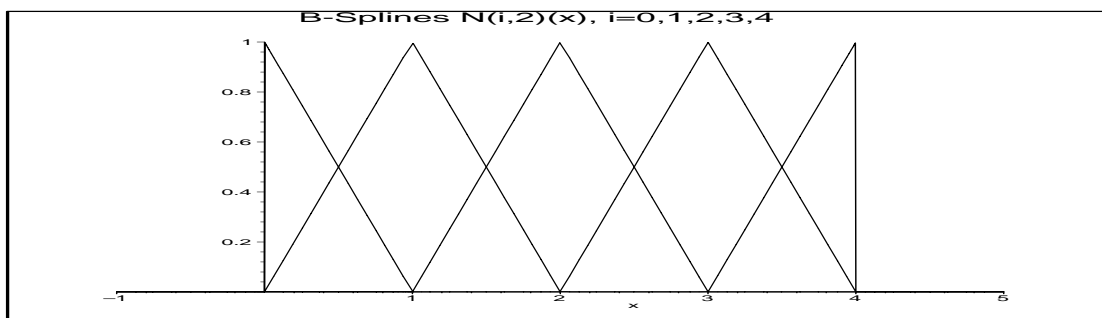


Abb. 1.24 B-Splines 1. Grades  $N_{i,2}(x)$ ,  $i = 0, 1, 2, 3, 4$



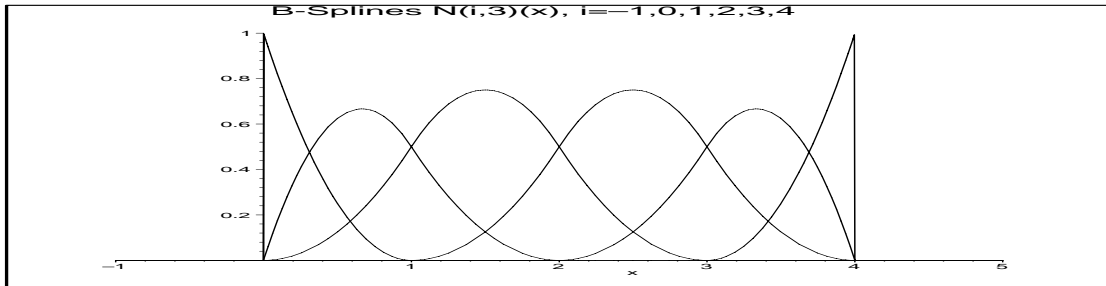
$k = 3$ :  $\Delta_{erw} = \{0, 0, 0, 1, 2, 3, 4, 4, 4\}$ ,  $dim = 6$ .

$N_{i,3}(x)$	Träger	3. Parameter in <code>bspline()</code>
$N_{-1,3}(x)$	$[\xi_{-1}, \xi_2] = [0, 1], \xi_{-1} = \xi_1$	$[0, 0, 0, 1]$
$N_{0,3}(x)$	$[\xi_0, \xi_3] = [0, 2], \xi_0 = \xi_1$	$[0, 0, 1, 2]$
$N_{1,3}(x)$	$[\xi_1, \xi_4] = [0, 3]$	$[0, 1, 2, 3]$
$N_{2,3}(x)$	$[\xi_2, \xi_5] = [1, 4]$	$[1, 2, 3, 4]$
$N_{3,3}(x)$	$[\xi_3, \xi_6] = [2, 4], \xi_6 = \xi_5$	$[2, 3, 4, 4]$
$N_{4,3}(x)$	$[\xi_4, \xi_7] = [3, 4], \xi_7 = \xi_5$	$[3, 4, 4, 4]$

$$N_{-1,3}(x) = \begin{cases} (1-x)^2, & x \in [0, 1] \\ 0, & \text{sonst} \end{cases} \quad N_{4,3}(x) = \begin{cases} (x-3)^2, & x \in [3, 4] \\ 0, & \text{sonst} \end{cases}$$

$$N_{0,3}(x) = \begin{cases} -\frac{3}{2}x^2 + 2x, & x \in [0, 1] \\ \frac{3}{2} - x + \frac{1}{2}(x-1)^2, & x \in [1, 2] \\ 0, & \text{sonst} \end{cases} \quad N_{3,3}(x) = \begin{cases} \frac{1}{2}(x-2)^2, & x \in [2, 3] \\ -\frac{5}{2} + x - \frac{3}{2}(x-3)^2, & x \in [3, 4] \\ 0, & \text{sonst} \end{cases}$$

$$N_{1,3}(x) = \begin{cases} \frac{1}{2}x^2, & x \in [0, 1] \\ \frac{1}{2}x^2 - \frac{3}{2}(x-1)^2, & x \in [1, 2] \\ \frac{1}{2}(3-x)^2, & x \in [2, 3] \\ 0, & \text{sonst} \end{cases} \quad N_{2,3}(x) = \begin{cases} \frac{1}{2}(x-1)^2, & x \in [1, 2] \\ \frac{1}{2}(x-1)^2 - \frac{3}{2}(x-2)^2, & x \in [2, 3] \\ \frac{1}{2}(4-x)^2, & x \in [3, 4] \\ 0, & \text{sonst} \end{cases}$$

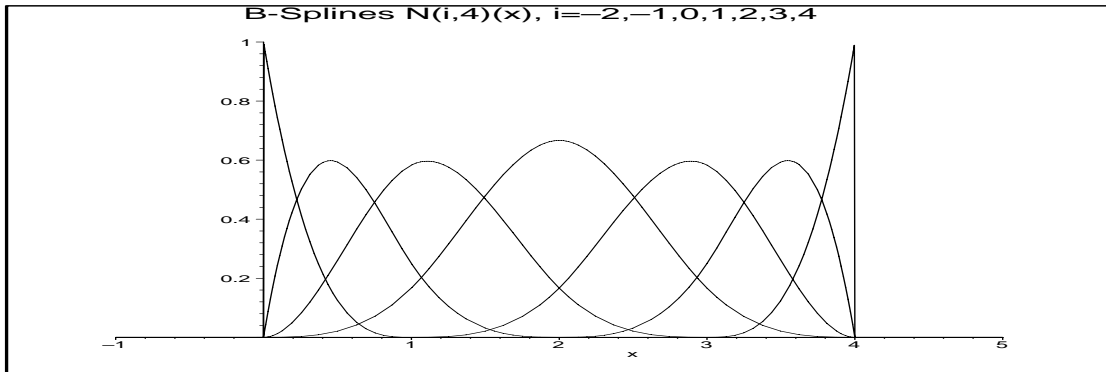


**Abb. 1.25** B-Splines 2. Grades  $N_{i,3}(x)$ ,  $i = -1, 0, 1, 2, 3, 4$

$k = 4$ :  $\Delta_{erw} = \{0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4\}$ ,  $dim = 7$ .

