Numerik II – Numerische Analysis

Prof. Dr. Gerlind Plonka-Hoch

Wintersemester 07/08

Inhaltsverzeichnis

1 Interpolation

Idee: Zu einer Funktion $f(x)$ finde man ein Polynom (oder eine andere gut handhabbare Funktion), das mit $f(x)$ an gewissen Stellen übereinstimmt.

Anwendung: Konstruktion von Zwischenwerten für eine Funktion, von der nur einige Funktionswerte bekannt sind.

1.1 Algebraische Interpolation

Sei Π_n die Menge aller Polynome p vom Grad $\leq n$ der Form

$$p(x) = a_0 + a_1 x + \ldots + a_n x^n, \quad a_j \in \mathbb{R}(\mathbb{C}).$$

Interpolationsproblem 1.1. *Gegeben: $n+1$ paarweise verschiedene Punkte $x_0, \ldots, x_n \in \mathbb{R}$, die* **Stützstellen** *oder* **Knoten** *genannt werden, sowie $n+1$ zugehörige Werte $y_0, \ldots, y_n \in \mathbb{R}$,* **Funktions-** *oder* **Stützwerte** *genannt.*
Gesucht: $p \in \Pi_n$, das die Interpolationsbedingungen

$$p(x_k) = y_k, \qquad k = 0, \ldots, n, \tag{1.1}$$

erfüllt.

Satz 1.2. *Das Interpolationsproblem 1.1 besitzt genau eine Lösung.*

Beweis. 1) Existenz: Das Polynom $p \in \Pi_n$ hat die Form

$$p(x) = \sum_{j=0}^{n} a_j x^j$$

und ist durch $a_0, \ldots, a_n$ eindeutig bestimmt. Aus den Interpolationsbedingungen (1.1) folgt

$$p(x_k) = \sum_{j=0}^{n} a_j x_k^j = y_k, \quad k = 0, \ldots, n.$$

Wir erhalten ein lineares Gleichungssystem (LGS) der Form

$$\underbrace{\begin{pmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ \vdots & & & & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{pmatrix}}_{\text{Vandermonde-Matrix } \mathbf{V}=(x_j^k)_{j,k=1}^n} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}.$$

Wegen $\det \mathbf{V} = \prod_{0 \le j < k \le n} (x_k - x_j) \neq 0$ besitzt das System eine eindeutige Lösung.

2) Die Eindeutigkeit folgt aus der Tatsache, dass $1, x, \ldots, x^n$ eine Basis von Π_n bilden. Denn angenommen, für $p_1, p_2 \in \Pi_n$ ist

$$p_1(x_k) = p_2(x_k) = y_k, \qquad k = 0, \ldots, n,$$

erfüllt. Das bedeutet

$$p_1(x_k) - p_2(x_k) = 0, \quad k = 0, \ldots, n,$$

d.h., das Polynom $q(x) := p_1(x) - p_2(x)$ hat mindestens $n+1$ Nullstellen. Wegen $q \in \Pi_n$ folgt $q \equiv 0$, und damit ist $p_1 \equiv p_2$. □

Alle Elemente von Π_n besitzen höchstens n verschiedene Nullstellen.
→ Verallgemeinerung möglich:

Definition 1.3. *Sei U ein (linearer) Untervektorraum (UVR) von $C[a,b]$ mit $\dim U = n$. Hat jedes Element $u \in U, u \neq 0$, in $[a,b]$ höchstens $n-1$ verschiedene Nullstellen, so heißt U* **Haarscher Raum**. *Eine Basis $\{u_1, \ldots, u_n\}$ eines Haarschen Raumes heißt* **Tschebyscheff-System**.

Satz 1.4. *Ist U ein UVR von $C[a,b]$ mit $\dim U = n$ so sind folgende Aussagen äquivalent:*

(i) U ist Haarscher Raum.

(ii) Zu gegebenen Wertepaaren (x_k, y_k), $k = 1, \ldots, n, (x_k \in [a,b]$ paarweise verschieden) existiert genau ein $u \in U$ mit $u(x_k) = y_k$, $k = 1, \ldots, n$.

(iii) Für je n paarweise verschiedene Punkte $x_1, \ldots x_n \in [a,b]$ und jede Basis $\{u_1, \ldots, u_n\}$ von U gilt $\det(u_j(x_k))_{j,k=1}^n \neq 0$.

Beweis. (i) ⇒ (ii): Sei U Haarscher Raum mit $\dim U = n$. Wähle Basis $\{u_1, \ldots, u_n\}$ von U. Dann ist jedes $u \in U$ eindeutig darstellbar als $u = \sum_{j=1}^n a_j u_j$ mit $a_j \in \mathbb{R}$. Die Interpolationsbedingungen liefern dann das LGS

$$u(x_k) = \sum_{j=1}^{n} a_j u_j(x_k) = y_k, \qquad k = 1, \ldots, n.$$

mit der Koeffizientenmatrix $(u_j(x_k))_{k,j=1}^n$. Diese hat jedoch vollen Rang n, denn: Gäbe es eine Basisfunktion u_j, so dass

$$u_j(x_k) = \sum_{\substack{l=1 \\ l\neq j}}^{n} b_l\, u_l(x_k), \qquad k = 1, \ldots, n,$$

so hätte $u_j(x) - \sum_{\substack{l=1 \\ l\neq j}}^{n} b_l u_l(x)$ mindestens n Nullstellen und wäre damit die Null-Funktion. Dann wäre u_j als Linearkombination der anderen Basisfunktionen darstellbar. Das ist ein Widerspruch zur Annahme, dass $\{u_1, \ldots, u_n\}$ eine Basis ist.
(ii) $\Rightarrow$ (iii), (iii) $\Rightarrow$ (i): in den Übungen. □

Beispiele:

1. Die Monome $\{1, x, \ldots, x^n\}$ bilden ein Tschebyscheff-System auf jedem Intervall $[a, b] \subset \mathbb{R}$.
2. $\{1, e^x, \ldots, e^{nx}\}$ bilden ein Tschebyscheff-System auf jedem Intervall $[a, b] \subset \mathbb{R}$.
3. $\{1, \sin x, \cos x, \ldots, \sin nx, \cos nx\}$ bilden ein Tschebyscheff-System auf jedem reellen halboffenen Intervall der Länge 2π.

Definition 1.5. *Es seien $n+1$ paarweise verschiedene Punkte $x_k \in \mathbb{R}, k = 0, \ldots, n$, gegeben. Dann heißen*

$$l_j(x) := \prod_{\substack{k=0 \\ k\neq j}}^{n} \frac{x - x_k}{x_j - x_k} \quad \textit{für } j = 0, \ldots, n,$$

die zu diesem Knoten gehörenden **Lagrange-Grundpolynome**.

Es gilt

$$l_j(x_k) = \delta_{jk} = \begin{cases} 1, & j = k \\ 0, & j \neq k \end{cases}.$$

Setzen wir $w_{n+1}(x) := \Pi_{k=0}^n (x - x_k)$, so erhalten wir eine neue Darstellung von $l_j(x)$:

$$\begin{aligned} l_j(x) &= \frac{w_{n+1}(x)}{(x - x_j)} \frac{1}{\prod_{\substack{k=0 \\ k\neq k}}^{n} (x_j - x_k)} = \frac{w_{n+1}(x)}{(x - x_j)} \lim_{x \to x_j} \frac{(x - x_j)}{w_{n+1}(x)} \\ &= \frac{w_{n+1}(x)}{(x - x_j)} \lim_{x \to x_j} \frac{1}{w'_{n+1}(x)} = \frac{w_{n+1}(x)}{(x - x_j) w'_{n+1}(x_j)}. \end{aligned}$$

Satz 1.6. *Die eindeutige Lösung des Interpolationsproblems 1.1 lässt sich in der* **Lagrange-Form**

$$p(x) = \sum_{j=0}^{n} y_j \; l_j(x) = \sum_{j=0}^{n} \; y_j \cdot \prod_{\substack{k=0 \\ k \neq j}}^{n} \frac{x - x_k}{x_j - x_k} \tag{1.2}$$

darstellen.

Beweis. Wegen $l_j(x_k) = \delta_{jk}, \; 0 \leq j, k \leq n$, folgt

$$p(x_k) = \sum_{j=0}^{n} y_j l_j(x_k) = y_k, \; k = 0, \ldots, n.$$

□

Die Lagrange-Form ist für numerische Berechnungen kaum geeignet falls n groß ist.

Beispiel: $n = 2$ Gegeben:

x_k	0	1	3
y_k	1	3	2

Gesucht: $p(2)$, wobei $p(x_k) = y_k$ für $k = 0, 1, 2$.

Wir erhalten:

$$l_0(x) = \frac{(x-1)(x-3)}{(0-1)(0-3)}, \;\; l_1(x) = \frac{(x-0)(x-3)}{(1-0)(1-3)}, \; l_2(x) = \frac{(x-0)(x-1)}{(3-0)(3-1)}$$

Also folgt

$$p(2) = 1 \cdot l_0(2) + 3 \cdot l_1(2) + 2 \cdot l_2(2) = 1 \cdot \frac{(-1)}{3} \; + 3 \cdot 1 + 2 \cdot \frac{1}{3} \; = \frac{10}{3}.$$

Rekursive Berechnung

Gegeben sind $(x_k, y_k), k = 0, 1, \ldots, n$, mit $x_k \in [a, b]$ paarweise verschieden.
Wähle $k_0, \ldots, k_r \in \{0, 1, \ldots, n\}$ paarweise verschieden. Sei $p_{k_0,\ldots,k_r} \in \Pi_r$ das Interpolationspolynom, das die Interpolationsbedingungen an den Stellen $x_{k_0}, \ldots, x_{k_r}$ erfüllt, d.h.

$$p_{k_0,\ldots,k_r}(x_{k_j}) = y_{k_j}, \qquad j = 0, \ldots, r.$$

Lemma 1.7. *Es gilt die Rekursionsformel*

(i) $p_k(x) \equiv y_k, \qquad k \in \{0, \ldots, n\},$

(ii) $$p_{k_0,\ldots,k_r}(x) = \frac{(x - x_{k_0})p_{k_1,\ldots,k_r}(x) - (x - x_{k_r})p_{k_0,\ldots,k_{r-1}}(x)}{x_{k_r} - x_{k_0}}.$$

Beweis. (i) Wegen $p_k \in \Pi_0$ ist p_k eine konstante Funktion mit

$$p_k(x_k) = y_k \quad \Rightarrow\ p_k(x) \equiv y_k.$$

(ii) Sei

$$p(x) = \frac{(x - x_{k_0})p_{k_1,\ldots,k_r}(x) - (x - x_{k_r})p_{k_0,\ldots,k_{r-1}}(x)}{(x_{k_r} - x_{k_0})}.$$

Dann folgt $p(x) \in \Pi_r$, denn $p_{k_1,\ldots,k_r}, p_{k_0,\ldots,k_{r-1}} \in \Pi_{r-1}$. Weiter gilt

$$\begin{aligned} p(x_{k_0}) &= \frac{0 - (x_{k_0} - x_{k_r})\overbrace{p_{k_0,\ldots,k_{r-1}}(x_{k_0})}^{y_{k_0}}}{(x_{k_r} - x_{k_0})} = y_{k_0}, \\ p(x_{k_r}) &= \frac{(x_{k_r} - x_{k_0})p_{k_1,\ldots,k_r}(x_r) - 0}{(x_{k_r} - x_{k_0})} = y_{k_r}, \end{aligned}$$

und für $j = 1, 2, \ldots, r - 1$,

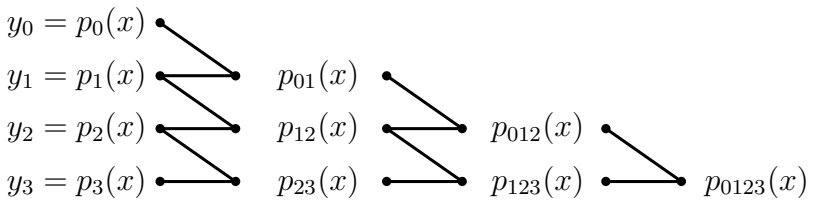
$$p(x_{k_j}) = \frac{(x_{k_j} - x_{k_0})y_{k_j} - (x_{k_j} - x_{k_r})y_{k_j}}{(x_{k_r} - x_{k_0})} = y_{k_j}\ .$$

Somit folgt $p(x_{k_j}) = y_{k_j}$ für $j = 0, \ldots, r$. Wegen Satz 1.2. folgt $p = p_{k_0,\ldots,k_r}$. □

Algorithmus von Neville

Gegeben: $(x_k, y_k),\ k = 0, \ldots, n$, mit $x_k \in [a, b]$ paarweise verschieden, $x \in [a, b]$ fest.
Gesucht: $p(x)$ wobei $p \in \Pi_n$ die Interpolationsbedingungen $p(x_k) = y_k,\ k = 0, \ldots, n$ erfüllt.

Symmetrisches Dreiecksschema:



Für $p_{123}(x)$ gilt z.B.

$$p_{123}(x) = \frac{(x - x_1)p_{23}(x) - (x - x_3)p_{12}(x)}{x_3 - x_1}.$$

Beispiel:
Gegeben:

x_k	0	1	3
y_k	1	3	2

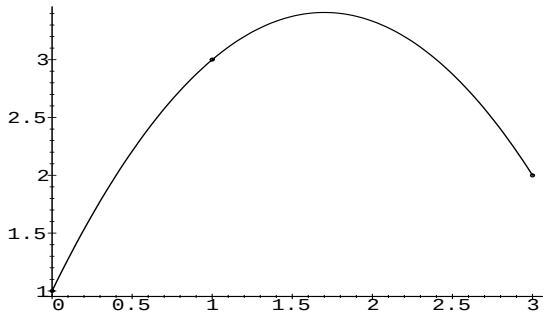
Gesucht: Funktionswert an der Stelle $x = 2$.

	$r = 0$		$r = 1$		$r = 2$
$x_0 = 0$	$p_0(2) = 1$				
		$\ddots$			
$x_1 = 1$	$p_1(2) = 3$	$\dots$	$p_{01}(2) = 5$		
		$\ddots$		$\ddots$	
$x_2 = 3$	$p_2(2) = 2$	$\dots$	$p_{12}(2) = \frac{5}{2}$	$\dots$	$p_{012}(2) = \frac{10}{3}$

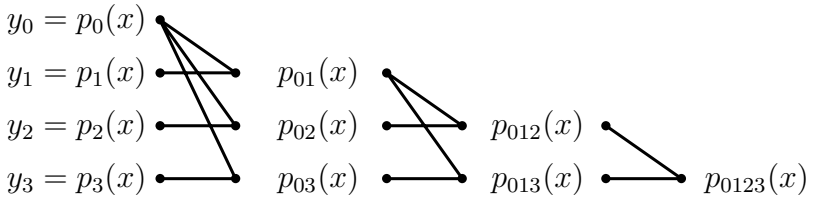
$$\begin{aligned} p_{01}(2) &= \tfrac{(2-0)\cdot 3-(2-1)\cdot 1}{1-0} = 5, \quad p_{12}(2) = \tfrac{(2-1)\cdot 2-(2-3)\cdot 3}{3-1} = \tfrac{5}{2}, \\ p_{012}(2) &= \tfrac{(2-0)\cdot \frac{5}{2} -(2-3)\cdot 5}{3-0} = \tfrac{10}{3} \end{aligned}$$

Aufwand: 4 Subtraktionen, 2 Multiplikationen, 1 Division pro "Zwischenpolynom".
Für $n + 1$ Knoten benötigen wir $\sum_{j=1}^{n} j = \frac{(n+1)n}{2}$ „Zwischenpolynome", d.h., insgesamt sind also $2(n + 1)n$ Additionen/Subtraktionen und $\frac{3}{2}\, n(n + 1)$ Multiplikationen/Divisionen zur Berechnung eines Wertes des Interpolationspolynoms nötig.

Bemerkung:
(i) Alternativ kann man die **Methode von Aitken** verwenden, die auf einem unsymmetrischen Tableau basiert:



Quadratisches Interpolationspolynom (siehe Beispiel)



(ii) Der Algorithmus eignet sich nicht gut zur Berechnung der Koeffizienten des Interpolationspolynoms.

1.2 Newton-Darstellung des Interpolationspolynoms

Idee: Wählt man die Monome $\{1, x, \ldots, x^n\}$ als Basis von Π_n, so liefert das Interpolationsproblem 1.1 ein LGS mit der Vandermonde-Matrix $\mathbf{V} = (x_k^j)_{k,j=0}^n$ als Koeffizientenmatrix. Wählt man die Lagrange-Grundpolynome als Basis von Π_n, so liefert das Interpolationsproblem 1.1 ein LGS mit der Einheitsmatrix als Koeffizientenmatrix. $((l_j(x_k))_{k,j=0}^n = \mathbf{I}_{n+1})$.

Gesucht: Basis $\{u_j, j = 0, \ldots, n\}$ so, dass $(u_j(x_k))_{k,j=0}^n$ eine Dreiecksmatrix ist.

Definition 1.8. *Die* **Newton-Grundpolynome** *seien rekursiv durch*

$$\begin{aligned} u_0(x) &:= 1 \\ u_j(x) &:= (x - x_{j-1})u_{j-1}(x), \quad j = 1, \ldots, n, \end{aligned}$$

bzw. explizit durch $u_j(x) := \prod_{k=0}^{j-1}(x - x_k)$, $j = 1, \ldots, n$, *definiert.*

Wir erhalten

$$u_j(x_k) = \begin{cases} 0 & , k < j, \\ \prod_{r=0}^{j-1} (x_k - x_r) & , k \geq j. \end{cases}$$

Mit dem Ansatz $p(x) = \sum_{j=0}^{n} a_j u_j(x) \in \Pi_n$ erhalten wir aus den Interpolationsbedingungen $p(x_k) = y_k, k = 0, \ldots, n$ das LGS

$$\begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ 1 & x_1 - x_0 & 0 & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & & \ddots & 0 \\ 1 & x_n - x_0 & \ldots & \ldots & \prod_{l=0}^{n-1}(x_n - x_l) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}.$$

Die Berechnung der Koeffizienten a_j kann nun durch Vorwärtselimination erfolgen. Wir zeigen, dass die a_j sogenannte dividierte Differenzen sind.