$N_{i,4}(x)$	Träger	3. Parameter in <code>bspline()</code>
$N_{-2,4}(x)$	$[\xi_{-2}, \xi_2] = [0, 1], \xi_{-2} = \xi_1$	$[0, 0, 0, 0, 1]$
$N_{-1,4}(x)$	$[\xi_{-1}, \xi_3] = [0, 2], \xi_{-1} = \xi_1$	$[0, 0, 0, 1, 2]$
$N_{0,4}(x)$	$[\xi_0, \xi_4] = [0, 3], \xi_0 = \xi_1$	$[0, 0, 1, 2, 3]$
$N_{1,4}(x)$	$[\xi_1, \xi_5] = [0, 4]$	$[0, 1, 2, 3, 4]$
$N_{2,4}(x)$	$[\xi_2, \xi_6] = [1, 4], \xi_7 = \xi_5$	$[1, 2, 3, 4, 4]$
$N_{3,4}(x)$	$[\xi_3, \xi_7] = [2, 4], \xi_7 = \xi_5$	$[2, 3, 4, 4, 4]$
$N_{4,4}(x)$	$[\xi_4, \xi_8] = [3, 4], \xi_8 = \xi_5$	$[3, 4, 4, 4, 4]$

$$\begin{aligned}
N_{-2,4}(x) &= \begin{cases} (1-x)^3, & x \in [0, 1] \\ 0, & \text{sonst} \end{cases} \\
N_{-1,4}(x) &= \begin{cases} \frac{7}{4}x^3 - \frac{9}{2}x^2 + 3x, & x \in [0, 1] \\ \frac{1}{4} - \frac{3}{4}(x-1) + \frac{3}{4}(x-1)^2 - \frac{1}{4}(x-1)^3, & x \in [1, 2] \\ 0, & \text{sonst} \end{cases} \\
N_{0,4}(x) &= \begin{cases} -\frac{11}{12}x^3 + \frac{3}{2}x^2, & x \in [0, 1] \\ \frac{7}{12} + \frac{1}{4}(x-1) - \frac{5}{4}(x-1)^2 + \frac{7}{12}(x-1)^3, & x \in [1, 2] \\ \frac{1}{6} - \frac{1}{2}(x-2) + \frac{1}{2}(x-2)^2 - \frac{1}{6}(x-2)^3, & x \in [2, 3] \\ 0, & \text{sonst} \end{cases} \\
N_{1,4}(x) &= \begin{cases} \frac{1}{6}x^3, & x \in [0, 1] \\ \frac{1}{6} + \frac{1}{2}(x-1) + \frac{1}{2}(x-1)^2 - \frac{1}{2}(x-1)^3, & x \in [1, 2] \\ \frac{4}{6} - (x-2)^2 + \frac{1}{2}(x-2)^3, & x \in [2, 3] \\ \frac{1}{6} - \frac{1}{2}(x-3) + \frac{1}{2}(x-3)^2 - \frac{1}{6}(x-3)^3 = \frac{1}{6}(4-x)^3, & x \in [3, 4] \end{cases} \\
N_{3,4}(x) &= \begin{cases} \frac{1}{4}(x-2)^3, & x \in [2, 3] \\ \frac{1}{4} + \frac{3}{4}(x-3) + \frac{3}{4}(x-3)^2 - \frac{7}{4}(x-3)^3, & x \in [3, 4] \\ 0, & \text{sonst} \end{cases} \\
N_{4,4}(x) &= \begin{cases} (x-3)^3, & x \in [3, 4] \\ 0, & \text{sonst.} \end{cases}
\end{aligned}$$



**Abb. 1.26** B-Splines 3. Grades  $N_{i,4}(x)$ ,  $i = -2, -1, 0, 1, 2, 3, 4$

Die  $k$ -fache Wiederholung der 1. und letzten Stützstelle kann man auch umgehen. Der innere Standard-B-Spline  $N_{ik}(x)$  mit seinem Träger  $[\xi_i, \xi_{i+k}]$  wird einfach nach links und rechts verschoben, so dass an den Rändern keine Extra-Formeln zu berechnen sind. Was von diesen B-Splines über das Stützstellenintervall hinausgeht, wird abgeschnitten. Man spricht dann auch von abgebrochenen B-Splines.

Für den Fall  $k = 4$  mit  $n + 1$  äquidistanten Stützstellen

$$x_i = x_0 + ih, \quad i = 0, 1, \dots, n, \quad x_0 = a, \quad h = \frac{b-a}{n},$$

vergrößert man die Unterteilung um die Punkte  $x_{-3}, x_{-2}, x_{-1}$  nach links und  $x_{n+1}, x_{n+2}$ ,  $x_{n+3}$  nach rechts und definiert somit modifizierte Basisfunktionen 3. Grades.

**Definition 1.13 Globale kubische Basissplines**

Der globale kubische Basisspline  $\tilde{N}_{i,4}(x) \in \mathcal{C}^2(-\infty, \infty)$  mit seinem kompakten Träger  $[x_{i-2}, x_{i+2}]$  ist intervallweise definiert und hat die Form

$$\tilde{N}_{i,4}(x) = \frac{1}{h^3} \begin{cases} (x - x_{i-2})^3, & x \in [x_{i-2}, x_{i-1}], \\ h^3 + 3h^2(x - x_{i-1}) + 3h(x - x_{i-1})^2 - 3(x - x_{i-1})^3, & x \in [x_{i-1}, x_i], \\ h^3 + 3h^2(x_{i+1} - x) + 3h(x_{i+1} - x)^2 - 3(x_{i+1} - x)^3, & x \in [x_i, x_{i+1}], \\ (x_{i+2} - x)^3, & x \in [x_{i+1}, x_{i+2}], \\ 0, & \text{sonst.} \end{cases} \quad (1.87)$$

Man prüft nach, dass  $\tilde{N}_{i,4}(x)$  folgende Funktions- und Ableitungswerte an den Stützstellen hat.

	$x_{i-2}$	$x_{i-1}$	$x_i$	$x_{i+1}$	$x_{i+2}$	sonst
$\tilde{N}_{i,4}(x_j)$	0	1	4	1	0	0
$[\tilde{N}_{i,4}(x_j)]'$	0	$3/h$	0	$-3/h$	0	0
$[\tilde{N}_{i,4}(x_j)]''$	0	$6/h^2$	$-12/h^2$	$6/h^2$	0	0

**Tab. 1.5**  $\tilde{N}_{i,4}(x)$  und seine ersten Ableitungen bei  $x_j$

Die Funktionen  $\tilde{N}_{i,4}(x)$ ,  $i = -1, 0, \dots, n, n+1$ , stellen eine Basis für den Raum  $\mathcal{S}_{4,\Delta}$  dar. Der Ansatz für die Splinefunktion ist

$$s(x) = \sum_{i=-1}^{n+1} c_i \tilde{N}_{i,4}(x). \quad (1.88)$$

Die  $n+3$  Unbekannten  $c_i$  ergeben sich aus der Interpolationsforderung

$$y_j = s(x_j) = \sum_{i=-1}^{n+1} c_i \tilde{N}_{i,4}(x_j), \quad j = 0, 1, \dots, n,$$

sowie zwei zusätzlichen Bedingungen, wie z. B. bei natürlichen Splines, der Gestalt

$$0 = s''(x_0) = \sum_{i=-1}^{n+1} c_i [\tilde{N}_{i,4}(x_0)]'', \quad 0 = s''(x_n) = \sum_{i=-1}^{n+1} c_i [\tilde{N}_{i,4}(x_n)]''.$$

Somit gelangen wir auf die Lösung eines LGS mit einer Koeffizientenmatrix in "Fast"-Tridiagonalform.

$$\begin{pmatrix} \frac{6}{h^2} & -\frac{12}{h^2} & \frac{6}{h^2} & & & 0 \\ 1 & 4 & 1 & & & \\ & 1 & 4 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 4 & 1 \\ 0 & & & \frac{6}{h^2} & -\frac{12}{h^2} & \frac{6}{h^2} \end{pmatrix} \begin{pmatrix} c_{-1} \\ c_0 \\ c_1 \\ \dots \\ c_n \\ c_{n+1} \end{pmatrix} = \begin{pmatrix} 0 \\ y_0 \\ y_1 \\ \dots \\ y_n \\ 0 \end{pmatrix}. \quad (1.89)$$

Man eliminiert die beiden Nichtnullkomponenten außerhalb des Tridiagonalbandes und löst dann das LGS mit Gauß-Elimination.

Für jedes Teilintervall  $[x_i, x_{i+1}]$  ergibt sich der lokale kubische Spline

$$s^{(i)}(x) = c_{i-1} \tilde{N}_{i-1,4}(x) + c_i \tilde{N}_{i,4}(x) + c_{i+1} \tilde{N}_{i+1,4}(x) + c_{i+2} \tilde{N}_{i+2,4}(x), \quad i = 0, 1, \dots, n-1.$$

Die vier Beiträge der B-Splines auf diesem Intervall bezeichnet man als lokale kubische Basisfunktionen.

Auch die anderen Endbedingungen kann man analog einbeziehen.

Abschließend noch die Formel für eine lokale kubische Splinefunktion auf der Basis von Funktionswerten und 1. Ableitungen.

**Satz 1.27** *Ein interpolierender kubischer Spline lässt sich auf dem Intervall  $[x_i, x_{i+1}]$  wie folgt darstellen.*

$$\begin{aligned} s^{(i)}(x) = & y_i + \lambda_i(x - x_i) + \frac{\frac{y_{i+1} - y_i}{x_{i+1} - x_i} - \lambda_i}{x_{i+1} - x_i} (x - x_i)^2 \\ & + \frac{\lambda_{i+1} - 2\frac{y_{i+1} - y_i}{x_{i+1} - x_i} + \lambda_i}{(x_{i+1} - x_i)^2} (x - x_i)^2 (x - x_{i+1}), \quad i = 0, 1, \dots, n-1. \end{aligned} \quad (1.90)$$

Bei dem angegebenen Polynom handelt es sich um das kubische Hermitesche Interpolationspolynom zu den Daten  $(x_i, y_i, \lambda_i)$ ,  $(x_{i+1}, y_{i+1}, \lambda_{i+1})$ , d. h. es gilt

$$\left. \begin{aligned} s^{(i)}(x_i) &= y_i, & s^{(i)}(x_{i+1}) &= y_{i+1} \\ s^{(i)}(x_i)' &= \lambda_i, & s^{(i)}(x_{i+1})' &= \lambda_{i+1} \end{aligned} \right\} \quad i = 0, 1, \dots, n-1.$$

**Beweis.** Man prüft die Interpolationsbedingungen nach. □

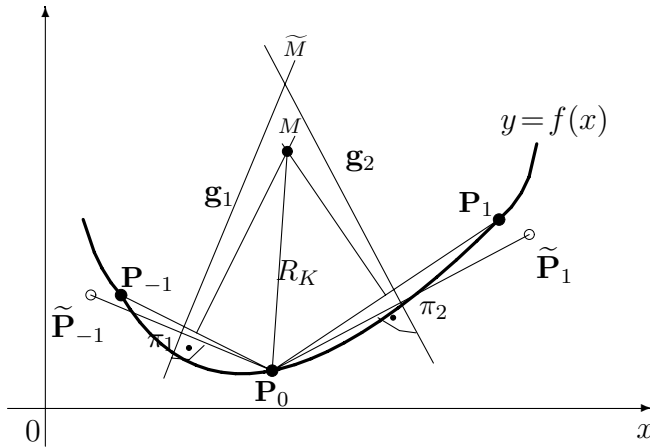
### 1.7.4 Krümmung einer Kurve

In Betrachtungen zu kubischen Splines spielt die zweite Ableitung, also die Krümmung der Kurve eine Rolle. Dies hat auch auf die glättenden Eigenschaften solcher Funktionen Einfluss. Deshalb soll die Definition des Krümmungsradius an  $f(x)$  in  $x$  hergeleitet werden.