Definition 1.9. *Sei $p_j \in \Pi_j$ das Interpolationspolynom zu den paarweise verschiedenen Stützstellen $x_0, \ldots, x_j$ und $p_j(x_k) = y_k$ für $k = 0, \ldots, j$. Dann heißt der Koeffizient der höchsten Potenz x^j von p_j* **dividierte Differenz j-ter Ordnung**.
Bezeichnung: $[y_0, \ldots, y_j] = [p_j(x_0), \ldots, p_j(x_j)] = p_j[x_0, \ldots, x_j]$.
(Falls für eine Funktion f gilt $f(x_k) = y_k,\ k = 0, \ldots, j$, schreibt man auch $[f(x_0), \ldots, f(x_j)]$ oder $f[x_0, \ldots, x_j]$).

Satz 1.10. *Für die dividierte Differenz $[y_0, \ldots, y_j]$ zu den paarweise verschiedenen Stützstellen x_k und den Stützwerten $y_k, k = 0, \ldots, j$, gilt:*

(i) $$[y_0, y_1, \ldots, y_j] = \sum_{r=0}^{j} y_r \prod_{\substack{k=0 \\ k \neq r}}^{j} \frac{1}{x_r - x_k}.$$

(ii) $[y_0, \ldots, y_j]$ *ist invariant gegenüber Permutation der Wertepaare* $(x_0, y_0), \ldots, (x_j, y_j)$.

(iii) $[y_0, \ldots, y_j]$ *lässt sich rekursiv berechnen:*

$$[y_i] := y_i, \quad i = 0, \ldots, j,$$

$$[y_i, \ldots, y_{i+l}] := \frac{[y_{i+1}, \ldots, y_{i+l}] - [y_i, \ldots, y_{i+l-1}]}{x_{i+l} - x_i}, \quad i = 0, 1, \ldots, j-l;\ l = 1, \ldots, j.$$

(iv) Es gilt $[y_0, \ldots, y_j, y_{j+1}] \ = p_j[x_0, \ldots, x_{j+1}] \ = 0$, *falls* $p_j \in \Pi_j$ *und* $p_j(x_k) = y_k,\ k = 0, \ldots, j+1$.

Beweis. (i) Aus der Lagrange-Darstellung von p_j

$$p_j(x) = \sum_{l=0}^{j} \ y_l \prod_{\substack{k=0 \\ k \neq l}}^{j} \frac{x - x_k}{x_l - x_k}$$

ergibt sich für den Koeffizienten von x^j:

$$a_j = \sum_{l=0}^{j} y_l \prod_{\substack{k=0 \\ k \neq l}}^{j} \frac{1}{x_l - x_k}.$$

Aus Definition 1.9 folgt damit die Behauptung.

(ii) Die Aussage ist klar, da das Interpolationspolynom p_j und damit auch sein Höchstkoeffizient eindeutig bestimmt ist.

(iii) Sei $p \in \Pi_{l-1}$ das Interpolationspolynom durch die l Punkte $(x_r, y_r), r = i, \ldots, i+l-1$, und $q \in \Pi_{l-1}$ das Interpolationspolynom mit $q(x_r) = y_r,\ r = i+1, \ldots, i+l$. Die Leitkoeffizienten von p und q sind $[y_i, \ldots, y_{i+l-1}]$ und $[y_{i+1} \ \ldots, y_{i+l}]$. Betrachte das Polynom $w \in \Pi_l$,

$$w(x) := \frac{1}{(x_{i+l} - x_i)}[(x - x_i)q(x) - (x - x_{i+l})p(x)], \qquad x \in \mathbb{R}.$$

Dann folgt

$$w(x_i) = \frac{-(x_i - x_{i+l})}{(x_{i+l} - x_i)} p(x_i) = y_i, \quad w(x_{i+l}) \ = \frac{x_{i+l} - x_i}{(x_{i+l} - x_i)} q(x_{i+l}) = y_{i+l},$$

und

$$w(x_j) = y_j, \quad j = i+1, \ldots, i+l-1.$$

Also ist $[y_i, \ldots, y_{i+l}]$ Höchstkoeffizient von $w(x)$. Somit hat der Leitkoeffizient von $w(x)$ die Form $\frac{1}{(x_{i+l} - x_i)}$ $([y_{i+1}, \ldots y_{i+l}] - [y_i \ldots y_{i+l-1}])$.

(iv) $[y_0, \ldots, y_{j+1}]$ ist Höchstkoeffizient des Interpolationspolynoms $p \in \Pi_{j+1}$ mit $p(x_k) = y_k, k = 0, \ldots, j+1$. Wegen Satz 1.2 ist jedoch $p = p_j$ und daher der Koeffizient der höchsten Potenz x^{j+1} gleich Null. □

Satz 1.11. *Das Interpolationspolynom* $p_n \in \Pi_n$, *das die Bedingungen* $p_n(x_k) = y_k,\ k = 0, \ldots, n$, *erfüllt, hat die Darstellung*

$$p_n(x) = [y_0] + [y_0, y_1](x - x_0) + \ldots + [y_0, \ldots, y_n] \ (x - x_0) \ldots (x - x_{n-1}).$$

Beweis. Sei

$$p_n(x) = \sum_{j=0}^{n} a_j u_j(x), \qquad u_j(x) := \prod_{r=0}^{j-1} (x - x_r).$$

Dann ist a_n der Höchstkoeffizient von p_n, also $a_n = [y_0, \ldots, y_n]$. Wegen $u_n(x_k) = 0,\ k = 0, \ldots, n-1$, gilt für das Polynom $p_{n-1}(x) = \sum_{j=0}^{n-1} a_j u_j(x)$ nun

$$p_{n-1}(x_k) = p_n(x_k) - a_n u_n(x_k) = y_k, \quad k = 0, \ldots, n-1,$$

d.h., $p_{n-1} \in \Pi_{n-1}$ ist Interpolationspolynom für $(x_k, y_k),\ k = 0, \ldots, n-1$.
Also $a_{n-1} = [y_0, \ldots, y_{n-1}]$ usw. □

Beispiel: $n = 2$

x_k	0	1	3
y_k	1	3	2

	$k=0$		$k=1$		$k=2$
$x_0 = 0$	$[y_0] = 1$				
		$\ddots$			
$x_1 = 1$	$[y_1] = 3$	$\ldots$	$[y_0, y_1] = 2$		
		$\ddots$		$\ddots$	
$x_2 = 3$	$[y_2] = 2$	$\ldots$	$[y_1, y_2] = -\frac{1}{2}$	$\ldots$	$[y_0, y_1, y_2] = -\frac{5}{6}$

$$[y_0, y_1] = \frac{y_1 - y_0}{x_1 - x_0} = \frac{3-1}{1-0} = 2, \quad [y_1, y_2] = \frac{y_2 - y_1}{x_2 - x_1} = \frac{(-1)}{2}, \quad \text{usw..}$$

$$\Rightarrow \quad p_{012}(x) = 1 + 2(x-0) - \tfrac{5}{6}\,(x-0)(x-1) = 1 + (x-0)\cdot(2 - \tfrac{5}{6}\,(x-1))$$

$$\Rightarrow \quad p_{012}(2) = 1 + (2-0)\cdot(2 - \tfrac{5}{6}\,\cdot(2-1)) = \tfrac{10}{3}.$$

Numerische Berechnung
Unter Verwendung der Newtonschen Interpolationsformel kann das Interpolationspolynom mit einer Art Hornerschema berechnet werden:

$$p_n(x) = [y_0] + (x - x_0)([y_0, y_1] + (x - x_1)\cdot([y_0, y_1, y_2] + \ldots + \\ + (x - x_{n-1})[y_0, \ldots, y_n])\ldots))$$

Die dividierten Differenzen werden mittels des Dreiecksschemas rekursiv berechnet:

$$
\begin{array}{llllllll}
x_0 & y_0 = & [y_0] & & & & & \\
 & & & \ddots & & & & \\
x_1 & y_1 = & [y_1] & \dots & [y_0,y_1] & & & \\
 & & & \ddots & & \ddots & & \\
x_2 & y_2 = & [y_2] & \dots & [y_1,y_2] & \dots & [y_0y_1y_2] & \\
\vdots & & & & \vdots & & & \ddots \\
x_n & y_n = & [y_n] & \dots & [y_{n-1},y_n] & & \dots & [y_0,\dots,y_n] \quad .
\end{array}
$$

Algorithmus (MAPLE) siehe gesondertes Blatt.

Aufwand: für $n+1$ Knoten
Dividierte Differenzen:

$$
\begin{aligned}
&\tfrac{(n+1)n}{2} \ \cdot \ \text{(2 Additionen, 1 Multiplikation)} \\
&= (n+1)n \ \text{ Additionen, } \ (n+1)\tfrac{n}{2} \text{ Multiplikationen.}
\end{aligned}
$$

Auswertung des Polynoms (Hornerschema): $2n$ Additionen + n Multiplikationen

Vergleich: Newton-Darstellung zur numerischen Auswertung ist günstiger als der Neville-Algorithmus.

Bemerkung: Wähle die Stützstellen x_k äquidistant, d.h.

$$x_k = x_0 + kh, \quad k = 0,\dots,n, \quad \text{mit } \ h > 0.$$

Die Vorwärtsdifferenzen seien durch

$$\Delta^0 y_k := y_k, \qquad \Delta^r y_k := \Delta^{r-1} y_{k+1} - \Delta^{r-1} y_k$$

rekursiv definiert, d.h. z.B.

$$
\begin{aligned}
\Delta^1 y_k &= y_{k+1} - y_k, \\
\Delta^2 y_k &= \Delta^1 y_{k+1} - \Delta^1 y_k = (y_{k+2} - y_{k+1}) - (y_{k+1} - y_k) = y_{k+2} - 2y_{k+1} - y_k.
\end{aligned}
$$

Dann gilt: $[y_k,\dots,y_{k+r}] = \frac{1}{h^r r!}\ \Delta^r y_k$.

Beweis. Übungsaufgabe. □

Das Interpolationspolynom hat dann die Newtondarstellung

$$\begin{aligned} p_n(x) &= \sum_{j=0}^{n} \frac{1}{h^j} \frac{\Delta^j y_0}{j!} \cdot \prod_{k=0}^{j-1} (x - x_0 - kh) \\ &\underset{t=\frac{x-x_0}{h}}{=} \sum_{j=0}^{n} \frac{1}{h^j} \frac{\Delta^j y_0}{j!} \prod_{k=0}^{j-1} h(t-k) \\ &= \sum_{j=0}^{n} \Delta^j y_0 \cdot \underbrace{\frac{1}{j!} \cdot t(t-1)\dots(t-j+1)}_{:=\binom{t}{j}} = \sum_{j=0}^{n} \Delta^j y_0 \binom{t}{j}. \end{aligned}$$

Satz 1.12 (Leibnizregel für dividierte Differenzen)**.** *Ist $f = g \cdot h$ das Produkt zweier Funktionen, so gilt für die dividierte Differenz $f[x_0, \dots, x_n]$ zu den Knoten $x_0 < \dots < x_n$*

$$f[x_0, \dots, x_n] = \sum_{i=0}^{n} g[x_0 \dots x_i] \; h[x_i \dots x_n] \; .$$

Beweis. Setze

$$u_i(x) := \prod_{k=0}^{i-1} (x - x_k), \qquad \tilde{u}_j(x) := \prod_{l=j+1}^{n} (x - x_l).$$

Nach Satz 1.11 hat das Interpolationspolynom $p \in \Pi_n$ von g mit $p(x_k) = g(x_k)$, $k = 0, \dots, n$, die Darstellung

$$p(x) = \sum_{i=0}^{n} g[x_0, \dots, x_i] \cdot u_i(x).$$

Analog folgt für das Interpolationspolynom $q \in \Pi_n$ von h mit $q(x_k) = h(x_k)$, $k = 0, \dots, n$,

$$q(x) = \sum_{j=0}^{n} h[x_j, \dots, x_n] \; \tilde{u}_j(x).$$

Dann interpoliert

$$p(x)q(x) = \sum_{i=0}^{n} \sum_{j=0}^{n} g[x_0, \dots, x_i] \; h[x_j, \dots, x_n] \; u_i(x) \tilde{u}_j(x)$$

die Funktion f in $x_0, \ldots, x_n$. Aber für $i > j$ ist $u_i(x_k)\tilde{u}_j(x_k) = 0,\ k = 0, \ldots, n$. Das Polynom

$$w(x) = \sum_{i=0}^{n} \sum_{j=i}^{n} g[x_0, \ldots, x_i] h[x_j, \ldots, x_n] \underbrace{u_i(x)}_{\text{Grad} i} \underbrace{\tilde{u}_j(x)}_{\text{Grad}(n-j)}$$

interpoliert f ebenfalls in $x_0, \ldots, x_n$. Da $w \in \Pi_n$, hat $w(x)$ nach Definition 1.9 den Leitkoeffizient

$$f[x_0, \ldots, x_n] = \sum_{i=0}^{n} g[x_0, \ldots, x_i]\ h[x_i, \ldots, x_n].$$

□

Wir wollen nun für die dividierte Differenz eine Integraldarstellung herleiten:

Satz 1.13. *Ist $f \in C^n[x_0, x_n]$, so hat die dividierte Differenz $f[x_0, \ldots, x_n]$ zu den Knoten $x_0 < \ldots < x_n$ die Integraldarstellung*

$$f[x_0, \ldots, x_n] = \int_{x_0}^{x_n} f^{(n)}(t) G_{n-1}(t) dt$$

mit dem **Peano-Kern**

$$G_{n-1}(t) := \sum_{k=0}^{n} w_k \cdot \frac{(x_k - t)_+^{n-1}}{(n-1)!}, \quad t \in \mathbb{R},$$

wobei

$$w_k := \prod_{\substack{l=0 \\ l \neq k}}^{n} \frac{1}{x_k - x_l},$$

und $x_+^n := \begin{cases} x^n & , x \geq 0 \\ 0 & , x < 0 \end{cases}$ *die so genannte abgeschnittene Potenz (truncated power function) ist.*

Beweis. Für $f \in C^n[x_0, x_n]$ gilt die Taylorformel

$$\begin{aligned} f(x) &= \underbrace{\sum_{j=0}^{n-1} \frac{f^{(j)}(x_0)}{j!}(x - x_0)^j}_{:= p_{n-1}(x)} + \underbrace{\int_{x_0}^{x} \frac{(x-t)^{n-1}}{(n-1)!} f^{(n)}(t) dt}_{\text{Restglied}} \\ &= p_{n-1}(x) + \int_{x_0}^{x_n} \frac{(x-t)_+^{n-1}}{(n-1)!} f^n(t) dt. \end{aligned}$$

Nach Satz 1.10 (i) war

$$\begin{aligned} f[x_0,\ldots,x_n] &= \sum_{k=0}^{n} w_k f(x_k) \quad \text{mit } w_k = \prod_{\substack{l=0\\ l\neq k}}^{n} \frac{1}{x_k - x_l} \\ \Rightarrow f[x_0,\ldots,x_n] &= \sum_{k=0}^{n} w_k \left[p_{n-1}(x_k) + \int_{x_0}^{x_n} \frac{(x_k - t)_+^{n-1}}{(n-1)!} f^{(n)}(t)dt \right]. \end{aligned}$$

Wegen $p_{n-1} \in \Pi_{n-1}$ gilt mit Satz 1.10 (iv)

$$\sum_{k=0}^{n} w_k p_{n-1}(x_k) = p_{n-1}[x_0,\ldots,x_n] \ = 0.$$

Daraus folgt

$$\begin{aligned} f[x_0,\ldots,x_n] &= \sum_{k=0}^{n} w_k \int_{x_0}^{x_n} \frac{(x_k - t)_+^{n-1}}{(n-1)!} \ f^{(n)}(t)dt \\ &= \int_{x_0}^{x_n} f^{(n)}(t) \cdot \underbrace{\sum_{k=0}^{n} w_k \cdot \frac{(x_k - t)_+^{n-1}}{(n-1)!}}_{G_{n-1}(t)} \ dt. \end{aligned}$$

□

1.3 Interpolationsfehler

Sei f eine Funktion und p_n das zugehörige Interpolationspolynom, d.h.,

$$f(x_k) = y_k = p_n(x_k), \ k = 0,\ldots,n.$$

Frage: Wie groß ist der Interpolationsfehler

$$r_n(x) := f(x) - p_n(x) \ ?$$

Satz 1.14 (Cauchy-Form)**.** *Ist f in einem Intervall I, das $x, x_0, \ldots, x_n$ enthält, $(n+1)$-mal stetig differenzierbar, so gilt*

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \ w_{n+1}(x),$$

wobei $w_{n+1}(x) := (x - x_0) \cdot \ldots \cdot (x - x_n)$ und $\xi_x \in I$ eine Zwischenstelle ist.

Beweis. 1) Für $x = x_k$ ist $f(x) - p_n(x) = 0$ und die Behauptung gilt.
2) Sei $x \neq x_k \; \forall \, k = 0, \ldots, n$, fest. Wir betrachten die Funktion

$$g_x(t) = g(t) := f(t) - p_n(t) - \left(\frac{f(x) - p_n(x)}{w_{n+1}(x)} \right) \cdot w_{n+1}(t).$$

Dann hat die Funktion g im Intervall I mindestens die Nullstellen $x_0, \ldots, x_n, x$, und $g \in C^{n+1}\,(I)$. Wir wenden den Satz von Rolle $(n+1)$-mal an. Dann existiert ein $\xi_x \in I$, so dass $g^{(n+1)}\,(\xi_x) = 0$ gilt. Aus

$$g^{(n+1)}(t) = f^{(n+1)}\,(t) - \underbrace{p_n^{(n+1)}(t)}_{=0} - \left(\frac{f(x) - p_n(x)}{w_{n+1}(x)} \right) \cdot (n+1)!$$

folgt

$$g^{(n+1)}(\xi_x) = f^{(n+1)}\,(\xi_x) - \left(\frac{f(x) - p_n(x)}{w_{n+1}(x)} \right) (n+1)! = 0$$

$$\Leftrightarrow \quad \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \cdot w_{n+1}(x) = f(x) - p_n(x).$$

□

Da die Lage von ξ_x nicht bekannt ist, schätzen wir $f^{(n+1)}(\xi_x)$ durch die Maximumnorm $\|f^{(n+1)}\|_\infty$ ab.