Gegeben sei die zweimal stetig differenzierbare Funktion  $f : [a, b] \rightarrow \mathbb{R}$  und ein  $x \in (a, b)$  mit  $f''(x) \neq 0$ .

#### Konstruktion des Schmiegekreises an $f$ in $x$

Sei  $y = f(x)$ ,  $y' = f'(x)$ ,  $y'' = f''(x)$ . Man wähle  $h > 0$  und bestimme den Kreis durch die Punkte  $\{\mathbf{P}_{-1}, \mathbf{P}_0, \mathbf{P}_1\} = \{(x-h, f(x-h)), (x, f(x)), (x+h, f(x+h))\}$  mit dem Mittelpunkt  $M$ .



**Abb. 1.27** Geometrische Erläuterung zum Schmiegekreis, hier ist  $\{\mathbf{P}_{-1}, \mathbf{P}_0, \mathbf{P}_1\} = \{(x-h_1, f(x-h_1)), (x, f(x)), (x+h_2, f(x+h_2))\}$

Dann gelten

$$\mathbf{P}_{-1} = (x-h, f(x-h)) \approx (x-h, y - hy' + h^2/2 y'') = \tilde{\mathbf{P}}_{-1},$$

$$\mathbf{P}_1 = (x+h, f(x+h)) \approx (x+h, y + hy' + h^2/2 y'') = \tilde{\mathbf{P}}_1.$$

Die Geraden  $\mathbf{g}_1$  und  $\mathbf{g}_2$  seien die Mittelsenkrechten auf den Verbindungsstrecken  $\overline{\tilde{\mathbf{P}}_{-1}\mathbf{P}_0}$  und  $\overline{\mathbf{P}_0\tilde{\mathbf{P}}_1}$ , also

$$\mathbf{g}_1 = \pi_1 + \lambda \mathbf{v}_1, \quad \mathbf{g}_2 = \pi_2 + \mu \mathbf{v}_2$$

mit den Streckenmittelpunkten

$$\pi_1 = \frac{1}{2}(\tilde{\mathbf{P}}_{-1} + \mathbf{P}_0) = \frac{1}{2}(2x-h, 2y - hy' + h^2/2 y''),$$

$$\pi_2 = \frac{1}{2}(\mathbf{P}_0 + \tilde{\mathbf{P}}_1) = \frac{1}{2}(2x+h, 2y + hy' + h^2/2 y''),$$

und den senkrechten Vektoren  $\mathbf{v}_1, \mathbf{v}_2$ , z. B.

$$\overline{\widetilde{\mathbf{P}}_{-1}\mathbf{P}_0} \perp \mathbf{v}_1, \quad \overline{\mathbf{P}_0\widetilde{\mathbf{P}}_1} \perp \mathbf{v}_2$$

$$\mathbf{v}_1 = (y' - h/2 y'', -1), \quad \mathbf{v}_2 = (y' + h/2 y'', -1),$$

$$\mathbf{P}_0 - \widetilde{\mathbf{P}}_{-1} = (h, hy' - h^2/2 y'') = h(1, y' - h/2 y'') \perp (y' - h/2 y'', -1).$$

$\widetilde{M}$  als Schnittpunkt der Geraden  $\mathbf{g}_1$  und  $\mathbf{g}_2$  führt auf die zwei Bedingungen

$$\begin{aligned} \frac{1}{2}(2x - h) + \lambda(y' - \frac{h}{2}y'') &= \frac{1}{2}(2x + h) + \mu(y' + \frac{h}{2}y''), \\ \frac{1}{2}(2y - hy' + \frac{h^2}{2}y'') - \lambda &= \frac{1}{2}(2y + hy' + \frac{h^2}{2}y'') - \mu. \end{aligned}$$

Die zweite Gleichung liefert  $\lambda = \mu - hy'$ . Durch Einsetzen in die erste Gleichung erhält man

$$-\mu y'' - y'^2 - \frac{h}{2}y'y'' = 1.$$

Im Limes  $h \rightarrow 0$  folgt  $\widetilde{M} \rightarrow M$  und  $\mu = (1 + y'^2)/(-y'')$ . Der Schnittpunkt der Geraden  $\mathbf{g}_1$  und  $\mathbf{g}_2$  liegt somit bei

$$M = (M_1, M_2) = (x_0, y_0) = (x + \mu y', y - \mu).$$

Der Radius des gesuchten Kreises ist

$$\begin{aligned} R_K &= [(M_1 - \mathbf{P}_{0,1})^2 + (M_2 - \mathbf{P}_{0,2})^2]^{1/2} \\ &= |\mu| \sqrt{y'^2 + 1} \\ &= \frac{(1 + y'^2)^{3/2}}{|y''|}. \end{aligned} \tag{1.91}$$

#### Definition 1.14 Schmiegekreis mit seinen Eigenschaften

Der Kreis mit dem Mittelpunkt  $M$  und Radius  $R_K$  heißt Schmiegekreis an  $f$  in  $x$ .

$R_K$  ist der **Krümmungsradius**, die inverse Größe

$$K = \frac{1}{R_K} = \frac{|y''|}{(1 + y'^2)^{3/2}} \tag{1.92}$$

die **Krümmung** von  $f$  in  $x$ .

### Glättungseigenschaften kubischer Splines

Als Maß für die Krümmung von  $f$  in  $x$  kann also  $f''(x)$  betrachtet werden. Ein Maß für die Krümmung im Intervall  $[a, b]$  ist die integrale Größe

$$\|f''\|_2 = \left( \int_a^b [f''(x)]^2 dx \right)^{1/2}. \quad (1.93)$$

Diese wird durch gewisse kubische Splines minimiert, wie der nachfolgende Satz zeigt.

**Satz 1.28** Gegeben seien die Unterteilung  $\Delta = \{a = x_0, x_1, \dots, x_{l+1} = b\}$  und die zweimal stetig differenzierbare Funktion  $f : [a, b] \rightarrow \mathbb{R}$ .

$p : [a, b] \rightarrow \mathbb{R}$  sei eine  $f$  interpolierende Funktion, d. h. es gelte  $p(x_i) = f(x_i)$ ,  $i = 0, 1, \dots, l+1$ .

Ist  $s$  ein interpolierender kubischer Spline von  $f$  mit der zusätzlichen Eigenschaft

$$[s''(x)(p'(x) - s'(x))]_{x=a}^b = 0, \quad (1.94)$$

so ist  $\|s''\|_2 \leq \|p''\|_2$ .

**Beweis.** Aus  $p'' = s'' + (p'' - s'')$  folgt

$$\int_a^b (p'')^2 dx = \int_a^b (s'')^2 dx + \underbrace{2 \int_a^b s''(p'' - s'') dx}_{(*)} + \underbrace{\int_a^b (p'' - s'')^2 dx}_{\geq 0} \geq \int_a^b (s'')^2 dx,$$

falls  $(*)$  verschwindet. Mittels partieller Integration und nach der Voraussetzung (1.94) gilt

$$\int_a^b s''(p'' - s'') dx = [s''(p' - s')]_a^b - \int_a^b s'''(p' - s') dx = - \int_a^b s'''(p' - s') dx.$$

Nach Definition ist  $s'''$  stückweise konstant und somit  $s'''(x) = \sigma_i$  für  $x \in (x_i, x_{i+1})$ .

Daher ist

$$\int_a^b s''(p'' - s'') dx = - \sum_{i=0}^l \sigma_i \int_{x_i}^{x_{i+1}} (p' - s') dx = - \sum_{i=0}^l \sigma_i [p(x) - s(x)]_{x_i}^{x_{i+1}} = 0$$

wegen der Interpolationseigenschaft.  $\square$

**Folgerung 1.29** Sei  $s$  ein kubischer Interpolationsspline. Ist  $p$  eine weitere zweimal stetig differenzierbare interpolierende Funktion, und erfüllt  $s$  die ergänzenden natürlichen Randbedingungen oder erfüllen  $s$  und  $p$  gleichzeitig eine der Bedingungen  $s''(a) = s''(b) = 0$  oder  $s'(a) = p'(a)$ ,  $s'(b) = p'(b)$ , so ist  $\|s''\|_2 \leq \|p''\|_2$ .

**Beweis.** Analog zum Beweis des Satzes 1.28.

Nach Voraussetzung gilt  $[s''(x)(p'(x) - s'(x))]_{x=a}^b = 0$ . □

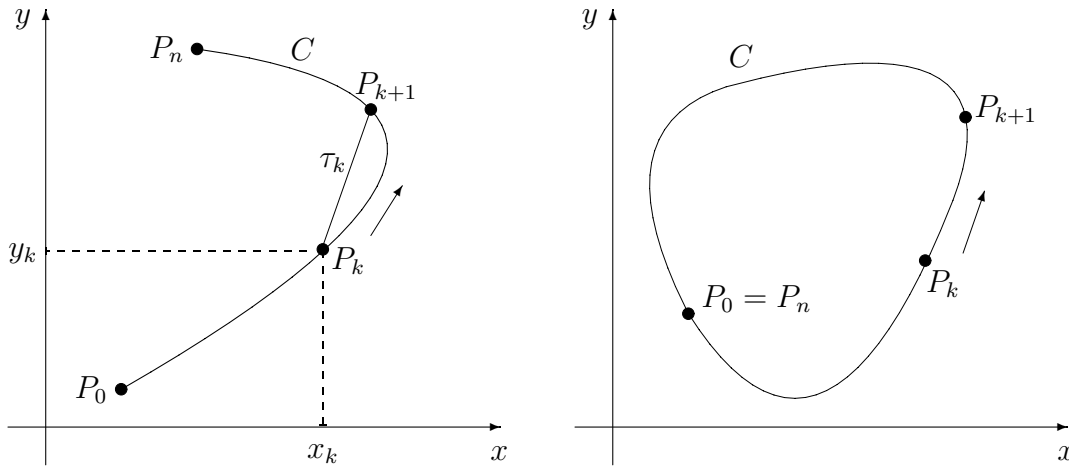
### 1.7.5 Parametrische Splines zur Kurvendarstellung

Ziel ist die Interpolation mit entsprechender Glattheit der punktweise gegebenen Kurve im  $\mathbb{R}^2$

$$C = \{P_k \mid P_k = (x_k, y_k), k = 0, 1, \dots, n\}. \quad (1.95)$$

sowie eine kurze Beschreibung des Algorithmus zur Konstruktion des Splines.  
Die Parameterdarstellung von  $C$  sei

$$x = x(t), y = y(t), \quad a \leq t \leq b. \quad (1.96)$$



**Abb. 1.28** Offene bzw. geschlossene Kurven in der Ebene mit Parametrisierung

#### Algorithmus für parametrische Splines

1. Bestimmung der  $t$ -Werte  $t_k$  zu  $x_k = x(t_k)$ ,  $y_k = y(t_k)$ .

Man berechnet die Approximation der Bogenlänge zwischen  $P_k = (x_k, y_k)$  und  $P_{k+1} = (x_{k+1}, y_{k+1})$  durch die Verbindungsstrecke gemäß

$$\begin{aligned} \tau_k &= t_{k+1} - t_k = \sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2}, \\ t_{k+1} &= t_k + \tau_k, \quad k = 0, 1, \dots, n-1, \quad t_0 = 0. \end{aligned} \quad (1.97)$$



2. Interpolation der zwei Funktionen  $x(t), y(t)$  durch die kubischen Splinefunktionen  $s_x(t)$  bzw.  $s_y(t)$  mit

$$s_x(t_k) = x_k, \quad s_y(t_k) = y_k, \quad k = 0, 1, \dots, n, \quad (1.98)$$

und den Glattheitsforderungen für kubische Splines.

**Bemerkung 1.7** (1) Sinnvoll ist die Approximation offener Kurven durch natürliche Splines sowie die Approximation geschlossener Kurven durch periodische Splines. (2) Die Approximation von Raumkurven  $(x(t), y(t), z(t))$ ,  $a \leq t \leq b$ , kann analog erfolgen unter Einbeziehung der Werte  $z_k$ .

## 1.8 Differentiation

Bei der Berechnung von Ableitungswerten auf der Basis einer gegebenen Referenz verwendet man oft Differenzenquotienten entsprechender Genauigkeitsordnung, um an den Stützstellen Näherungswerte dafür zu gewinnen. Möchte man jedoch noch Ableitungen an Zwischenstellen haben, ist zunächst die Konstruktion einer hinreichend glatten interpolierenden Funktion angebracht, die dann einfach abgeleitet werden kann. Zwei Aufgabenstellungen sollen hier Gegenstand der Betrachtungen sein.