Folgerung 1.15. *Unter den Voraussetzungen von Satz 1.14 gilt die Abschätzung*

$$|f(x) - p_n(x)| \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!} \; |w_{n+1}(x)| \quad \forall \, x \in I,$$

wobei

$$\|f^{(n+1)}\|_\infty := \max_{t \in I} \; |f^{(n+1)}(t)| \; .$$

Eine andere Fehlerdarstellung erhält man mit Hilfe dividierter Differenzen. Die Voraussetzung der Differenzierbarkeit ist hier nicht notwendig!

Satz 1.16. *Bei der Interpolation von $f \in C^1(I)$ durch ein Polynom $p_n \in \Pi_n$ in den paarweise verschiedenen Stützstellen $x_k, k = 0, \ldots, n$ gilt*

$$f(x) - p_n(x) = f[x_0, \ldots, x_n, x] \; w_{n+1}(x).$$

Beweis. 1) Sei zunächst $x \neq x_k, k = 0, \ldots, n$, fest. Es war

$$p_n(t) = f[x_0] + f[x_0, x_1](t - x_0) + \ldots + f[x_0, x_1, \ldots, x_n](t - x_0) \ldots (t - x_{n-1}).$$

Betrachte nun das Interpolationspolynom für die Stützstellen $x_0, \ldots, x_n, x$ und die Stützwerte $f(x_0), \ldots, f(x_n), f(x)$,

$$\begin{aligned} p_{n+1}(t) &= f[x_0] + f[x_0, x_1](t - x_0) + \ldots + f[x_0, \ldots, x_n]\ (t - x_0) \ldots (t - x_{n-1}) \\ &\quad + f[x_0, \ldots, x_n, x]\ (t - x_0) \ldots (t - x_{n-1})(t - x_n). \end{aligned}$$

Dann ergibt sich

$$p_{n+1}(t) - p_n(t) = f[x_0, \ldots, x_n, x](t - x_0) \ldots (t - x_n).$$

Setze nun $t = x$. Dann folgt $p_{n+1}(x) = f(x)$, und damit erhalten wir

$$f(x) - p_n(x) = f[x_0, \ldots, x_n, x]\ \cdot \underbrace{(x - x_0) \ldots (x - x_n)}_{w_{n+1}(x)}.$$

2) Für $x = x_k,\ k = 0, \ldots, n$, lässt sich $f[x_0, \ldots, x_n, x]$ folgendermaßen definieren:

$$f[x_k, x_k] = \lim_{x \to x_k} \frac{f(x) - f(x_k)}{x - x_k} = f'(x_k),$$

sonst, wie bisher

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

Die Behauptung folgt nun mit $w_{n+1}(x_k) = 0, \quad k = 0, \ldots, n.$ □

Durch Vergleich mit der Cauchy-Darstellung des Interpolationsfehlers finden wir

Folgerung 1.17. *Ist $f \in C^n[x_0, x_n]$, so hat die dividierte Differenz zu den Knoten $x_0 < \ldots < x_n$ die Darstellung*

$$f[x_0, \ldots, x_n] = \frac{1}{n!}\ f^{(n)}(\xi)$$

für ein $\xi \in [x_0, x_n]$.

Optimale Stützstellenwahl

Problem: Wie können wir durch günstige Wahl der Stützstellen den Interpolationsfehler $f(x) - p(x)$ minimieren?
Erinnerung: Es war $|f(x) - p(x)| \leq \frac{\|f^{n+1}\|_\infty}{(n+1)!}\ |w_{n+1}(x)|$.

Idee: Wähle $x_0, \ldots x_n \in [a, b]$ so, dass $\max_{x \in [a,b]}\ |w_{n+1}(x)|$ möglichst klein wird.

Beispiel:

Sei $[a,b]=[-1,1]$ mit äquidistanter Knotenfolge $x_0=-1, x_1=-1+\frac{2}{n}, \ldots, x_n=1$, d.h. $x_k=-1+\frac{2k}{n}$ für $k=1,\ldots,n$. Sei $x\in[x_j,x_{j+1}]$.
1) Für $k<j$: $\underbrace{x_j-x_k}_{2(j-k)/n}\le x-x_k\le\underbrace{x_{j+1}-x_k}_{2(j+1-k)/n}$.
2) Für $k\ge j+1$: $\underbrace{x_k-x_j+1}_{2(k-j-1)/n}\ \le|x_k-x|\ \le\underbrace{x_k-x_j}_{2(k-j)/n}$.
Daraus folgt einerseits

$$\begin{aligned}|w_{n+1}(x)| &\le \prod_{k=0}^{j-1}2\frac{(j+1-k)}{n}\ \cdot\prod_{k=j+2}^{n}2\frac{(k-j)}{n}\cdot|x_j-x||x_{j+1}-x|\\ &= (\frac{2}{n})^{n-1}\ \cdot(j+1)!\cdot(n-j)!|x-x_j|\ |\ x_{j+1}-x|\end{aligned}$$

und andererseits

$$|w_{n+1}(x)|\ge(\frac{2}{n})^{n-1}\ j!(n-j-1)!|x-x_j||x_{j+1}-x|.$$

$|w_{n+1}\ (x)|$ kann für x in der Nähe des Intervallrandes recht groß werden, z.B.

$$x=\frac{x_{n+1}+x_n}{2}\ \Rightarrow|\ w_{n+1}\ (x)|\ge(\frac{2}{n})^{n-1}\ \cdot(n-1)!\cdot\frac{1}{n}\ \cdot\frac{1}{n}$$

$$\begin{aligned}n=5: \quad |w_6(0,825)| &= 0.113462959\approx\max_{x\in[-1,1]}\ |w_6(x)|,\\ n=6: \quad |w_7(0.865)| &= 0.0692257\approx\max_{x\in[-1,1]}\ |w_7(x)|.\end{aligned}$$

Ist eine andere Knotenwahl günstiger? Ja!
Wir wählen die Stützstellen $x_k:=\cos\frac{(2k+1)\pi}{2(n+1)},\ \ k=0,\ldots,n$.
Dann ist

$$T_{n+1}(x):=2^n\prod_{k=0}^{n}(x-\cos\frac{(2k+1)\pi}{2(n+1)}\ )$$

das **Tschebyscheff-Polynom** vom Grad $n+1$.

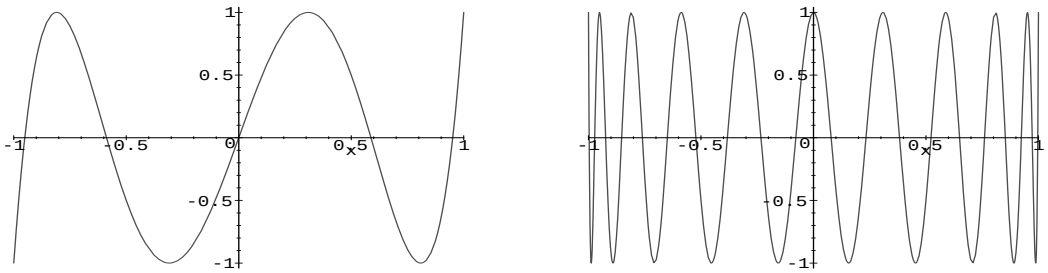
Satz 1.18. *Das Tschebyscheff-Polynom lässt sich für $x\in[-1,1]$ in der Form*

$$T_n(x)=\cos n(\arccos x),\quad n=0,1,2,\ldots$$

darstellen, d.h. $T_n(\cos\varphi)=\cos(n\varphi)$. $(x:=\cos\varphi\in[-1,1])$.
Die Tschebyscheff-Polynome genügen der Rekursionsformel

$$T_{n+1}(x)=2xT_n(x)-T_{n-1}(x),\quad n\ge1$$

mit $T_0(x)=1, T_1(x)=x$.



Tschebyscheff-Polynome vom Grad 5 (links) und vom Grad 20 (rechts)

Beweis. Sei $\tilde{T}_n(x) := \cos n(\arccos x)$ für $x \in [-1,1]$. Da sich $\cos(n\varphi)$ als Linearkombination von $\cos^j \varphi,\ j = 0,\ldots,n$, darstellen lässt, ist $\tilde{T}_n(x) \in \Pi_n$, denn: Es gilt

$$\cos(n\varphi) = \cos((n-1)\varphi + \varphi) = \cos(n-1)\varphi \cos\varphi - \sin(n-1)\varphi \sin\varphi.$$

Außerdem gilt $\sin\alpha \sin\beta = \frac{1}{2}[\cos(\alpha-\beta) - \cos(\alpha+\beta)]$. Damit folgt

$$\sin(n-1)\varphi \sin\varphi = \frac{1}{2}\cos(n-2)\varphi - \frac{1}{2}\cos n\varphi$$

Einsetzen ergibt

$$\frac{1}{2}\cos n\varphi = \cos\varphi \cos(n-1)\varphi - \frac{1}{2}\cos(n-2)\varphi.$$

Also ist $\tilde{T}_n \in \Pi_n$, und die Rekursionsformel gilt. Weiter erhalten wir

$$\begin{aligned} \tilde{T}_n(\cos\tfrac{(2k+1)\pi}{2n}) &= \cos n(\arccos(\cos\tfrac{(2k+1)\pi}{2n})) = \cos(n\tfrac{(2k+1)\pi}{2n}) \\ &= \cos\tfrac{(2k+1)\pi}{2} = 0. \end{aligned}$$

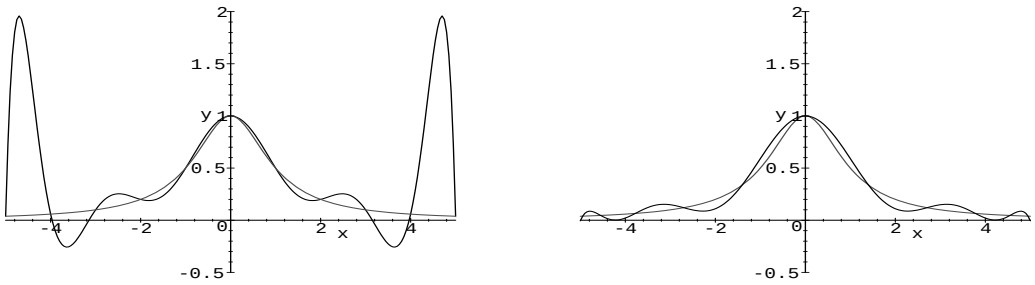
Das bedeutet $\tilde{T}_n = T_n$. (Höchstkoeffizient 2^{n-1} folgt aus Rekursionsformel.) □

Beachte: Bei Knotenwahl $x_k = \cos\frac{(2k+1)\pi}{2(n+1)}, \quad k = 0,\ldots,n$, ist $w_{n+1}(x) = \frac{T_{n+1}(x)}{2^n}$.

Satz 1.19. *Das Polynom $2^{-n}T_{n+1}(x)$ ist vom Grad $n+1$ mit höchstem Koeffizienten Eins, und es gilt*

$$2^{-n} = \max_{x\in[-1,1]} \frac{|T_{n+1}(x)|}{2^n} \ \leq \max_{x\in[-1,1]} |w_{n+1}(x)|$$

für jedes Polynom $w_{n+1}(x)$ vom Grad $n+1$ mit höchstem Koeffizient 1.



Interpolation für äquidistante Knoten (links) und Tschebyscheff-Knoten (rechts) für $n = 10$.

Beweis. Das Polynom $T_{n+1}(x)$ hat den Höchstkoeffizient 2^n. Dies folgt induktiv aus der Rekursionsformel in Satz 1.18.
Aus der Definition von $T_{n+1}(x)$ folgt $|T_{n+1}(x)| \leq 1$ für $x \in [-1, 1]$. Die Extremalwerte werden an den $n+2$ Punkten $x_k = \cos \frac{k\pi}{n+1}$, $k = 0, \ldots, n+1$, angenommen:

$$T_{n+1}(\cos \frac{k\pi}{n+1}) = \cos\left((n+1)\frac{k\pi}{(n+1)}\right) = \cos(k\pi) = \begin{cases} 1 & , k \text{ gerade} \\ -1 & , k \text{ ungerade.} \end{cases}$$

Sei $w_{n+1} \in \Pi_{n+1}$ beliebig mit Höchstkoeffizient 1. Angenommen $|w_{n+1}(x)| < 2^{-n}$ $\forall\ x \in [-1, 1]$. Dann ist $p(x) = w_{n+1}(x) - \frac{1}{2^n} T_{n+1}(x)$ alternierend positiv bzw. negativ an den Stellen $\frac{k\pi}{n+1}$, $k = 0, \ldots, n+1$, hat also mindestens $n+1$ Nullstellen. Da aber $p \in \Pi_n$ folgt damit $p \equiv 0$. Dies ist ein Widerspruch zur Annahme. □

Beispiel: Für die Knotenwahl $x_k = \cos \frac{(2k+1)\pi}{2(n+1)}$, $k = 0, \ldots, n$ erhalten wir nun

$$\begin{array}{lll} n = 5: & \max_{x\in[-1,1]} |w_6(x)| & = 2^{-5} = 0.3125, \\ n = 6: & \max_{x\in[-1,1]} |w_7(x)| & = 2^{-6} = 0.015625 \end{array}$$

Beispiel: (Runge)
Es sei $f(x) = \frac{1}{x^2+25}$. Wir betrachten Interpolation auf dem Intervall $[-5, 5]$.

1. Äquidistante Knoten: $x_k = -5 + \frac{10k}{n}$, $k = 0, \ldots, n$, $\Rightarrow$ Divergenz für $n \to \infty$.

2. Tschebyscheff-Knoten: $x_k = 5 \cos \frac{(2k+1)\pi}{2(n+1)})$, $k = 0, \ldots, n$, $\Rightarrow$ Konvergenz für $n \to \infty$.

Einfluss von Störungen der Funktionswerte

Die Stützwerte $y_k, k = 0, \ldots, n$, seien fehlerbehaftet. Es gelte:

$$\tilde{y}_k = y_k + \varepsilon_k, \quad |\varepsilon_k| \ < \varepsilon \text{ für } k = 0, \ldots, n.$$

Das mit den korrekten Stützwerten y_k berechnete Interpolationspolynom p sowie das mit $\tilde{y}_k$ berechnte Polynom $\tilde{p}$ haben folgende Lagrangedarstellung

$$p = \sum_{k=0}^{n} y_k l_k, \qquad \tilde{p} = \sum_{k=0}^{n} \tilde{y}_k l_k.$$

Sei

$$\eta = \tilde{p} - p = \sum_{k=0}^{n} (\tilde{y}_k - k_k)\, l_k.$$

Dann folgt

$$|\eta(x)| \leq \sum_{k=0}^{n} |\varepsilon| \mid l_k(x)| \ \leq \varepsilon \cdot L_n(x),$$

wobei $L_n(x) := \sum_{k=0}^{n} |l_k(x)|$. Wir erhalten für $a = x_0 < x_1 \ldots < x_n = b$

$$\max_{x \in [a,b]} |\tilde{p}(x) - p(x)| \ = \max_{x \in [a,b]} |\eta(x)| \ \leq \varepsilon \cdot \underbrace{\max_{x \in [a,b]} L_n(x)}_{=:\Lambda_n}.$$

Die Konstante $\Lambda_n := \max_{x \in [a,b]} L_n(x)$ ist nur abhängig von den Stützstellen $x_0 < x_1 \ldots < x_n$. Sie ist ein Maß dafür, wieweit sich der Eingangsfehler verstärkend auf das Endresultat auswirkt.

Beispiel: $[a, b] \ = [1, 1]$

n	Λ_n für äquidistante Knoten	Λ_n für Tscheyscheff-Knoten
5	$3,106292$	$2,104398$
10	$29,890695$	$2,489430$
15	$512,052451$	$2,727778$
20	$10986,533993$	$2,900825$

(Λ_n divergiert für beide Knotensequenzen gegen ∞, jedoch unterschiedlich schnell!)

1.4 Rationale Interpolation

Definition 1.20. *Die Menge der rationalen Funktionen ist die Menge*

$$\mathcal{R}(l,m) := \{r = \frac{p}{q} : p \in \Pi_l, q \in \Pi_m, q(x) \not\equiv 0\}.$$

Die Funktionen aus $\mathcal{R}(l,m)$ hängen von $l+m+1$ Parameter ab, da man die $(l+1)+(m+1) = l+m+2$ Polynomkoeffizienten noch einer Normierung unterwerfen kann. Wir betrachten das **Interpolationsproblem:**

Zu $n+1$ paarweise verschiedenen Stützstellen $x_0, \ldots, x_n (n := l+m)$ und $(n+1)$ zugehörigen Stützwerten $y_0, \ldots, y_n$ finde man eine Funktion $r \in \mathcal{R}(l,m)$ mit

$$r(x_k) = \frac{p(x_k)}{q(x_k)} = y_k, \qquad k = 0, \ldots, n. \tag{1.3}$$

Falls $r(x)$ die Bedingungen $(*)$ erfüllt, folgt

$$p(x_k) = y_k \cdot q(x_k) \qquad k = 0, \ldots, n. \tag{1.4}$$

Beachte: (1.3) und (1.4) sind nicht äquivalant!

Satz 1.21.