(A1) Gegeben ist  $y = f(x)$  für  $a \leq x \leq b$ .

Gesucht ist eine in der Gestalt einfache Näherungsfunktion für die Ableitung

$$y = g(x) \text{ mit } g(x) \approx f'(x) \text{ für alle } x \in [a, b].$$

(A2) Gegeben sind die Punkte  $(x_k, y_k)$ ,  $k = 0, 1, \dots, n$  (Referenz).

Gesucht sind Näherungen der Ableitungen an den Stützstellen  $x_k$  mit

$$y'_k \approx f'(x_k), \quad y''_k \approx f''(x_k), \dots \text{ für } k = 0, 1, \dots, n.$$

### Differentiation durch kubische Splines

Auf Grund der geringen Schwankungen zwischen den Stützstellen  $x_k$  eignet sich die kubische Splinefunktion  $s(x) = s^{(k)}(x)$  für  $x \in [x_k, x_{k+1}]$  mit

$$s^{(k)}(x) = a_k + b_k(x - x_k) + c_k(x - x_k)^2 + d_k(x - x_k)^3$$

besonders gut zur Approximation der 1. und 2. Ableitung.

### Lösung der Aufgabe (A1)

1. Auswahl von  $n + 1$  Stützstellen  $x_k$  und Berechnung der Werte  $f_k = f(x_k)$ .

2. Ermittlung der (natürlichen, eingespannten, periodischen) kubischen Splinefunktion  $s(x)$ .

3. Differentiation liefert

$$\begin{aligned}s^{(k)}(x)' &= b_k + 2c_k(x - x_k) + 3d_k(x - x_k)^2, \\ s^{(k)}(x)'' &= 2c_k + 6d_k(x - x_k), \quad k = 0, 1, \dots, n-1.\end{aligned}$$

4. Berechnung der 1. und 2. Ableitung an beliebigen Stellen  $x \in [a, b]$ .

### Lösung der Aufgabe (A2)

1. Ermittlung der (natürlichen, eingespannten, periodischen) kubischen Splinefunktion  $s(x)$ .

2. Einsetzen der Stützstellen  $x_k$  in  $s^{(k)}(x)'$ ,  $s^{(k)}(x)''$ ,  $s^{(k)}(x)'''$  liefert

$$\begin{aligned}s^{(k)}(x_k)' &= b_k \approx y'_k, \quad k = 0, 1, \dots, n, \\ s^{(k)}(x_k)'' &= 2c_k \approx y''_k, \\ s^{(k)}(x_k)''' &= 6d_k \approx y'''_k,\end{aligned}$$

wobei man bei der dritten Ableitung des Splines mit Sprüngen rechnen muss.

**Beispiel 1.22**  $y = f(x) = \sin(x)$ ,  $0 \leq x \leq \pi$ .

Die Bestimmung des natürlichen Splines  $s(x)$ ,  $s(x) = s^{(k)}(x)$  für  $x \in [x_k, x_{k+1}]$ ,  $x_k = k\pi/4$ ,  $k = 0, 1, 2, 3, 4$ , ergibt

$$s(x) = \begin{cases} 0 + 0.9977253085x + 0x^2 - 0.1579135105x^3, & x \in [0, \frac{\pi}{4}] \\ 0.7071067812 + 0.7054983314(x - \frac{\pi}{4}) + \\ \quad -0.3720749433(x - \frac{\pi}{4})^2 - 0.0654099177(x - \frac{\pi}{4})^3, & x \in [\frac{\pi}{4}, \frac{\pi}{2}] \\ 1 + 0(x - \frac{\pi}{2}) - 0.5261934310(x - \frac{\pi}{2})^2 + 0.0654099177(x - \frac{\pi}{2})^3, & x \in [\frac{\pi}{2}, \frac{3\pi}{4}] \\ 0.7071067812 - 0.7054983314(x - \frac{3\pi}{4}) + \\ \quad -0.3720749433(x - \frac{3\pi}{4})^2 + 0.1579135105(x - \frac{3\pi}{4})^3, & x \in [\frac{3\pi}{4}, \pi] \end{cases}$$

Damit berechnet man Approximationen der Ableitungen von  $f(x)$  durch  $s'(x)$ ,  $s''(x)$ .

$k$	$s'(x_k)$	$R'(x_k)$	$s''(x_k)$	$R''(x_k)$
0	0.997 725	-0.0023	0	0
1	0.705 498	-0.0016	-0.744 150	-0.037
2	0	0	-1.052 386	-0.052
3	-0.705 498	0.0016	-0.744 150	-0.037

**Tab. 1.6** Näherungswerte  $s'(x_k)$ ,  $s''(x_k)$  und deren Fehler

$$R^{(l)}(x_k) = s^{(l)}(x_k) - f^{(l)}(x_k), \quad l = 1, 2$$

# Literaturverzeichnis

- [1] BERESIN, I. S. und N. P. SHIDKOW: *Numerische Methoden*. Bd. 1,2. DVW Berlin 1970, 1971.
- [2] BODEWIG, E.: *Matrix calculus*. New York 1959.
- [3] COLLATZ, L.: *Funktionalanalysis und numerische Mathematik*. Springer-Verlag Berlin 1964.
- [4] DEMIDOVICH, B. P., I. A. MARON und E. S. SCHUWALOWA: *Numerische Methoden der Analysis*. Math. für Naturwiss. und Technik, Bd. 14. DVW Berlin 1968.
- [5] DEUFLHARD, P. und H. HOHMANN: *Numerische Mathematik*. Walter de Gruyter Berlin 1991.
- [6] DEUFLHARD, P. und H. HOHMANN: *Numerische Mathematik*. 1: Eine algorithmisch orientierte Einführung. 3. überarbeitete und erweiterte Auflage, Lehrbuch. Walter de Gruyter Berlin 2002.
- [7] DIETEL, J.: *Formelsammlung zu Numerischen Mathematik mit Turbo Pascal-Programmen* (TPNUM). Rechenzentrum der RWTH Aachen 1993.
- [8] DONNER, K.: *Skalierung von Matrizen und numerische Stabilität der Gauß-Elimination*. Preprint Universität Passau, MIP-9514 September 1995.
- [9] FADDEJEW, D. K. und W. N. FADDEJEW: *Numerische Methoden der linearen Algebra*. Math. für Naturwiss. und Technik, Bd. 10. DVW Berlin 1973.
- [10] FAIRES, J. D. und R. L. BURDEN: *Numerische Methoden. Näherungsverfahren und ihre praktische Anwendung*. Spektrum Akademischer Verlag Heidelberg 1994.
- [11] GANTMACHER, F.R.: *Teorija matric*. Moskwa 1954.
- [12] GOLUB, G. H. und C. VAN LOAN: *Matrix Computations*. The John Hopkins University Press Baltimore 1989.
- [13] HACKBUSCH, W.: *Iterative Lösung großer schwachbesetzter Gleichungssysteme*. Leitfäden der angewandten Mathematik und Mechanik Band 69. B. G. Teubner Stuttgart 1991, 1993.
- [14] HANKE-BOURGEOIS, M.: *Grundlagen der Numerischen Mathematik und des wissenschaftlichen Rechnens*. Mathematische Leitfäden. B. G. Teubner GmbH, Stuttgart 2002.
- [15] HÄMMERLIN G. und K.-H. HOFFMANN: *Numerische Mathematik*. Grundwissen Mathematik 7. Springer-Verlag Berlin 1991, 1994.

- [16] HERMANN, M.: *Numerische Mathematik*. R. Oldenbourg Verlag München 2001.
- [17] HUCKLE, T. und S. SCHNEIDER: *Numerik für Informatiker*. Springer-Verlag 2002.
- [18] ISAACSON, E. und H. B. KELLER: *Analyse numerischer Verfahren*. Edition Leipzig 1972.
- [19] KIELBASINSKI, A. und H. SCHWETLICK: *Numerische lineare Algebra*. DVW Berlin 1988.
- [20] KOSE, K., R. SCHRÖDER und K. WIELICZEK: *Numerik sehen und verstehen*. Ein kombiniertes Lehr- und Arbeitsbuch mit Visualisierungssoftware. Vieweg Braunschweig 1992.
- [21] LAX, P. D.: *Linear Algebra*. Pure and Applied Mathematics. John Wiley & Sons New York 1997.
- [22] LOCHER, F.: *Numerische Mathematik - für Informatiker*. Springer-Verlag Berlin 1992.
- [23] LUETKEPOHL, H.: *Handbook of matrices*. John Wiley & Sons Chichester 1996.
- [24] MAESS, G.: *Vorlesungen über Numerische Mathematik I, II*. Akademie-Verlag Berlin 1984, 1988. Verfahren. Vieweg Braunschweig 1999.
- [25] MEIS, TH. und U. MARCOWITZ: *Numerische Behandlung partieller Differentialgleichungen*. Springer-Verlag Berlin 1978.
- [26] MEISTER, A.: *Numerik linearer Gleichungssysteme*. Ein Einführung in moderne
- [27] MOHR, R.: *Numerische Methoden in der Technik*. Ein Lehrbuch mit Matlab-Routinen. Vieweg Wiesbaden 1998.
- [28] NEUNDORF, W.: *Numerische Mathematik - Vorlesungen, Übungen, Algorithmen und Programme*. Shaker Verlag Aachen 2002.
- [29] OEVEL, W.: *Einführung in die Numerische Mathematik*. Spektrum Akademischer Verlag Heidelberg 1996.
- [30] OPFER, G.: *Numerische Mathematik für Anfänger*. Vieweg Studium Grundkurs Mathematik Wiesbaden 1993, 3. überarbeitete und erw. Auflage 2001.
- [31] PAULIN, G. und E. GRIEPENTROG: *Numerische Verfahren der Programmiertechnik*. Verlag Technik Berlin 1975.
- [32] PLATO, R.: *Numerische Mathematik kompakt*. Grundlagenwissen für Studium und Praxis. Vieweg Wiesbaden 2000.
- [33] QUARTERONI, A., SACCO, R. und F. SALERI: *Numerische Mathematik 1,2*. Springer-Verlag 2002.
- [34] RALSTON, A.: *A First Course in Numerical Analysis*. McGraw-Hill New York 1965.
- [35] REIMER, M.: *Grundlagen der Numerischen Mathematik*. I und II. AULA-Verlag Wiesbaden 1980, 1982.
- [36] RICE, J. R.: *Numerical Methods, Software and Analysis*. 2nd Edition. Academic Press Inc. Boston 1993.

- [37] ROOS, H.-G. und H. SCHWETLICK: *Numerische Mathematik*. Das Grundwissen für jedermann. B. G. Teubner Stuttgart 1999.
- [38] RUTISHAUSER, H.: *Vorlesungen über Numerische Mathematik*. Bd. 1,2. Birkhäuser Verlag Basel 1976.
- [39] SAMARSKIJ, A. A.: *Theorie der Differenzenverfahren*. Akademische VG Geest & Porzig K.-G. Leipzig 1984.
- [40] SCHABACK, R. und H. WERNER: *Numerische Mathematik*. Springer-Verlag Berlin 1993.
- [41] SCHUPPAR, B.: *Elementare Numerische Mathematik*. Eine problemorientierte Einführung für Lehrer und Studierende. Vieweg Wiesbaden 1998.
- [42] SCHWARZ, H. R.: *Numerische Mathematik*. B. G. Teubner Stuttgart 1988, 1997.
- [43] SCHWARZ, H. R., H. RUTISHAUSER und E. STIEFEL: *Numerik symmetrischer Matrizen*. B. G. Teubner Stuttgart 1972.
- [44] SCHWETLICK, H. und H. KRETZSCHMAR: *Numerische Verfahren für Naturwissenschaftler und Ingenieure*. Fachbuchverlag Leipzig 1991.
- [45] SPÄTH, H.: *Numerik*. Vieweg Wiesbaden 1994.
- [46] STOER, J.: *Numerische Mathematik I*. Heidelberger Taschenbücher 105. Springer-Verlag Berlin 1993.
- [47] STOER, J. und R. BULIRSCH: *Einführung in die Numerische Mathematik II*. Heidelberger Taschenbücher 114. Springer-Verlag Berlin 1990.
- [48] STOER, J. und R. BULIRSCH: *Numerical mathematics 2*. An Introduction - under consideration of lectures by F. L. Bauer. 4. neu bearbeitete und erweiterte Auflage. Springer-Verlag Berlin 2000.
- [49] ÜBERHUBER, C.: *Computer-Numerik 1,2*. Springer-Verlag Berlin 1995.
- [50] VARGA, R.S.: *Matrix iterative analysis*. Prentice Hall, Englewood Cliffs, N. J., 1962.
- [51] WELLER, F.: *Numerische Mathematik für Ingenieure und Naturwissenschaftler*. Eine Einführung für Studium und Praxis. Vieweg Braunschweig 1996.
- [52] WERNER, J.: *Numerische Mathematik 1, 2*. Vieweg Studium Aufbaukurs Mathematik 1992.
- [53] WILKINSON, J. H. und C. REINSCH: *Linear Algebra*. Handbook for automatic computation, Vol. II. Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen, Bd. 186. Springer-Verlag Berlin 1971.
- [54] ZLATEV, Z.: *Computational Methods for General Sparse Matrices*. Math. and Its Appl. Vol.65. Kluwer Academic Publishers London 1991.
- [55] ZURMÜHL, R.: *Praktische Mathematik für Ingenieure und Physiker*. Springer-Verlag Berlin 1965.
- [56] ZURMÜHL, R. und S. FALK: *Matrizen und ihre Anwendungen*. Teil 2, Numerische Methoden. Springer-Verlag Berlin 1984.