(i) Das Problem (1.4) *hat immer eine nichttriviale Lösung.*

(ii) Sind p_1, q_1 und p_2, q_2 jeweils Lösungen von (1.4)*, so gilt*

$$p_1 \cdot q_2 = p_2 \cdot q_1.$$

Beweis. (i) Sei $\mathbf{a} = (a_0, a_1, \ldots, a_l)^T, \mathbf{b} = (b_0, b_1, \ldots, b_m)^T$ und

$$p(x) = \sum_{j=0}^{l} a_j x^j = (1, x, \ldots, x^l) \cdot \mathbf{a}\ ,\ q(x) = (1, \ldots, x^m) \cdot \mathbf{b}.$$

Dann hat (1.4) die Form

$$\begin{aligned} \sum_{j=0}^{l} a_j x_k^j - y_k \sum_{r=0}^{m} b_r x_k^r &= 0, \quad k = 0, \ldots, n \\ \Leftrightarrow (1, x_k, \ldots, x_k^l) \cdot \mathbf{a} - y_k (1, x_k, \ldots, x_k^m) \cdot \mathbf{b} &= 0, \quad k = 0, \ldots, n. \end{aligned}$$

Dies ist ein homogenes LGS. In Matrixschreibweise folgt

$$\begin{pmatrix} 1 & x_0 & \dots & x_0^l \\ 1 & x_1 & \dots & x_1^l \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \dots & x_n^l \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_l \end{pmatrix} - \begin{pmatrix} y_0 & & & \\ & y_1 & & \\ & & \ddots & \\ & & & y_n \end{pmatrix} \begin{pmatrix} 1 & x_0 & \dots & x_0^m \\ 1 & x_1 & \dots & x_1^m \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \dots & x_n^m \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_m \end{pmatrix} = \mathbf{0}$$

bzw.

$$\left(\begin{array}{cccc|ccc} 1 & x_0 & \dots & x_0^l & y_0 & y_0x_0 & \dots & y_0x_0^m \\ 1 & x_1 & \dots & x_1^l & y_1 & y_1x_1 & \dots & y_1x_1^m \\ \vdots & \vdots & & & \vdots & & & \vdots \\ 1 & x_n & \dots & x_n^l & y_n & y_nx_n & \dots & y_nx_n^m \end{array} \right) \begin{pmatrix} \mathbf{a} \\ \\ -\mathbf{b} \end{pmatrix} = \mathbf{0}$$

Dieses LGS hat $n+1$ Gleichungen und $n+2$ Unbekannte mit einer Koeffizientenmatrix $\mathbf{A} \in \mathbb{R}^{(n+1)\times(n]2)}$. Es besitzt mindestens eine nichttriviale Lösung, da $rg(\mathbf{A}) \leq n+1 = rg(\mathbf{A}, \mathbf{0})$. Wegen

$$rg \begin{pmatrix} 1 & x_0 & \dots & x_0^l \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \dots & x_n^l \end{pmatrix} = l+1$$

folgt für alle nichttrivialen Lösungen $\mathbf{b} \neq \mathbf{0}$. Also ist $q(x) \not\equiv 0$, und das Problem (1.4) hat mindestens eine nichttriviale Lösung.

(ii) Sei

$$\begin{aligned} p_1(x_k) &= y_k q_1(x_k), \quad k = 0, \dots, n, \\ p_2(x_k) &= y_k q_2(x_k), \quad k = 0, \dots, n, \end{aligned}$$

mit $p_1, p_2 \in \Pi_l, q_1, q_2 \in \Pi_m$. Dann besitzt $p_1q_2 - p_2q_1$ den Grad $\leq l+m = n$, und es gilt

$$\begin{aligned} p_1(x_k)q_2(x_k) - p_2(x_k)q_1(x_k) &= y_k q_1(x_k)q_2(x_k) - y_k q_2(x_k)q_1(x_k) \\ &= 0, \quad k = 0, \dots, n. \end{aligned}$$

Nach Satz 1.2 folgt $p_1q_2 - p_2q_1 \equiv 0$. □

Frage: Wann folgt aus (1.4) auch (1.3)?

Für jedes $k = 0, \dots, n$, unterscheiden wir 2 Fälle:

1. Fall: $q(x_k) \neq 0 \quad \Rightarrow \frac{p(x_k)}{q(x_k)} = y_k \Longleftrightarrow p(x_k) = y_k q(x_k)$.

2. Fall: $q(x_k) = 0 \quad \Rightarrow p(x_k) = y_k \cdot \underbrace{q(x_k)}_{=0} = 0$.

Dann enthalten die Polynome p und q den Faktor $(x-x_k)^{d_p}$ bzw. $(x-x_k)^{d_q}$ mit $d_p, d_q \in \mathbb{N}, d_p, d_q \geq 1$. Sei $d = \min\{d_p, d_q\}$. Setze

$$p^*(x) := \frac{p(x)}{(x-x_k)^d}, \qquad q^*(x) := \frac{q(x)}{(x-x_k)^d}$$

und q^* oder p^* hat keine Nullstelle $x = x_k$. Das Polynompaar p^*, q^* erfüllt dann für alle $j = 0, \ldots, n, j \neq k$ die Bedingung (1.4), denn

$$p(x_j) = y_j q(x_j) \Longleftrightarrow \frac{p(x_j)}{(x_j-x_k)^d} = y_j \frac{q(x_j)}{(x_j-x_k)^d} \qquad (j \neq k).$$

Also: Für $q(x_k) = 0$ ist (1.4) für **jede** Wahl von y_k erfüllt, aber $\frac{p(x_k)}{q(x_k)} = \frac{p^*(x_k)}{q^*(x_k)}$ ist ein fester Wert, der im Allgemeinen nicht mit y_k übereinstimmt. In diesem Fall heißt (x_k, y_k) **unerreichbarer** Punkt.

Beispiel: Sei $r(x) = \frac{a_1 x + a_0}{b_1 x + b_0}$ d.h. also ($l = 1, m = 1$ und damit $n = 2$.) Wähle nun $(x_0, y_0) = (-1, \frac{1}{3}), (x_1, y_1) = (0, 1)$ und $(x_2, y_2) = (1, \frac{1}{3})$. Betrachte

$$(a_1 x_j + a_0) - y_j(b_1 x_j + b_0) = 0, \qquad j = 0, 1, 2.$$

Das führt zu folgendem LGS

$$\begin{aligned} a_1 \cdot (-1) + a_0 - \tfrac{1}{3}\,(b_1 \cdot (-1) + b_0) &= 0 \\ a_1 \cdot 0 + a_0 - 1(b_1 \cdot 0 + b_0) &= 0 \\ a_1 \cdot 1 + a_0 - \tfrac{1}{3}\,(b_1 \cdot 1 + b_0) &= 0 \end{aligned}$$

$$\Longleftrightarrow \begin{pmatrix} 1 & -1 & -\frac{1}{3} & \frac{1}{3} \\ 1 & 0 & -1 & 0 \\ 1 & 1 & -\frac{1}{3} & -\frac{1}{3} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ b_0 \\ b_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\left(\begin{array}{cccc|c} 1 & -1 & -\frac{1}{3} & \frac{1}{3} & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 1 & 1 & -\frac{1}{3} & -\frac{1}{3} & 0 \end{array}\right) \Longleftrightarrow \left(\begin{array}{cccc|c} 1 & -1 & -\frac{1}{3} & \frac{1}{3} & 0 \\ 0 & 1 & -\frac{2}{3} & -\frac{1}{3} & 0 \\ 0 & 2 & 0 & -\frac{2}{3} & 0 \end{array}\right)$$

$$\Longleftrightarrow \left(\begin{array}{cccc|c} 1 & -1 & -\frac{1}{3} & \frac{1}{3} & 0 \\ 0 & 1 & -\frac{2}{3} & -\frac{1}{3} & 0 \\ 0 & 0 & -\frac{4}{3} & 0 & 0 \end{array}\right).$$

Das bedeutet $b_0 = 0$ und mit dem Parameter $b_1 = 3c$ dann $a_1 = c$ ($c \in \mathbb{R}$) und $a_0 = c - c = 0$.
Wir erhalten

$$r(x) = \frac{cx}{3cx} = \frac{1}{3} \quad \text{für} \quad x \neq 0.$$

Also ist $(x_1, y_1) = (0, 1)$ ein unerreichbarer Punkt.

Aus den Vorüberlegungen ergibt sich nun

Satz 1.22. *Erfüllen p, q die Bedingung* (1.4) *und sind sie zusätzlich teilerfremd, so löst $r = \frac{p}{q}$ das Problem* (1.3).

Bemerkung: Es stellt sich die Frage, ob man den Daten (x_j, y_j) a priori ansehen kann, dass die gefundene Interpolierende zwischen den Interpolationspunkten stetig ist, d.h. der Nenner $\neq 0$ ist. Dieses Problem ist für $m \geq 2$ ungelöst.

Der Kettenbruchalgorithmus
Eine Methode zur Darstellung und Berechnung der rationalen Interpolanten ist die Kettenbruchdarstellung.

Ansatz: $r(x) = a_0 + \cfrac{x - x_0}{a_1 + \cfrac{x - x_1}{a_2 + \ddots + \cfrac{x - x_{n-1}}{a_n}}}$

Kurzschreibweise:

$$r(x) = a_0 + \frac{x - x_0|}{|a_1} + \frac{x - x_1|}{|a_2} + \ldots + \frac{x - x_{n-1}|}{|a_n}$$

$r(x)$ ist nun eine rationale Funktion.

Beispiel: ($n = 3$)

$$r(x) = a_0 + \frac{(x - x_0)}{a_1 + \left(\frac{(x-x_1)}{a_2 + \left(\frac{x - x_2}{a_3}\right)}\right)} = a_0 + \frac{(x - x_0)}{a_1 + \frac{(x-x_1)a_3}{a_2 a_3 + (x - x_2)}}$$

$$= a_0 + \frac{(x - x_0)(a_2 a_3 + (x - x_2))}{a_1 a_2 a_3 + a_1(x - x_2) + (x - x_1)a_3}$$

$$= \frac{a_0 a_1 a_2 a_3 + a_0 a_1 (x - x_2) + a_0 a_3 (x - x_1) + a_2 a_3 (x - x_0) + (x - x_0)(x - x_2)}{a_1 a_2 a_3 + a_1(x - x_2) + (x - x_2)a_3}$$

Also ist $r \in \mathcal{R}(2,1)$. Wir betrachten nun das Interpolationsproblem $r(x_k) = y_k,\ k = 0,1,2,3$ und erhalten

$$\begin{aligned}
r(x_0) &= a_0 = y_0 \\
r(x_1) &= a_0 + \frac{(x_1 - x_0)}{a_1 + 0} = y_1 \Longrightarrow a_1 = \frac{x_1 - x_0}{y_1 - y_0} \\
r(x_2) &= a_0 + \frac{(x_2 - x_0)}{a_1 + \frac{(x_2 - x_1)}{a_2}} = y_2 \Longrightarrow \frac{x_2 - x_0}{a_1 + \frac{(x_2 - x_1)}{a_2}} = y_2 - a_0 \\
\Longrightarrow & \quad \frac{1}{a_1 + \frac{(x_2 - x_1)}{a_2}} = \frac{y_2 - a_0}{x_2 - x_0} = \frac{y_2 - y_0}{x_2 - x_0} \\
\Longrightarrow & \quad a_1 + \frac{(x_2 - x_1)}{a_2} = \frac{x_2 - x_0}{y_2 - y_0} \Longrightarrow a_2 = \frac{x_2 - x_1}{\left(\frac{x_2 - x_0}{y_2 - y_0}\right) - \left(\frac{x_1 - x_0}{y_1 - y_0}\right)}
\end{aligned}$$

usw. für $k = 3$.

Definition 1.23. *Gegeben seien $(x_0, \ldots, x_n)$ und $(y_0, \ldots, y_n)$. Dann sind:*

$$\begin{aligned}
\nabla^0(y_0) &:= y_0, \quad \nabla^1(y_0, y_1) := \frac{x_1 - x_0}{y_1 - y_0} \\
\nabla^n(y_0, \ldots, y_n) &:= \frac{x_n - x_{n-1}}{\nabla^{n-1}(y_0, \ldots, y_{n-2}, y_n) - \nabla^{n-1}(y_0, \ldots, y_{n-1})}
\end{aligned}$$

die **inversen Differenzenquotienten** *zu den Vektoren $(x_j, y_j),\ j = 0, \ldots, n$. Dabei setzen wir voraus, dass die Nenner $\neq 0$ sind.*

Beachte: Beim inversen Differenzenquotienten kommt es auf die Reihenfolge der Argumente an!

Wir erhalten die Kettenbruchdarstellung:

$$\begin{aligned}
\nabla^n(y_0, \ldots, y_n) &= \frac{x_n - x_{n-1}}{-\nabla^{n-1}(y_0, \ldots, y_{n-1}) + \left(\frac{x_n - x_{n-2}}{\nabla^{n-2}(y_0, \ldots, y_{n-3}, y_n) - \nabla^{n-2}(y_0, \ldots, y_{n-2})}\right)} \\
&= \frac{x_n - x_{n-1}}{-\nabla^{n-1}(y_0, \ldots, y_{n-1}) + \frac{x_n - x_{n-2}}{-\nabla^{n-2}(y_0, \ldots, y_{n-2}) + \frac{x_n - x_{n-3}}{-\Delta^{n-3}\ (y_0, \ldots, y_{n-3}) + \nabla^{n-3}(y_0, \ldots, y_{n-4}, y_n)}}} \\
&= \ldots \\
&= \frac{x_n - x_{n-1}|}{|-\nabla^{n-1}(y_0, \ldots, y_{n-1})} + \frac{x_n - x_{n-2}|}{|-\nabla^{n-2}(y_0, \ldots, y_{n-2})} + \ldots + \frac{x_n - x_0|}{|y_n - y_0}. \qquad (1.5)
\end{aligned}$$

Satz 1.24. *Existieren die Differenzenquotienten $a_j := \nabla^j(y_0, \ldots, y_j)$ für $j = 0, \ldots, n$, so stellt der mit diesem Koeffizienten gebildeten Kettenbruch $r(x) = a_0 + \frac{x - x_0|}{|a_1} + \frac{x - x_1|}{|a_2} + \ldots + \frac{x - x_{n-1}|}{|a_n}$ eine rationale Interpolierende mit $r(x_k) = y_k,\ k = 0, \ldots, n$, dar.*

Beweis. Durch vollständige Induktion:
1) Induktionsanfang ($n = 1$): Sei $r_1(x) = a_0 + \frac{x-x_0}{a_1}$. Dann ist $r_1(x_0) = y_0 = a_0$ und $r_1(x_1) = y_1 = a_0 + \frac{(x_1-x_0)}{a_1}$. Umformung ergibt

$$y_1 - a_0 = \frac{(x_1 - x_0)}{a_1} \quad \Leftrightarrow \quad a_1 = \frac{x_1 - x_0}{y_1 - y_0} \; = \nabla^1(y_0, y_1)$$

falls $y_1 \neq y_0$.
2) Induktionsvoraussetzung: Sei nun

$$r_{n-1}(x) = a_0 + \frac{(x - x_0)|}{|a_1} + \ldots + \frac{x - x_{n-2}|}{|a_{n-1}}$$

mit $a_j = \nabla^j(y_0, \ldots, y_j),\ j = 0, \ldots, n-1,$ und $r_{n-1}(x_k) = y_k,\ k = 0, \ldots, n-1$.
3) Induktionsschritt: Wir betrachten nun

$$r_n(x) = a_0 + \frac{x - x_0|}{|a_1} \; + \ldots + \frac{x - x_{n-2}|}{|a_{n-1}} + \frac{x - x_{n-1}|}{|a_n}$$

und $a_j = \nabla^j(y_0, \ldots, y_j)$ für $j = 0, \ldots, n-1$.
Dann ist

$$r_n(x_k) = a_0 + \frac{x_k - x_0|}{|a_1} \; + \ldots + \frac{x_k - x_{k-1}|}{|a_k} \; + 0 = r_{n-1}(x_k) = y_k$$

für $k = 0, \ldots, n-1$ nach Induktionsvoraussetzung. Wir zeigen, dass

$$r_n(x_n) = y_n \Longleftrightarrow a_n = \nabla^n(y_0, \ldots, y_n).$$

Sei

$$y_n = r_n(x_n) = a_0 + \frac{x_n - x_0|}{|a_1} + \ldots + \frac{x - x_{n-2}|}{|a_{n-1}} + \frac{x_n - x_{n-1}|}{|a_n}.$$

Durch Umstellen folgt

$$\begin{aligned}
& (y_n - a_0)(a_1 + \frac{x_n - x_1|}{|a_2} \; + \ldots + \frac{x_n - x_{n-1}\,|}{|a_n} \;) = (x_n - x_0) \\
\Rightarrow \quad & (a_1 + \frac{x_n - x_1|}{|a_2} \; + \ldots + \frac{x_n - x_{n-1}|}{|a_n}) = \frac{x_n - x_0}{y_n - y_0} = \nabla^1(y_0, y_n) \\
\Rightarrow \quad & \big(\nabla^1(y_0, y_n) - \nabla^1(y_0, y_1)\;\big)\,(a_2 + \frac{x_n - x_2|}{|a_3} \; + \ldots + \frac{x_n - x_{n-1}|}{|a_n}) = x_n - x_1 \\
\Rightarrow \quad & (a_2 + \frac{x_n - x_1|}{|a_3} + \ldots + \frac{x_n - x_{n-1}|}{|a_n}) = \nabla^1(y_0, y_1) - \nabla^1(y_0, y_1)) \\
& = \nabla^2(y_0, y_1, y_n)
\end{aligned}$$

Rekursive Rechnung ergibt

$$\begin{aligned} \Rightarrow a_n &= \frac{x_n - x_{n-1}|}{|-\nabla^{n-1}(y_0,\ldots,y_{n-1})} + \frac{x_n - x_{n-2}|}{-|\nabla^{n-2}\,(y_0,\ldots,y_{n-2}\,)} + \ldots + \frac{x_n - x_0|}{|y_n - y_0} \\ &= \nabla^n(y_0,\ldots,y_n) \end{aligned}$$

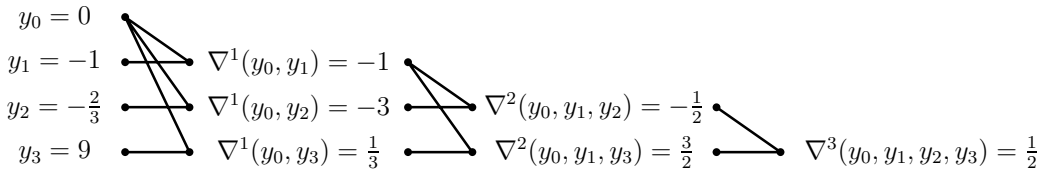
nach (1.5). □

Bemerkung: Treten bei der Division Nullen auf, so kann man durch Vertauschen der Interpolationspunkte versuchen, diese Schwierigkeiten zu umgehen.

Beispiel zur Berechnung der inversen Differenzenquotienten:
Gegeben:

x_i	0	1	2	3
y_i	0	-1	$-\frac{2}{3}$	9

Dreiecksschema:

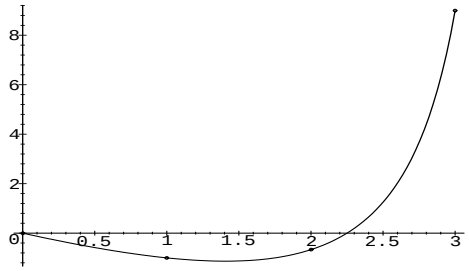


$$\begin{aligned} \nabla^1(y_0,y_1) &= \frac{x_1 - x_0}{y_1 - y_0} = \frac{1}{-1} = -1, \quad \nabla^1(y_0,y_2) = \frac{x_2 - x_0}{y_2 - y_0} = \frac{2}{-\frac{2}{3}} = -3, \\ \nabla^2(y_0,y_1,y_2) &= \frac{x_2 - x_1}{\nabla^1(y_0,y_2) - \nabla^1(y_0,y_1)} = \frac{2-1}{-3-(-1)} = -\frac{1}{2} \quad \text{usw.} \end{aligned}$$

Wir erhalten:

$$r_3(x) = 0 + \frac{x|}{|-1} + \frac{x-1|}{|-\frac{1}{2}} + \frac{x-2|}{|\frac{1}{2}} = \ldots = \frac{4x^2 - 9x}{-2x + 7}\,.$$

Die Koeffizienten in der Kettenbruchdarstellung können also iterativ mit Hilfe der Rekursionsformel in einem Dreiecksschema berechnet werden.
Zur Berechnung von Funktionswerten von $r_n(x)$ lässt sich ein dem Horner-Schema ähnliches Schema verwenden.



rationale Interpolation für $(l, m) = (2, 1)$ (siehe Beispiel).

Beispiel: $(n = 3)$

$$r_3(x) = a_0 + \frac{x - x_0}{(a_1 + \frac{x-x_1}{(a_2 + \frac{x-x_2}{a_3})})} \qquad \Rightarrow \qquad \begin{aligned} z_0 &= 0 \\ z_1 &= \tfrac{x-x_2}{a_3+z_0} \\ z_2 &= \tfrac{x-x_1}{a_2+z_1} \\ z_3 &= \tfrac{x-x_0}{a_1+z_2} \\ z_4 &= a_0 + z_3 \end{aligned}$$

1.5 Trigonometrische Interpolation

Will man eine periodische Funktion interpolieren, so ist es sinnvoll, trigonometrische Polynome zu verwenden anstatt algebraischer Polynome.

Definition 1.25. *Für $c_k \in \mathbb{C}, k = -n, \ldots, n,\ c_n \neq 0$, bezeichnet man*

$$p_n(t) = \sum_{k=-n}^{n} c_k e^{ikt} \qquad (t \in \mathbb{R})$$

als (komplexes) trigonometrisches Polynom vom Grad n. Ein Polynom der Form

$$\tilde{p}_n(t) = \frac{a_0}{2} + \sum_{k=1}^{n} a_k \cos(kt) + b_k \sin(kt) \quad (a_k, b_k \in \mathbb{R})$$

mit $a_n \neq 0$ oder $b_n \neq 0$ ist ein reelles trigonometrisches Polynom vom Grad n. Sei $\mathcal{T}_n$ die Menge aller trigonometrischen Polynome vom Grad $\leq n$ mit $\dim \mathcal{T}_n = 2n + 1$.

Wir erhalten aus der Eulerschen Formel $e^{it} = \cos t + i \sin t$ bzw. $e^{-it} = \cos t - i \sin t$,

$$\cos t = \frac{(e^{it} + e^{-it})}{2}, \qquad \sin t = \frac{(e^{it} - e^{-it})}{2i},$$

einen Zusammenhang zwischen der reellen und der komplexen Darstellung:

$$\begin{aligned}
\tilde{p}_n(t) &= \frac{a_0}{2} + \sum_{k=1}^{n} a_k \frac{(e^{ikt} + e^{-ikt})}{2} + b_k \frac{(e^{ikt} - e^{-ikt})}{2i} \\
&= \underbrace{\frac{a_0}{2}}_{:=c_0} + \sum_{k=1}^{n} \underbrace{\frac{(a_k - ib_k)}{2}}_{:=c_k} e^{ikt} + \underbrace{\frac{(a_k + ib_k)}{2}}_{:=c_{-k}} e^{-ikt} \\
&= \sum_{k=-n}^{n} c_k e^{ikt}.
\end{aligned}$$

Umgekehrt gilt:

$$\begin{aligned}
p_n(t) &= \sum_{k=-n}^{n} c_k e^{ikt} = \sum_{k=-n}^{n} c_k(\cos kt + i \sin kt) \\
&= c_0 \cdot \underbrace{\cos(0 \cdot t)}_{1} + \sum_{k=1}^{n} (c_k \cos kt + c_{-k} \underbrace{\cos(-k)t)}_{\cos kt} \\
&\quad + i \sum_{k=1}^{n} (c_k \sin kt + c_{-k} \underbrace{\sin(-k)t)}_{-\sin kt} \\
&= c_0 + \sum_{k=1}^{n} (c_k + c_{-k}) \cos kt + i(c_k - c_{-k}) \sin kt.
\end{aligned}$$

Wir erhalten:

Satz 1.26. *Ein reelles trigonometrisches Polynom*

$$\tilde{p}_n(t) = \frac{a_0}{2} + \sum_{k=1}^{n} a_k \cos(kt) + b_k \sin(kt), \quad a_k, b_k \in \mathbb{R}$$

lässt sich auch in komplexer Form darstellen

$$\tilde{p}_n(t) = \sum_{k=-n}^{n} c_k e^{ikt}$$

wobei $c_0 := \frac{a_0}{2}$, $c_k := \frac{a_k - ib_k}{2}$ *und* $c_{-k} := \frac{a_k + ib_k}{2} = \overline{c}_k$ *für* $k = 1, \ldots, n$. *Umgekehrt ist ein Polynom* $p_n(t) = \sum_{k=-n}^{n} c_k e^{ikt}$ *reell, wenn* c_0, $c_k + c_{-k}$ *und* $i(c_k - c_{-k})$ *für* $k = 1, \ldots, n$ *relle Zahlen sind, d.h., falls* $c_{-k} = \overline{c}_k$ *für* $k = 0, \ldots, n$ *gilt.*

Sei $N := 2n + 1$. Das bedeutet $\dim \mathcal{T}_n = N$.
Das **trigonometrische Interpolationsproblem** lässt sich folgendermaßen formulieren:
Gesucht ist ein trigonometrisches Polynom $p \in \mathcal{T}_n$ mit der Eigenschaft

$$p(x_k) = y_k, \quad k = 0, \ldots, N-1 \tag{1.6}$$

wobei $y_k \in \mathbb{C}$ (oder $y_k \in \mathbb{R}$) gegebene Daten sind, und $x_0 < x_1 < \ldots < x_{N-1} \in [0, 2\pi)$.

Satz 1.27. *Das trigonometrische Interpolationsproblem* (1.6) *besitzt genau eine Lösung.*

Beweis. (reeller Fall) Da die Basis $\{1, \sin x, \ldots, \sin(nx), \cos x, \ldots, \cos(nx)\}$ ein Tschebyscheff-System bildet (siehe Abschnitt 1.1), ist $\mathcal{T}_n$ ein Haarscher Raum. Die Behauptung folgt nun aus Satz 1.4. □

Sei nun $\mathcal{T}'_n = \operatorname{span}\{1, \sin(x), \cos(x), \ldots, \sin(n-1)x, \cos(n-1)x, \cos nx\}$, d.h. $\sin(nx)$ ist aus der Basis entfernt worden. Dann ist $\dim \mathcal{T}'_n = 2n$.
Setze nun $N := 2n$. Wir betrachten die äquidistanten Stützstellen

$$x_k = \frac{2\pi k}{N} \quad k = 0, \ldots, N-1.$$

Satz 1.28. *Seien für $N = 2n$, $n \in \mathbb{N}$, die Stützstellen $x_k = \frac{2\pi k}{N}$, $k = 0, \ldots, N-1$ und die Stützwerte $y_k \in \mathbb{R}, k = 0, \ldots, N-1$ gegeben. Dann besitzt das trigonometrische Interpolationsproblem $p(x_k) = y_k$, $k = 0, \ldots, N-1$ die eindeutige Lösung*

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

mit

$$\begin{aligned} a_j &:= \frac{2}{N} \sum_{k=0}^{N-1} y_k \cos(jx_k), \quad j = 0, \ldots, n, \\ b_j &:= \frac{2}{N} \sum_{k=0}^{N-1} y_k \sin(jx_k), \quad j = 1, \ldots, n-1. \end{aligned}$$

Ferner gilt

$$p(x) = \frac{1}{N} \sum_{k=0}^{N-1} y_k D'_n(x - x_k),$$

wobei $D'_n(x) := \sum_{k=-n+1}^{n-1} e^{ikx} + \frac{1}{2}(e^{inx} + e^{-inx})$ der n-te modifizierte **Dirichlet-Kern** *ist.*

Für den Beweis des Satzes benötigen wir folgendes Lemma:

Lemma 1.29. *Für* $\omega_N := e^{\frac{-2\pi i}{N}}$ *gilt*

$$\sum_{j=0}^{N-1} \omega_N^{jk} = \begin{cases} N & k \equiv 0 \bmod N, \\ 0 & k \not\equiv 0 \bmod N. \end{cases}$$

Beweis. 1) Sei $k \equiv 0 \bmod N$. Dann folgt

$$\sum_{j=0}^{N-1} \omega_N^{jk} = \sum_{j=0}^{N-1} \underbrace{(e^{-2\pi i/N})^{jk}}_{1} = N.$$

2) Mit $k \not\equiv 0 \bmod N$ erhalten wir eine geometrische Reihe

$$\Rightarrow \sum_{j=0}^{N-1} (\underbrace{\omega_N^k}_{\neq 1})^j = \frac{1-(\omega_N^k)^N}{1-\omega_N^k} = \frac{1-(\omega_N^N)^k}{1-\omega_N^k} = 0 \quad \text{mit} \quad \omega_N^N = 1.$$

□

Beweis. von Satz 1.28:
1) p hat in komplexer Schreibweise die Form

$$p(x) = \sum_{j=-n+1}^{n-1} c_j e^{ijx} + c_n \frac{(e^{inx} + e^{-inx})}{2}.$$

Setzen wir die Interpolationsbedingung ein, so ergibt sich

$$p(x_k) = \sum_{j=-n+1}^{n-1} c_j \underbrace{e^{ij\frac{2\pi k}{N}}}_{:=\omega_N^{-jk}} + c_n \underbrace{\cos\frac{2\pi nk}{N}}_{(-1)^k=\omega_N^{-nk}}, \quad k = 0, \ldots, N-1,$$

und damit

$$p(x_k) = y_k = \sum_{j=-n+1}^{n} c_j \omega_N^{-jk}, \quad k = 0, \ldots, N-1.$$

Mit Lemma 1.29 folgt für $r = -n+1, \ldots, n$:

$$\frac{1}{N} \sum_{k=0}^{N-1} \underbrace{p(x_k)}_{=y_k} \omega_N^{kr} = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{j=-n+1}^{n} c_j \omega_N^{-jk}\, \omega_N^{kr} = \frac{1}{N} \sum_{j=-n+1}^{n} c_j \underbrace{\sum_{k=0}^{N-1} \omega_N^{k(r-j)}}_{N\delta_{r-j,0}}.$$

Damit erhalten wir

$$\frac{1}{N}\sum_{k=0}^{N-1} y_k\,\omega_N^{kr} = \frac{1}{N}c_r\,N = c_r, \quad r = -n+1,\ldots,n. \tag{1.7}$$

Mit Satz 1.26 war $a_r = c_r + c_{-r}, b_r = i(c_r - c_{-r})$ für $r = 1,\ldots,n-1$ und $a_0 = 2c_0, a_n = 2c_n$. Das ergibt

$$\begin{aligned}
a_r &= \frac{1}{N}\sum_{k=0}^{N-1} y_k(\omega_N^{kr} + \omega_N^{-kr}) = \frac{1}{N}\sum_{k=0}^{N-1} y_k 2\cos(x_k r),\\
&\quad \text{mit } \omega_N^{kr} + \omega_N^{-kr} = e^{\frac{-2\pi kr}{N}i} + e^{\frac{2\pi kr}{N}i} = 2\cos(\tfrac{2\pi kr}{N}) = 2\cos(x_k r)\\
&\quad \text{für } r = 0,\ldots,n-1,\\
a_n &= \frac{2}{N}\sum_{k=0}^{N-1} y_k\omega_N^{nk} = \frac{2}{N}\sum_{k=0}^{N-1} y_k\cos(x_k n),\\
&\quad \text{mit } \omega_N^{nk} = (-1)^k = \cos(x_k n),\\
b_r &= \frac{1}{N}\sum_{k=0}^{N-1} y_k i(\omega_N^{kr} - \omega_N^{-kr}) = \frac{1}{N}\sum_{k=0}^{N-1} y_k 2\sin(x_k r),\\
&\quad \text{mit } i(\omega_N^{kr} - \omega_N^{-kr}) = i(e^{\frac{-2\pi kr}{N}i} - e^{\frac{2\pi kr}{N}i}) = -2i^2\sin(x_k r) = 2\sin(x_k r),\\
&\quad \text{für } r = 1,\ldots,n-1.
\end{aligned}$$

2) Die Eindeutigkeit der Lösung folgt aus der Invertierbarkeit der Koeffizientenmatrix des linearen Gleichungssystems (1.7).
3) Mit (1.7) folgt für das Interpolationspolynom

$$\begin{aligned}
p(x) &= \sum_{j=-n+1}^{n-1} c_j e^{ijx} + c_n\Big(\frac{e^{inx}+e^{-inx}}{2}\Big)\\
&= \sum_{j=-n+1}^{n-1}\Big(\frac{1}{N}\sum_{k=0}^{N-1} y_k \overbrace{\omega_N^{kj}}^{e^{-ix_k j}}\Big)e^{ijx} + \Big(\frac{1}{N}\sum_{k=0}^{N-1} y_k \underbrace{\omega_N^{kn}}_{(-1)^k = e^{-ix_k n}}\Big)\Big(\frac{e^{inx}+e^{-inx}}{2}\Big)\\
&= \frac{1}{N}\sum_{k=0}^{N-1} y_k\left(\sum_{j=-n+1}^{n-1} e^{ij(x-x_k)} + \frac{1}{2}\,(e^{in(x-x_k)} + e^{-in(x-x_k)})\right)\\
&= \frac{1}{N}\sum_{k=0}^{N-1} y_k D_n'(x-x_k).
\end{aligned}$$

□

Zur numerischen Berechnung des Interpolationspolynoms müssen entweder die Koeffizienten a_j, b_j aus Satz 1.28 oder alternativ

$$c_j = \frac{1}{N} \sum_{k=0}^{N-1} y_k \, \omega_N^{kj} \quad j = -n+1, \ldots, n$$

berechnet werden. Dafür lässt sich die schnelle diskrete Fouriertransformation (FFT) nutzen.

Definition 1.30. *Die* **diskrete Fourier-Transformation** *der Länge N ist eine Abbildung von $\mathbb{C}^N$ in $\mathbb{C}^N$ (DFT(N)), die jedem Vektor $\mathbf{a} = (a_j)_{j=0}^{N-1} \in \mathbb{C}^N$ den Vektor $\hat{\mathbf{a}} = (\hat{a}_k)_{k=0}^{N-1} \in \mathbb{C}^N$ mit*

$$\hat{a}_k := \sum_{j=0}^{N-1} a_j \omega_N^{jk}, \qquad k = 0, \ldots, N-1,$$

zuordnet. Der Vektor $\hat{\mathbf{a}}$ heißt die diskrete Fourier-Transformierte von $\mathbf{a}$.

Es gilt:

$$\hat{\mathbf{a}} = \mathbf{F}_N \mathbf{a}$$

mit $\omega_N := e^{\frac{-2\pi i}{N}}$ und

$$\mathbf{F}_N := (\omega_N^{jk})_{j,k=0}^{N-1} = \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & \omega_N & \omega_N^2 & & \omega_N^{N-1} \\ 1 & \omega_N^2 & \omega_N^4 & & \omega_N^{2(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega_N^{N-1} & \omega_N^{2(N-1)} & \ldots & \omega_N^{(N-1)(N-1)} \end{pmatrix}.$$

Die Matrix $\mathbf{F}_N$ heißt die N-te **Fouriermatrix**. $\mathbf{F}_N$ enthält nur N verschiedene Einträge.

Lemma 1.31. *Die Fouriermatrix $\mathbf{F}_N$ ist invertierbar, und es gilt*

$$\mathbf{F}_N^{-1} = \frac{1}{N} \, \overline{\mathbf{F}}_N = \frac{1}{N} \, (\omega_N^{-jk})_{j,k=0}^{N-1}.$$

Beweis. Sei $\mathbf{F}_N \, \overline{\mathbf{F}}_N = (a_{j,l})_{j,l=0}^{N-1}$. Dann folgt nach Lemma 1.29

$$a_{j,l} = \sum_{k=0}^{N-1} \omega^{jk} \, \overline{\omega}^{kl} = \sum_{k=0}^{N-1} \omega^{k(j-l)} = \left\{ \begin{array}{ll} N, & j = l \\ 0, & j \neq l \end{array} \right\} = N \cdot \delta_{j,l}.$$

□

Definition 1.32. *Die inverse diskrete Fourier-Transformation* ($IDFT(N)$) *der Länge* N *ist durch* $\hat{\mathbf{a}} \mapsto \mathbf{a} := \mathbf{F}_N^{-1}\ \hat{\mathbf{a}}$ *erklärt, d.h.*

$$a_j := \frac{1}{N}\sum_{k=0}^{N-1} \hat{a}_k \omega_N^{-jk}, \qquad j = 0, \ldots, N-1.$$

Bei direkter Berechnung der $DFT(N)$ benötigen wir $\mathcal{O}(N^2)$ arithmetische Operationen. Wir wollen einen Algorithmus herleiten, der nur $\mathcal{O}(N \log N)$ arithmetische Operationen braucht.