# Index

- Approximation
  - stetige Approximation im Mittel, 58
    - Fourier-Koeffizienten, 58
    - Fourier-Polynom, 58
- baryzentrische Koordinaten, 18
- Differenz, 19
  - dividierte, 19
  - Rückwärtsdifferenz, 19
  - Steigung, 20
  - Vorwärtsdifferenz, 19
- diskrete Fourier-Transformation, 55
  - Butterfly-Algorithmus, 55
  - Fourier-Koeffizienten, 55
  - schnelle Fourier-Transformation, 55
- Gibbsscher Effekt, 39
- Krümmung einer Kurve, 89
  - Krümmung, 90
  - Krümmungsradius, 89
  - Schmiegekreis, 89
- Laplacescher Entwicklungssatz, 7
- Matrix
  - Vandermondesche Matrix, 7, 51
- Norm eines linearen Raums, 11
  - Maximumnorm, 12
- orthogonale Funktionensysteme
  - Tschebyscheff-Polynome, 45
  - Minimax-Eigenschaft, 45
- Partialbruchzerlegung, 18
- Polynom, 1
  - Deflation, 2
- einfaches Horner-Schema, 1
- Horner-Schema, 1
  - einfaches, 1
  - inverses, 3
  - vollständiges, 2
- Newtonsche Form, 3
- Normalform, 1
- reduzierte, 2
- Polynominterpolation im  $\mathbb{R}^1$ , 5
  - Aitken-Interpolation, 23, 29
    - Schema von Aitken, 29
    - Schema von Aitken-Neville, 29
  - allgemeine lineare Interpolation, 6
  - allgemeine Referenz, 35
  - Basisfunktionen, 5
  - Haarsche Determinante, 7
  - Hermite-Interpolation, 31
    - Interpolationspolynom, 32
    - kubische Hermite-Basis, 36
  - Interpolationsaufgabe, 6, 31
  - Interpolationsfehler, 6, 8, 21, 32
  - Kondition, 11
  - Konvergenz, 38
    - Konvergenzsätze, 43
  - Lagrange-Interpolation, 8
    - baryzentrische Darstellung, 18
    - Interpolationspolynom, 8
    - Lagrange-Basispolynome, 8, 26
  - Newton-Interpolation, 19
    - Interpolationspolynom, 21
    - Interpolationspolynom von Newton-Gregory, 25
    - Newton-Basispolynome, 21, 26
    - Schema der dividierten Differenzen, 20

- Referenz, 5, 8
  - Tschebyscheff-Referenz, 44
- Stützstelle, 5
- Stützwert, 5
- Taylor-Interpolation, 35
- Vandermondesche Matrix, 7
- Polynominterpolation im  $\mathbb{R}^2$ , 6
- trigonometrische Interpolation, 6, 47
  - Basisfunktionen, 50
  - Faltungsalgorithmus, 49
  - komplexe Einheitswurzeln, 48
  - komplexes Interpolationspolynom, 50
  - reelles Interpolationspolynom, 51
- Satz
  - MARSDEN-Identität, 78
  - WEIERSTRASSSCHER Approximationssatz, 43
  - Satz von FABER, 43
  - Satz von HOLLADAY zur Optimalität kubischer Splines, 73
  - Satz von ROLLE, 9
- Splineinterpolation im  $\mathbb{R}^1$ , 63
  - B-Splines, 74
    - MARSDEN-Identität, 78
    - abgebrochene B-Splines, 86
    - erweiterte Stützstellenfolge, 78, 80
    - globale kubische Basissplines, 87
    - Grad, 74
    - Hütchenfunktionen, 84
    - Ordnung, 74
    - Raum der Splines, 74
    - Zerlegung der Eins, 79
  - Glättungseigenschaften kubischer Splines, 91
  - kubische Splines, 70
    - eingespannter Spline, 71
    - Existenz und Eindeutigkeit, 73
    - Konstruktion, 71
    - natürlicher Spline, 70
    - Optimalität, 73
    - periodischer Spline, 71
    - Spline mit Not-a-knot-Bedingung, 71
  - linearer Spline, 65
  - quadratische Splines, 65
  - Typen von Splines, 65
- Splineinterpolation im  $\mathbb{R}^2$ , 92
  - parametrische Splines, 92

**Anschrift:**

Dr. rer. nat. habil. Werner Neundorf  
Technische Universität Ilmenau, Institut für Mathematik  
PF 10 05 65  
D - 98684 Ilmenau

E-mail : [werner.neundorf@tu-ilmenau.de](mailto:werner.neundorf@tu-ilmenau.de)  
Homepage : [http://www.mathematik.tu-ilmenau.de/~neundorf/index\\_de.html](http://www.mathematik.tu-ilmenau.de/~neundorf/index_de.html)