Sande-Tukey-Algorithmus (Radix-2-Algorithmus)
Sei $N = 2^t$ $(t > 1)$. Wir erhalten

$$\begin{aligned}
\hat{a}_k &= \sum_{j=0}^{N-1} a_j \omega_N^{jk} = \sum_{j=0}^{\frac{N}{2}-1} a_j \omega_N^{jk} + \sum_{j=0}^{\frac{N}{2}-1} a_{\frac{N}{2}+j} \underbrace{\omega_N^{(\frac{N}{2}+j)k}}_{(-1)^k \omega_N^{jk}} \\
&= \sum_{j=0}^{\frac{N}{2}-1} (a_j + (-1)^k a_{\frac{N}{2}+j}) \omega_N^{jk}, \quad k = 0, \ldots, N-1.
\end{aligned}$$

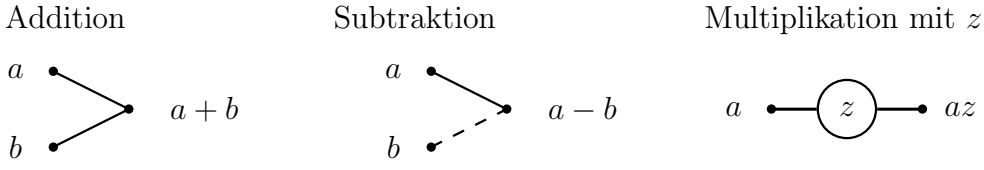
Also folgt

$$\begin{aligned}
\hat{a}_{2k} &= \sum_{j=0}^{\frac{N}{2}-1} (a_j + a_{\frac{N}{2}+j}) \omega_{\frac{N}{2}}^{jk}, \quad k = 0, \ldots, \frac{N}{2} - 1, \\
\hat{a}_{2k+1} &= \sum_{j=0}^{\frac{N}{2}-1} (a_j - a_{\frac{N}{2}+j}) \omega_N^j \omega_{\frac{N}{2}}^{jk}, \quad k = 0, \ldots, \frac{N}{2} - 1.
\end{aligned}$$

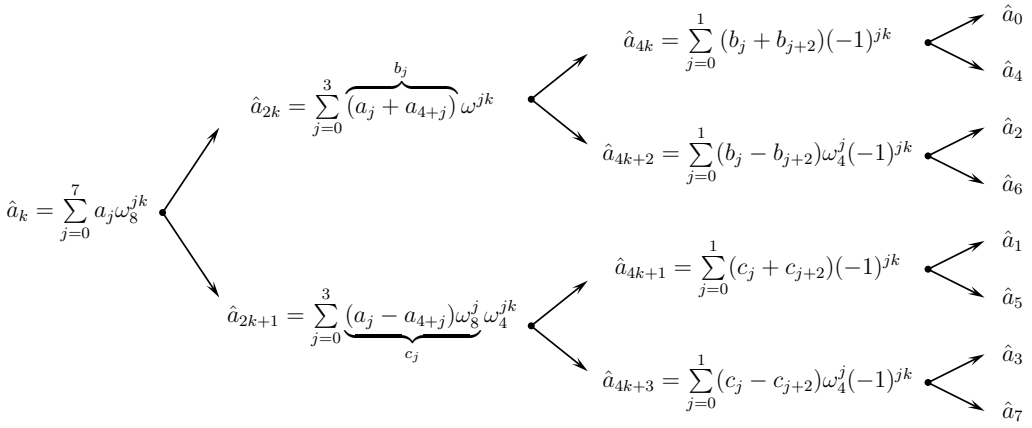
D.h., die $DFT(N)$ wird in 2 $DFT(\frac{N}{2})$ zerlegt. Für die Berechnung der "neuen Koeffizienten" für diese 2 $DFT(\frac{N}{2})$ benötigen wir $\frac{N}{2}$ Additionen, $\frac{N}{2}$ Subtraktionen und $\frac{N}{2}$ Multiplikationen mit den Drehfaktoren ω_N^j.
Auf die $DFT(\frac{N}{2})$ können wir die Idee erneut anwenden.

Beispiel: $N = 8$

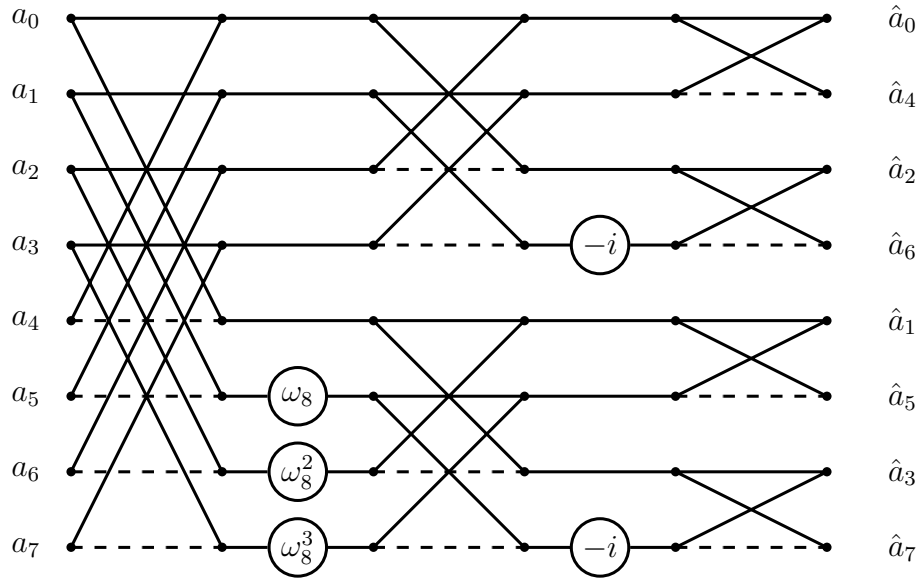
Die schnelle DFT lässt sich mit Hilfe eines Signalflussgraphen darstellen.
Im Signalflussgraph verwenden wir die folgenden Darstellungen.



Iterative Anwendung des Teile- und Herrsche-Prinzips führt im Fall $N = 8$ zu



Signalfluss-Graph:



Arithmetische Komplexität des Sande-Tukey-Algorithmus

Sei M_t die (höchstmögliche) Anzahl der komplexen Multiplikationen und A_t die Anzahl der komplexen Additionen für die $DFT(N)$ $(N = 2^t)$. Dann folgen aus dem Algorithmus die Rekursionen

$$M_t = 2M_{t-1} + \underbrace{\frac{N}{2}}_{=2^{t-1}}, \; A_t = 2A_{t-1} + \underbrace{N}_{=2^t.} \tag{1.8}$$

Satz 1.33. *Der Sande-Tukey-Algorithmus zur Berechnung der $DFT(N)$ mit $N =$*

2^t *benötigt höchstens*

$$\begin{array}{rcccl} M_t & = & \frac{N}{2}\,\log_2 N & = & 2^{t-1}t \quad \textit{komplexe Multiplikationen und} \\ A_t & = & N\log_2 N & = & 2^t t \quad \textit{komplexe Additionen.} \end{array}$$

Beweis. (durch vollständige Induktion)
1) Induktionsanfang: Betrachte die DFT der Länge 2 ($t = 1$):

$$\begin{array}{l} \hat{a}_0 = \sum_{j=0}^{1} a_j\,\omega_2^{j\cdot 0} = a_0 + a_1 \\ \hat{a}_1 = \sum_{j=0}^{1} a_j \omega_2^{j\cdot 1} = a_0 - a_1 \end{array} \Rightarrow M_1 = 1,\ \ A_1 = 2.$$

2) Induktionsschritt: Sei $M_t = 2^{t-1}\ t,\ A_t = 2^t t$. Dann folgt aus den Rekursionen in (1.8)

$$\begin{array}{ll} M_{t+1} & = 2M_t + 2^t = 2^t t + 2^t = 2^t\,(t+1), \\ A_{t+1} & = 2A_t + 2^{t+1}\ = 2(2^t t) + 2^{t+1}\ = 2^{t+1}\ (t+1). \end{array}$$

□

Beispiel: Sei $N = 512 = 2^9$.
1) Bei direkter Berechnung der DFT(512) benötigt man rund $N^2 = 512^2 = 262144$ komplexe Multiplikationen.

2) Bei Verwendung des Sande-Tukey-Algorithmus benötigen wir etwa $\frac{N}{2}\ \log_2 N = 256 \cdot 9 = 2304$ komplexe Multiplikationen.
Der Rechenaufwand wird also um das 100-fache reduziert!

Wir wenden die FFT nun auf das trigonometrische Interpolationsproblem an. Es war

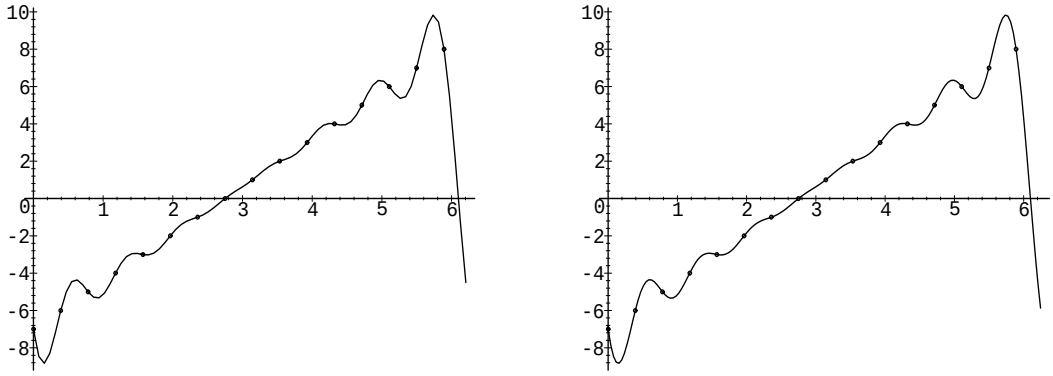
$$p(x) = \sum_{j=-n+1}^{n-1} c_j\, e^{ijx}\ + c_n \cos(nx) \quad \text{mit} \quad c_j := \frac{1}{N}\ \sum_{k=0}^{N-1} y_k\, \omega_N^{kj}$$

und $N = 2n$. Sei nun $N = 2^t$. Wir bezeichnen mit $\mathbf{y}\ = (y_0, y_1, \ldots, y_{N-1})^T$ den Vektor der Stützwerte.
Wir berechnen $\hat{\mathbf{y}}\ = \mathbf{F}_N \mathbf{y}$ mit dem schnellen FFT-Algorithmus und erhalten

$$c_j = \begin{cases} \frac{1}{N}\ \hat{y}_j & \text{für} \quad j = 0, \ldots, n, \\ \frac{1}{N}\ \hat{y}_{N+j} & \text{für} \quad j = -n+1, \ldots, -1, \end{cases}$$

denn

$$c_{N+j} = \frac{1}{N}\ \sum_{k=0}^{N-1} y_k \underbrace{\omega_N^{k(N+j)}}_{\omega_N^{kj}} = c_j.$$



Interpolationspolynome für die Sägezahnkurve mit $N = 16$ und $r = 5$ (rechts) sowie $N = 16$ und $r = 10$ (links).

Berechnung des trigonometrischen Interpolationspolynoms

Setze nun $\tilde{\mathbf{c}} = (\tilde{c}_j)_{j=0}^{N-1} := (c_0, \ldots, c_n, c_{-n+1}, \ldots, c_{-1}) \in \mathbb{C}^N$ so dass $\tilde{\mathbf{c}} = \frac{1}{N}\,\hat{\mathbf{y}}$.
Wir betrachten $p(x)$ auf dem feineren Gitter $\frac{2\pi l}{rN},\ l = 0, \ldots, Nr-1, r > 1, r \in \mathbb{N}$.
Setze $x = \frac{2\pi}{rN}(kr+s),\ k = 0, \ldots, N-1; s = 0, \ldots, r-1$. Wir erhalten für jedes $s \in \{0, \ldots, r-1\}$

$$
\begin{aligned}
p(\frac{2\pi(kr+s)}{rN}) &= \sum_{j=0}^{n-1} c_j e^{(\frac{2\pi ij(kr+s)}{rN})} + \sum_{j=-n+1}^{-1} c_j\ e^{(\frac{2\pi ij(kr+s)}{rN})} + c_n \cos(\frac{2\pi n(kr+s)}{rN}) \\
&= \sum_{j=0}^{n-1} c_j e^{(\frac{2\pi ij(kr+s)}{rN})} + \sum_{j'=n+1}^{N-1} c_{j'-N} e^{\frac{2\pi i(j'-N)(kr+s)}{rN}} + c_n \cos(\pi k + \frac{2\pi ns}{rn}) \\
&= \sum_{j=0}^{n-1} \tilde{c}_j \omega_N^{-jk}\, \omega_{rN}^{-js} + \sum_{j=n+1}^{N-1} \tilde{c}_j\, \omega_{rN}^{(-jkr-js+Nrk+Ns)} \\
&\quad + \tfrac{c_n}{2} \cdot (\omega_{rN}^{ns} \cdot (-1)^k + \omega_{rN}^{-ns}(-1)^k) \\
&= \sum_{j=0}^{N-1} \tilde{c}_j\, (\rho_s)_j\, \omega_N^{-jk} \qquad \text{(DFT der Länge } N\text{)}
\end{aligned}
$$

mit

$$
(\rho_s)_j := \begin{cases} \omega_{rN}^{-js}, & j = 0, \ldots, n-1, \\ \frac{1}{2}(\omega_{rN}^{-ns} + \omega_{rN}^{ns}) = \cos(\frac{\pi s}{r}), & j = n, \\ \omega_{rN}^{(N-j)s}, & j = n+1, \ldots N-1. \end{cases} \tag{1.9}
$$

Diese Werte können vorberechnet werden.

Wir erhalten den folgenden Algorithmus.

Algorithmus:

Gegeben: $N, r \in \mathbb{N},\ N, r \geq 2,\ N = 2^t$,
Stützstellen $x_k := \frac{2\pi k}{N}, \quad k = 0, \ldots, N-1$
Stützwerte $y_k := f(x_k), \quad k = 0, \ldots, N-1.$

Gesucht: $p(\frac{2\pi l}{rN})$ für $l = 0, \ldots, rN-1$, wobei $p(x_k) = y_k$.

1. Berechne mittels DFT(N)

$$\tilde{\mathbf{c}} \ = \frac{1}{N}\ \mathbf{F}_N \mathbf{y}.$$

2. Setze $\mathbf{p}_0 := \mathbf{y}$.

3. Für $s = 1, \ldots, r-1$ bilde die Vektoren

$$\mathbf{d}_s := \tilde{\mathbf{c}}\ \circ \rho_s = \begin{pmatrix} \tilde{c}_0(\rho_s)_0 \\ \vdots \\ \tilde{c}_{N-1}\ (\rho_s)_{N-1} \end{pmatrix} \quad \text{(Hadamard-Produkt)}$$

mit ρ_s aus (1.9).

4. Berechne mittels DFT(N) für $s = 1, \ldots, r-1$

$$\mathbf{p}_s := \overline{\mathbf{F}}_N \mathbf{d}_s = \mathbf{U}\mathbf{F}_N \mathbf{d}_s,$$

wobei

$$U := \begin{pmatrix} 1 & 0 & \ldots & 0 \\ 0 & & & 1 \\ \vdots & 0 & \iddots & \\ 0 & 1 & & 0 \end{pmatrix}$$

die Umkehrmatrix der Dimension N ist.

Ausgabe: $p(\frac{2\pi l}{rN})$ für $l = 0, \ldots, rN-1$, wobei

$$\big(p\big(\frac{2\pi(kr+s)}{rN}\big)\big)_{k=0}^{N-1} =: \mathbf{p}_s, \quad s = 0, \ldots, r-1.$$

Berechnungsaufwand (DFT mit Sande-Tukey-Algorithmus):

Schritt 1: $\frac{N}{2}\ \log_2 N + N$ komplexe Multiplikationen
$N \log N$ komplexe Additionen

Schritt 3: $(r-1) \cdot N$ komplexe Multiplikationen

Schritt 4: $(r-1) \cdot \frac{N}{2}\ \log_2 N$ komplexe Multiplikationen
$(r-1)N \log N$ komplexe Additionen

Zur Berechnung von $r\,N$ Funktionswerten des Interpolationspolynoms benötigen wir also insgesamt

$$\begin{array}{lll} rN+ & r\frac{N}{2}\,\log N & \text{komplexe Multiplikationen,} \\ & rN\log N & \text{komplexe Additionen.} \end{array}$$

1.6 Spline-Interpolation

Problem: Selbst bei glatten Funktionen $f : [a,b] \to \mathbb{R}$ kann bei Polynom-Interpolation der Interpolationsfehler zwischen f und dem durch die Interpolationsbedingung $p_n(x_k) = f(x_k),\ k = 0,\ldots,n$ festgelegten Polynom sehr groß werden.

Idee: Nutze nur Polynome niedrigen Grades zur Interpolation, die an den Stützstellen "gut zusammenpassen".

Definition 1.34. *Durch die Knotenfolge $X := (x_i)_{i=0}^n$ sei eine Zerlegung des Intervalls $[a,b]$ gegeben, wobei $a = x_0 < x_1 < \ldots < x_n = b$. Dann heißen die Funktionen aus dem linearen Raum*

$$S_m(X) := \{s \in C^{m-1}[a,b] : s|_{[x_i,x_{i+1}]} \in \Pi_m, 0 \le i \le n-1\}$$

polynomiale Spline-Funktionen (Splines) vom Grad *m auf der Zerlegung X.*

Beispiel:
1) $m = 1$ (lineare Splines)

$$S_1(x) = \{s \in C^0[a,b] : s|_{[x_i,x_{i+1}]} \in \Pi_1,\ 0 \le i \le n-1\}$$

2) $m = 3$ (kubische Splines)

$$S_3(X) = \{\ s \in C^2[a,b] : s|_{[x_i,x_{i+1}]} \in \Pi_3,\ 0 \le i \le n-1\}$$

Satz 1.35. *Durch $B := \{1, x, \ldots, x^m, (x-x_1)_+^m, \ldots, (x-x_{n-1})_+^m\}$ ist eine Basis von $S_m(X)$ gegeben. Insbesondere gilt* $\dim S_m(X) = m+n$.

Beweis. Zur Konstruktion von $s \in S_m(X)$ haben wir höchstens $m+n$ Freiheitsgrade, denn:
Auf dem Intervall $[x_0,x_1]$ ist ein beliebiges Polynom vom Grad m wählbar, wir haben also $m+1$ Freiheitsgrade. Das Polynom auf $[x_1,x_2]$ muss nun m Stetigkeits-

bzw. Differenzierbarkeitsbedingungen erfüllen, so dass nur noch ein zusätzlicher Freiheitsgrad entsteht.
Analog erhalten wir für die Intervalle $[x_2, x_3], \ldots, [x_{n-1}, x_n]$ jeweils einen weiteren Freiheitsgrad. Damit folgt dim $S_m(X) \leq m + n$.
Wir zeigen, dass die $m + n$ Funktionen in $B \subset S_m(X)$ linear unabhängig sind:
Sei

$$s(x) = \sum_{j=0}^{m} a_j x^j + \sum_{k=1}^{n-1} b_k (x - x_k)_+^m = 0 \qquad \forall\, x \in [a, b]\ .$$

Für $x \in [x_0, x_1]$ gilt $(x - x_k)_+^m = 0, \quad k = 1, \ldots, n-1$. Also muss gelten

$$\sum_{j=0}^{m} a_j x^j = 0 \quad \forall\, x \in [x_0, x_1] \qquad \Rightarrow \qquad a_0 = a_1 = \ldots = a_m = 0.$$

Für $x \in [x_1, x_2]$ gilt $(x - x_k)_+^m = 0,\ k = 2, \ldots, n-1$, d.h. $b_1 (x - x_1)_+^m = 0\ \forall\, x \in [x_1, x_2]$ und damit $b_1 = 0$. Analog lässt sich $b_2 = b_3 = \ldots = b_{n-1} = 0$ folgern. Da $1, x, \ldots, x^m, (x - x_1)_+^m, \ldots, (x - x_{n-1})_+^m \in S_m(X)$ ergibt sich die Behauptung. □

Die Basis in Satz 1.35 ist jedoch ungeeignet für numerische Berechnungen. Wir wollen daher eine numerisch stabilere Basis konstruieren, deren Elemente kompakten Träger besitzen.

Definition 1.36. *Zu allen $i \in \mathbb{Z}$ seien paarweise verschiedene Punkte $x_i \in \mathbb{R}$ mit $-\infty < \ldots < x_{-1} < x_0 < x_1 < \ldots < \infty$ vorgegeben. Dann heißen die durch*

$$N_{i,m}(t) := (x_{i+m} - x_i) \cdot u_{t,m-1}[x_i, \ldots, x_{i+m}]$$

mit $u_{t,m-1}(x) := (x - t)_+^{m-1}$ für $i \in \mathbb{Z}, m \in \mathbb{N}$, definierten Funktionen $N_{i,m} : \mathbb{R} \to \mathbb{R}$ **(normalisierte) B-Splines der Ordnung** *m zur Knotenfolge $X = (x_i)_{i \in \mathbb{Z}}$.*

Beispiel: $m = 1$:

$$u_{t,0}(x) = (x - t)_+^0 = \begin{cases} 1, & x > t, \\ 0, & x \leq t, \end{cases}$$

$$u_{t,0}[x_i, x_{i+1}] \ = \frac{u_{t,0}(x_{i+1}) - u_{t,0}(x_i)}{x_{i+1} - x_i}.$$

1. Fall : $x_i, x_{i+1} > t \quad \Rightarrow \quad u_{t,0}[x_i, x_{i+1}] \ = \frac{1-1}{x_{i+1}-x_i} = 0,$
2. Fall : $x_i, x_{i+1} \ \leq t \quad \Rightarrow \quad u_{t,0}[x_i, x_{i+1}] \ = \frac{0-0}{x_{i+1}-x_i} = 0,$
3. Fall : $x_i \leq t < x_{i+1} \quad \Rightarrow \quad u_{t,0}[x_i, x_{i+1}] \ = \frac{1-0}{x_{i+1}-x_i} = \frac{1}{x_{i+1}-x_i}.$

Somit erhalten wir den B-Spline vom Grad 0 in der Form

$$N_{i,1}(t) = \begin{cases} 1, & x_i \leq t < x_{i+1}, \\ 0, & \text{sonst.} \end{cases}$$

Nach Satz 1.10 (i) folgt für $N_{i,m}$:

$$N_{i,m}(t) = (x_{i+m} - x_i) \sum_{r=i}^{i+m} (x_r - t)_+^{m-1} \Big(\prod_{\substack{k=i \\ k \neq r}}^{i+m} \frac{1}{x_r - x_k} \Big).$$

Ein Vergleich mit der Integraldarstellung dividierter Differenzen (Satz 1.13) liefert

Satz 1.37. *Es gilt*

$$f[x_i, \ldots, x_{i+m}] = \frac{1}{x_{i+m} - x_i} \cdot \frac{1}{(m-1)!} \int_{\mathbb{R}} f^{(m)}(t) N_{i,m}(t) dt,$$

d.h., der B-Spline ist bis auf Normierung der Peano-Kern der dividierten Differenzen. (Erinnerung: Peano-Kern $G_{n-1}(t) = \sum_{k=0}^{n} w_k \cdot \frac{(x_k - t)_+^n}{(n-1)!}$, $w_k = \prod_{\substack{l=0 \\ l \neq k}}^{n} \frac{1}{(x_k - x_l)}$.*)*

Wir leiten nun eine einfache Rekursionsformel für die Berechnung von $N_{i,m}$ her.

Satz 1.38. *Die B-Splines* $N_{i,m}$ *genügen für* $m \geq 2$ *der Rekursionsformel*

$$N_{i,m}(t) = \frac{t - x_i}{x_{i+m-1} - x_i} N_{i,m-1}(t) + \frac{x_{i+m} - t}{x_{i+m} - x_{i+1}} N_{i+1,m-1}(t).$$

Beweis. Es gilt

$$u_{t,m-1}(x) = (x - t)_+^{m-1} = (x - t) \cdot (x - t)_+^{m-2} = f(x) \cdot u_{t,m-2}(x),$$

wobei $f(x) := (x - t)$. Wir wenden die Leibniz-Formel für dividierte Differenzen (Satz 1.12) an und erhalten

$$u_{t,m-1}[x_i, \ldots, x_{i+m}] = \sum_{k=i}^{i+m} f[x_i, \ldots, x_k] \; u_{t,m-2}[x_k, \ldots, x_{i+m}]$$

$$= f[x_i] u_{t,m-2}[x_i, \ldots, x_{i+m}] + f[x_i, x_{i+1}] u_{t,m-2}[x_{i+1}, \ldots, x_{i+m}] + 0,$$

denn $f[x_i, \ldots, x_k] = 0$ für $k > i + 1$, da f ein Polynom vom Grad 1 ist (vgl. Satz 1.10(iv)). Mit

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

folgt

$$u_{t,m-1}[x_i, \ldots, x_{i+m}] = (x_i - t) \, u_{t,m-2}[x_i, \ldots, x_{i+m}] + 1 \cdot u_{t,m-2}[x_{i+1}, \ldots, x_{i+m}].$$

Unter Verwendung der Rekursionsformel für die dividierten Differenzen (Satz 1.10 (iii)) erhält man

$$\begin{aligned}
& u_{t,m-1}[x_i,\ldots,x_{i+m}] \\
&= (x_i - t)\left[\frac{u_{t,m-2}[x_{i+1},\ldots,x_{i+m}] - u_{t,m-2}[x_i,\ldots,x_{i+m-1}]}{x_{i+m}-x_i}\right] \\
&\quad + u_{t,m-2}[x_{i+1},\ldots,x_{i+m}] \\
&= \underbrace{(1+\frac{x_i-t}{x_{i+m}-x_i})}_{(\frac{x_{i+m}-t}{x_{i+m}-x_i})}\, u_{t,m-2}[x_{i+1},\ldots,x_{i+m}] - (\frac{x_i-t}{x_{i+m}-x_i}) u_{t,m-2}[x_i,\ldots,x_{i+m-1}].
\end{aligned}$$

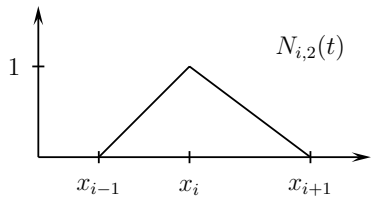
Damit folgt

$$\begin{aligned}
N_{i,m}(t) &= (x_{i+m}-x_i)\cdot u_{t,m-1}[x_i,\ldots,x_{i+m}\,] \\
&= (x_{i+m}-t)\underbrace{u_{t,m-2}[x_{i+1},\ldots,x_{i+m}]}_{\frac{1}{(x_{i+m}-x_{i+1})}N_{i+1,m-1}(t)} - (x_i-t)\underbrace{u_{t,m-2}[x_i,\ldots,x_{i+m-1}]}_{\frac{1}{(x_{i+m-1}-x_i)}N_{i,m-1}(t)}.
\end{aligned}$$

□

Beispiel: $m = 2$: linearer Spline (Hutfunktion)

$$\begin{aligned}
N_{i,2}(t) &= \frac{t-x_i}{x_{i+1}-x_i}\underbrace{N_{i,1}(t)}_{\chi_{[x_i,x_{i+1})}(t)} + \frac{x_{i+2}-t}{x_{i+2}-x_{i+1}}\underbrace{N_{i+1,1}(t)}_{\chi_{[x_{i+1},x_{i+2})}(t)} \\
&= \begin{cases} \frac{t-x_i}{x_{i+1}-x_i}, & t\in[x_i,x_{i+1}), \\ \frac{x_{i+2}-t}{x_{i+2}-x_{i+1}}, & t\in[x_{i+1},x_{1+2}), \\ 0, & \text{sonst.} \end{cases}
\end{aligned}$$



Eigenschaften von B-Splines

Satz 1.39. *Für $i \in \mathbb{Z}$ und $m \in \mathbb{N}$ seien $N_{i,m}$ die B-Splines zur Knotenfolge $X = (x_i)_{i\in\mathbb{Z}}$. Dann gilt:*

(i) $N_{i,m}$ verschwindet außerhalb von $[x_i, x_{i+m}]$, d.h. $\operatorname{supp} N_{i,m} = [x_i, x_{i+m}]$ *(kompakter Träger).*

(ii) $N_{i,m}$ ist in (x_i, x_{i+m}) positiv.

(iii) Die B-Splines bilden eine "Zerlegung der Eins", d.h.

$$\sum_{i\in\mathbb{Z}} N_{i,m}(t) = 1 \quad \forall\, t \in \mathbb{R}.$$

Beweis. Der Beweis erfolgt mittels vollständiger Induktion bzgl. der Ordnung m.
(i) Für $m = 1$ gilt $N_{i,1} = \chi_{[x_i,x_{i+1})}$, d.h. supp $N_{i,1} = [x_i, x_{i+1}]$.
Sei nun supp $N_{i,m-1} \in [x_i, x_{i+m-1}]$ $\forall i \in \mathbb{Z}, m \geq 2$. Dann folgt aus der Rekursionsformel

$$\text{supp}\, N_{i,m} \subseteq \text{supp}\ N_{i,m-1} \cup \text{supp}\ N_{i+1,m-1} = [x_i, x_{i+m}].$$

(ii) Für $m = 1$ ist die Behauptung richtig, da $N_{i,1}(t) > 0$ für $t \in (x_i, x_{i+1})$.
Nach der Rekursionsformel ist nun $N_{i,m}$ für $t \in (x_i, x_{i+m})$ eine positive Linearkombination von $N_{i,m-1}$, und $N_{i+1,m-1}$ und damit selbst positiv.
(iii) Es gilt für $m = 1$

$$\sum_{i\in\mathbb{Z}} N_{i,1}(t) = \sum_{i\in\mathbb{Z}} \chi_{[x_i,x_{i+1})}(t) = 1 \quad \forall\, t \in \mathbb{R}.$$

Die Behauptung sei nun für $m-1$ ($m \geq 2$) richtig. Mit der Rekursionsformel ergibt sich

$$\begin{aligned}
\sum_{i\in\mathbb{Z}} N_{i,m}(t) &= \sum_{i\in\mathbb{Z}} \frac{(t - x_i)}{(x_{i+m-1} - x_i)} N_{i,m-1}(t) + \frac{(x_{i+m} - t)}{(x_{i+m} - x_{i+1})} N_{i+1,m-1}(t) \\
&= \sum_{i\in\mathbb{Z}} \frac{(t - x_i)}{(x_{i+m-1} - x_i)} N_{i,m-1}(t) + \sum_{i'\in\mathbb{Z}} \frac{(x_{i'+m-1} - t)}{(x_{i'+m-1} - x_{i'})} N_{i',m-1}(t) \\
&= \sum_{i\in\mathbb{Z}} \frac{(t - x_i + x_{i+m-1} - t)}{(x_{i+m-1} - x_i)} N_{i,m-1}(t) = 1.
\end{aligned}$$

□

Satz 1.40. *Die B-Splines*

$$N_{i,m+1}(t) = (x_{i+m+1} - x_i) \cdot u_{t,m}[x_i, \ldots, x_{i+m+1}]$$

der Ordnung $m+1$ (Grad m) für $-m \leq i \leq n-1$ bilden eine Basis des Splineraumes $S_m(X)$, wobei $x_{-m} < x_{-m+1} < \ldots < \underbrace{x_0 = a < x_1 \ldots < x_n = b}_{:=X} < \ldots < x_{n+m}$.

Beweis. Aufgrund der Definition mittels $u_{t,m}(x) = (x-t)_+^m$ sind die $n+m$ B-Splines $N_{i,m+1}(t)$ in $S_m(X)$ enthalten. Wegen $\dim S_m(X) = n+m$ brauchen wir nur noch die lineare Unabhängigkeit der $N_{i,m+1}(t)$ für $i = -m, \ldots, n-1$ zeigen. Nach Satz 1.35 bildet $\{1, x, \ldots, x^m, (x-x_1)_+^m, \ldots, (x-x_{n-1})_+^m\}$ eine Basis von $S_m(X)$. Folglich bildet auch $\{(x-x_{-m})_+^m, \ldots, (x-x_0)_+^m, (x-x_1)_+^m \ldots (x-x_{n-1})_+^m\}$ eine Basis von $S_m(X)$, denn

$$(x-x_p)_+^m = (\sum_{k=0}^{m} \binom{m}{k} x^k (-x_p)^{m-k})_+, \quad p = -m, \ldots, 0,$$

werden in $[a,b]$ durch $\{1, x, \ldots, x^m\}$ erzeugt und sind linear unabhängig. Aus der Definition von $N_{i,m+1}$ folgt

$$N_{i,m+1}(t) = (x_{i+m-1} - x_i) \cdot u_{t,m}[x_i, \ldots, x_{i+m+1}] = (x_{i+m-1} - x_i) \cdot v_{t,m}[x_i, \ldots, x_{i+m+1}]$$

mit $v_{t,m}(x) := (-1)^{m+1}\ (t-x)_+^m$, denn

$$\begin{aligned} u_{t,m}(x) - v_{t,m}(x) &= (x-t)_+^m - (-1)^{m+1}(t-x)_+^m \\ &= \left\{ \begin{array}{ll} (x-t)^m, & x \geq t \\ 0, & x < t \end{array} \right\} + (-1)^m \cdot \left\{ \begin{array}{ll} 0, & x \geq t \\ (t-x)^m, & x < t \end{array} \right\} \\ &= (x-t)^m \ \in \Pi_m, \end{aligned}$$

und damit $(u_{t,m} - v_{t,m})[x_i, \ldots, x_{i+m+1}]\ = 0$.
Also folgt für $N_{i,m+1}$ nach Satz 1.10(i) die Darstellung

$$N_{i,m+1}\ (t) = \sum_{r=i}^{i+m+1} (t-x_r)_+^m \underbrace{(-1)^{m+1}\ (x_{i+m+1} - x_i) \prod_{\substack{k=i \\ k \neq r}}^{i+m+1} \frac{1}{(x_r - x_k)}}_{=:a_{r,i}}.$$

Für $t \in [a,b]$ erhalten wir nun

$$\begin{aligned} N_{-m,m+1}(t) &= a_{-m,-m}(t-x_{-m})_+^m + \qquad\qquad \ldots + a_{1,-m}(t-x_1)_+^m, \\ N_{-m+1,m+1}(t) &= 0 + a_{-m+1,-m+1}(t-x_{-m+1})_+^m + \ldots + a_{2,m+1}\ (t-x_2)_+^m, \\ \vdots \qquad & \vdots \qquad\qquad\quad \ddots \\ N_{m-1,m+1}(t) &= 0 + \qquad 0 + \ldots \qquad\qquad + 0 + a_{n-1,n-1}(t-x_{n-1})_+^m. \end{aligned}$$

Dabei ist $a_{k,k}$ für $k = -m, \ldots, n-1$, von Null verschieden, und die Koeffizientenmatrix ist invertierbar. □

Interpolationsproblem:
Gegeben seien die Knotenfolge

$$x_{-m} < x_{-m+1} \ < \ldots < x_{-1} < x_0 = a < x_1 \ldots < x_n = b < x_{n+1} \ < \ldots < x_{n+m}$$

und ein Intervall $[a, b]$ mit $x_0 = a$, $x_n = b$, sowie die Funktionswerte $y_k = f(x_k)$ für $k = 0, \ldots, n$.
Gesucht ist eine Splinefunktion $s \in S_m(X)$, so dass

$$s(x_k) = y_k, \quad k = 0, \ldots, n,$$

erfüllt ist.

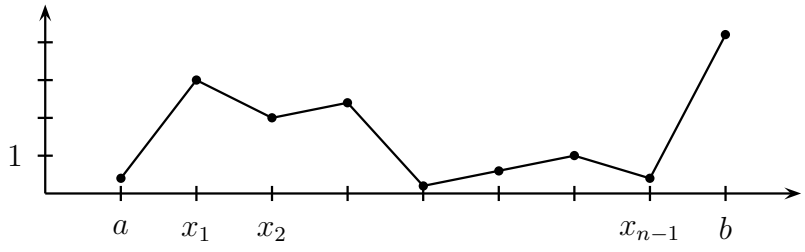
Da $\dim S_m(X) = n + m$, lässt sich mit Hilfe der $n + 1$ Interpolationsbedingungen nur für $m = 1$ (lineare Splines) die Splinefunktion eindeutig bestimmen. Für $m > 1$ benötigen wir Zusatzbedingungen zur Berechnung von s.
Wir betrachten hier nur die beiden Fälle $m = 1$ (lineare Splines) und $m = 3$ (kubische Splines) genauer.

Interpolation mit linearen Splines
Basisfunktionen:

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

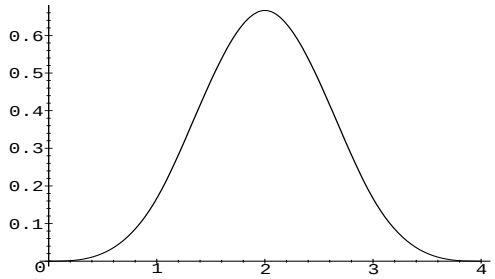
Wir erhalten $S_1(X) = \text{span}\,\{N_{-1,2}, N_{0,2}, \ldots, N_{n-1,2}\}$.
Offenbar gilt:

$$N_{i,2}(x_k) = \begin{cases} 1, & k = i + 1, \\ 0, & \text{sonst.} \end{cases}$$



Damit erfüllt die Funktion

$$s(x) = \sum_{i=-1}^{n-1} y_{i+1} \ N_{i,2}(x)$$



Kardinaler kubischer B-spline $N_{0,4}$ für die Knotenfolge $x_i = i$ $(i \in \mathbb{Z})$.

die Interpolationsbedingungen $s(x_k) = y_k,\ k = 0, \ldots, n.$

Interpolation mit kubischen Splines

Wir betrachten hier die äquidistante Knotenverteilung

$$x_i = x_0 + i \cdot h, \qquad h > 0, \qquad i = -3, \ldots, n+3.$$

Basisfunktionen:

$$N_{i,4}(x) =$$

$$= \frac{1}{6h^3} \begin{cases} (x - x_i)^3, & x \in [x_i, x_{i+1}), \\ h^3 + 3h^2(x - x_{i+1}) + 3h(x - x_{i+1})^2 - 3(x - x_{i+1})^3, & x \in [x_{i+1}, x_{i+2}), \\ h^3 + 3h^2(x_{i+3} - x) + 3h(x - x_{i+3})^2 - 3(x_{i+3} - x)^3, & x \in [x_{i+2}, x_{i+3}), \\ (x_{i+4} - x)^3, & x \in [x_{i+3}, x_{i+4}), \\ 0, & \text{sonst.} \end{cases}$$

Wegen $\dim S_m(X) = 3+n$ benötigen wir zur Berechnung des Interpolationssplines noch zwei zusätzliche Bedingungen.

(a) Hermite-Endbedingungen: $s'(a) = y'_0 = f'(x_0), \quad s'(b) = y'_n = f'(x_n).$

(b) Natürliche Endbedingungen: $s''(a) = s''(b) = 0.$

(c) Periodische Endbedingungen: $s'(a) = s'(b), \quad s''(a) = s''(b).$

Wir verwenden den Ansatz

$$s(x) = \sum_{i=-3}^{n-1} \alpha_i N_{i,4}(x)$$

und setzen die Interpolationsbedingungen ein:

$$s(x_k) = \sum_{i=-3}^{n-1} \alpha_i \, N_{i,4}(x_k), \quad k = 0, \ldots, n.$$

Aus den Endbedingungen erhalten wir
(a)

$$\begin{aligned} s'(x_0) &= \sum_{i=-3}^{-1} \alpha_i N'_{i,4}(x_0) = y'_0 = f'(x_0), \\ s'(x_n) &= \sum_{i=n-1}^{n-3} \alpha_i N'_{i,4}(x_0) = y'_n = f'(x_n), \end{aligned}$$

(b)

$$\begin{aligned} s''(x_0) &= \sum_{i=-3}^{-1} \alpha_i N''_{i,4}(x_0) = 0, \\ s''(x_n) &= \sum_{i=n-1}^{n-3} \alpha_i N''_{i,4}(x_n) = 0, \end{aligned}$$

(c)

$$\begin{aligned} \sum_{i=-3}^{-1} \alpha_i N'_{i,4}(x_0) &= \sum_{i=n-3}^{n-1} \alpha_i N'_{i,4}(x_n), \\ \sum_{i=-3}^{-1} \alpha_i N''_{i,4}(x_0) &= \sum_{i=n-3}^{n-1} \alpha_i N''_{i,4}(x_n). \end{aligned}$$

Zum Aufstellen des linearen Gleichungssystems benötigen wir die Werte von $N_{i,4}(x_k)$, $N'_{i,4}(x_k)$ und $N''_{i,4}(x_k)$:

	x_i	x_{i+1}	x_{i+2}	x_{i+3}	x_{i+4}
$N_{i,4}(x)$	0	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$	0
$N'_{i,4}(x)$	0	$\frac{1}{2h}$	0	$-\frac{1}{2h}$	0
$N''_{i,4}(x)$	0	$\frac{1}{h^2}$	$-\frac{2}{h^2}$	$\frac{1}{h^2}$	0

Wir erhalten also aus den Interpolationsbedingungen

$$\begin{aligned} s(x_k) &= \alpha_{k-3}\, N_{k-3,4}(x_k) + \alpha_{k-2} N_{k-2,4}(x_k) + \alpha_{k-1} N_{k-1,4}(x_k) = y_k \\ &= \frac{1}{6}\alpha_{k-3} + \frac{2}{3}\alpha_{k-2} + \frac{1}{6}\alpha_{k-1} = y_k, \quad k = 0, \ldots, n. \end{aligned}$$

und entsprechende Gleichungen aus den Endbedingungen.
Es ergeben sich folgende lineare Gleichungssysteme:

(a) Für Hermite-Endbedingungen:

$$\frac{1}{6}\begin{pmatrix} -\frac{3}{h} & 0 & \frac{3}{h} & 0 & 0 & \ldots & 0 \\ 1 & 4 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 4 & 1 & 0 & & \vdots \\ 0 & 0 & 1 & 4 & 1 & \ddots & \\ \vdots & & \ddots & \ddots & \ddots & \ddots & 0 \\ & & & 0 & 1 & 4 & 1 \\ 0 & \ldots & \ldots & 0 & -\frac{3}{h} & 0 & \frac{3}{h} \end{pmatrix} \begin{pmatrix} \alpha_{-3} \\ \alpha_{-2} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \alpha_{n-1} \end{pmatrix} = \begin{pmatrix} y_0' \\ y_0 \\ y_1 \\ \vdots \\ \vdots \\ y_n \\ y_n' \end{pmatrix},$$

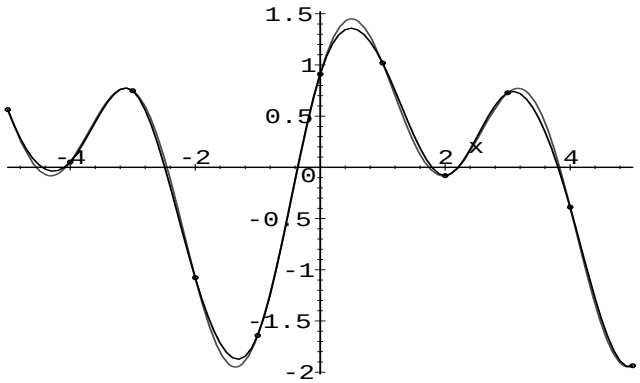
(b) Für natürliche Endbedingungen:

$$\frac{1}{6}\begin{pmatrix} \frac{6}{h^2} & -\frac{12}{h^2} & \frac{6}{h^2} & & & & \\ 1 & 4 & 1 & & 0 & & \\ & \ddots & \ddots & \ddots & & & \\ & 0 & \ddots & \ddots & \ddots & & \\ & & & 1 & 4 & 1 & \\ & & & \frac{6}{h^2} & -\frac{12}{h^2} & \frac{6}{h^2} & \end{pmatrix} \begin{pmatrix} \alpha_{-3} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \alpha_{n-1} \end{pmatrix} = \begin{pmatrix} 0 \\ y_0 \\ y_1 \\ \vdots \\ \vdots \\ y_n \\ 0 \end{pmatrix},$$

(c) Für periodische Endbedingungen:

$$\frac{1}{6}\begin{pmatrix} -\frac{3}{h} & 0 & \frac{3}{h} & 0 & \ldots & 0 & \frac{3}{h} & 0 & -\frac{3}{h} \\ \frac{6}{h^2} & -\frac{12}{h^2} & \frac{6}{h^2} & 0 & \ldots & 0 & -\frac{6}{h^2} & \frac{12}{h^2} & -\frac{6}{h^2} \\ 1 & 4 & 1 & 0 & \ldots & 0 & 0 & 0 & 0 \\ & \ddots & \ddots & \ddots & & & & & \\ & & & & & & & 0 & \\ & 0 & & & & & & & \\ & & & & & 1 & 4 & 1 & 0 \\ & & & & & 0 & 1 & 4 & 1 \end{pmatrix} \begin{pmatrix} \alpha_{-3} \\ \\ \vdots \\ \\ \vdots \\ \\ \alpha_{n-1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ y_0 \\ y_1 \\ \vdots \\ \\ y_{n-1} \end{pmatrix}.$$

Diese linearen Gleichungssysteme lassen sich mit einem Aufwand von $\mathcal{O}(N)$ Operationen berechnen, und die Koeffizientenmatrizen sind invertierbar.



Kubische Spline-Interpolante für $f(x) = \sin(2x+1) + \cos(x - \frac{3}{2})$.

Beispiel:
Wir wollen die Funktion $f(x) = \sin(2x+1) + \cos(x - \frac{3}{2})$ im Intervall $[-5, 5]$ an den Stützstellen $-5 + k$, $k = 0, 1 \ldots, 10$ mittels kubischer Splines interpolieren. Weiterhin wählen wir die Hermite-Endbedingungen

$$s'(-5) = f'(-5) \qquad s'(5) = f'(5).$$

Wir erhalten die folgende sehr gute Interpolation von f.

Zur Konvergenz kubischer Interpolationssplines:

Satz 1.41. *Sei eine Knotenfolge durch*

$$X_h = \{a = x_0, x_0 + h, x_0 + 2h, \ldots, x_0 + nh = b\}$$

gegeben. Sei $f \in C^4([a,b])$ und $L := \max_{x\in[a,b]} |f^{(4)}(x)|$. Sei ferner $s \in S_3(X_h)$ die kubische Spline-Interpolierende mit

$$s(a + kh) \ = f(a + kh), \quad k = 0, \ldots, n,$$

sowie $s'(a) = f'(a), s'(b) = f'(b)$ (Hermite-Endbedingungen).
Dann gilt für alle $x \in [a, b]$

$$|f^{(i)}(x) - s^{(i)}(x)| \leq 2L\, h^{4-i}, \quad i = 0, 1, 2, 3.$$

Beweis. Siehe Stoer, Numerische Mathematik I, S. 91 - 95. □

2 Approximation

Bei der Interpolation von $f \in C[a,b]$ wird eine einfach berechenbare Funktion h aus einem Untervektorraum von $C[a,b]$ gesucht, die in einer gewissen Anzahl von Punkten mit f übereinstimmt. Bei der Approximation suchen wir eine einfach strukturierte Funktion h, die die Funktion f in $[a,b]$ gut darstellt, so dass $\|f(x) - h(x)\|$ für alle $x \in [a,b]$ möglichst klein wird.

2.1 Existenz von Bestapproximation

Wir betrachten die folgenden Normen für $f \in C[a,b]$:

$$\|f\|_\infty := \max_{x\in[a,b]} |f(x)| \qquad \text{(Maximumnorm, Tschebyscheff-Norm)}$$

bzw.

$$\|f\|_p := \Big(\int_a^b |f(x)|^p dx \Big)^{\frac{1}{p}} \quad (L_p - \text{Norm}, 1 \le p < \infty).$$

Definition 2.1. *Sei U ein Untervektorraum des normierten Vektorraums V mit der Norm $\|.\| = \|.\|_V$. Ein Element h_0 von U heißt* **beste Approximation (Proximum)** *von $f \in V$, wenn*

$$\|f - h_0\|_V \le \|f - h\|_V \qquad \forall\, h \in U \tag{2.1}$$

gilt. Der optimale Wert

$$E_U(f) := \inf_{h\in U} \|f - h\|_V$$

heißt **Minimalabweichung** *von f bzgl. U. Das Problem* (2.1)*, eine beste Approximation $h_0 \in U$ von f zu finden, heißt* **lineares Approximationsproblem**.

Beispiel: Für $V = C[a,b]$ mit der Norm, $\|.\|_V = \|.\|_\infty$ spricht man von Tschebyscheff-Approximation.

Als Untervektorraum von $C[a,b]$ können wir z.B.

$U = \Pi_n$	(algebraische Polynome bis zum Grad n)
$U = \mathcal{T}_n$	(trigonometrische Polynome vom Grad n)
$U = S_m(X)$	(Spline-Raum)

wählen.

Satz 2.2. *Es sei U ein endlichdimensionaler Untervektorraum des normierten Vektorraums V mit der Norm $\|.\| = \|.\|_V$. Dann gibt es zu jedem $f \in V$ eine beste Approximation.*

Beweis. Falls es eine beste Approximation h_0 für f gibt, gilt

$$\|h_0\| \leq \|f - h_0\| + \|f\|.$$

Andererseits gilt nach Definition 2.1

$$\|f - h_0\| \leq \|f - 0 \underset{\in U}{\|} = \|f\|$$

da die Nullfunktion im Untervektorraum U enthalten ist. Also folgt

$$\|h_0\| \leq 2\|f\|.$$

Wir können uns daher bei der Suche nach h_0 auf die Kugel

$$K := \{\ h \in U : \|h\| \leq 2\|f\|\}$$

beschränken. Diese Kugel ist für endlichdimensionale Räume U kompakt. Die Funktion $\|f - h\|$ ist auf K stetig, hat daher auf K ein Minimum. Dies liefert eine Bestapproximation. □

Die Bestapproximation muss nicht eindeutig sein. Wir erhalten

Satz 2.3. *Die Menge der besten Approximationen aus U an ein Element $f \in V$ ist konvex.*

Beweis. Seien $h_0, \tilde{h}_0$ zwei beste Approximationen an $f \in V$, d.h.

$$\|f - h_0\| = \|f - \tilde{h}_0\| = E_U(f).$$

Dann folgt für $h = \lambda\, h_0 + (1 - \lambda)\tilde{h}_0$ und beliebiges $\lambda \in [0, 1]$

$$\begin{aligned} \|f - h\| &= \|f - \lambda h_0 - (1 - \lambda)\tilde{h}_0\| = \|\lambda(f - h_0) + (1 - \lambda)(f - \tilde{h}_0\| \\ &\leq \lambda\, \|f - h_0\| + (1 - \lambda)\, \|f - \tilde{h}_0\| = E_U(f). \end{aligned}$$

□

Ob die Bestapproximation h_0 an ein $f \in V$ eindeutig ist oder nicht hängt von den Eigenschaften der Norm $\|.\|_V$ ab.

Definition 2.4. *Die Norm $\|\cdot\| = \|\cdot\|_V$ eines Vektorraumes V heißt* **streng konvex**, *wenn für jedes Paar $x, y \in V$ mit $x \neq y$ und $\|x\| = \|y\| = 1$ gilt*

$$\|x + y\| < 2.$$

Beispiele:

1) Sei $V = \mathbb{R}^2$ mit der Maximumnorm $\|\mathbf{x}\|_V = \max(|x_1|, |x_2|)$, $\mathbf{x} := (x_1, x_2)^T$. Wähle nun $U = \mathbb{R} = \{(x, 0)^T \ : x \in \mathbb{R}\}$. Sei nun $\mathbf{y} \ = (0, 1)^T$. Dann hat jeder Punkt $\mathbf{x}_0 = (x, 0)^T \ \in U$ mit $|x| \leq 1$ den Abstand 1 von $\mathbf{y}$ und ist damit beste Approximation an $\mathbf{y}$, denn

$$\|(0, 1)^T - (x, 0)^T \ \| = \|(-x, 1)\| = 1.$$

2) Sei $V = \mathbb{R}^2$ mit der Euklidischen Norm $\|\mathbf{x}\|_V = \sqrt{x_1^2 + x_2^2}$ und U wie in Beispiel 1. Dann ist nur $\mathbf{x}_0 = (0, 0)^T$ Bestapproximation von $\mathbf{y} = (0, 1)^T \in V$, denn

$$\|(0, 1)^T - (x, 0)^T\|_2 = \|(-x, 1)^T \ \|_2 = \sqrt{x^2 + 1} \ \geq \sqrt{0 + 1} \ = 1.$$

In diesem Fall ist die beste Approximation also eindeutig bestimmt.

Die Euklidische Norm ist streng konvex, denn für

$$\mathbf{x} \ = (x_1, x_2)^T \ , \ \mathbf{y} \ = (y_1, y_2)^T \quad \text{mit} \quad \|\mathbf{x}\|_2 = \|\mathbf{y}\|_2 = 1$$

folgt

$$\|\mathbf{x} \ + \mathbf{y}\|_2^2 = (x_1 + y_1)^2 + (x_2 + y_2)^2 = \underbrace{x_1^2 + x_2^2}_{1} + \underbrace{y_1^2 + y_2^2}_{1} + \underbrace{2x_1y_1 \ + 2x_2y_2}_{<2} \ < 4.$$

3) Sei $V = C[a, b]$ mit der Norm $\|f\|_\infty = \max\limits_{x \in [a,b]} |f(x)|$. Diese Norm ist nicht streng konvex. Wähle z.B.

$$f(x) \equiv 1 \quad \text{in} \quad [-1, 1], \qquad g(x) = x \quad \text{in} \quad [-1, 1].$$

Dann folgt

$$\|f\| = \|g\| = 1, \quad f \neq g \quad \text{und} \quad \|f + g\| = \|1 + x\| = 2.$$

4) Sei $V = C[a, b]$ mit der Norm

$$\|f\|_V^2 = \int_a^b |f(x)|^2 dx.$$

Diese Norm ist streng konvex.

Beweis: Seien $f, g \in C[a, b]$ mit $\|f\|_2 = \|g\|_2 = 1$, $f \neq g$ gegeben. Dann gilt

$$\begin{aligned} \|f + g\|_2^2 \ &= \ \int_a^b |f(x) + g(x)|^2 dx = \int_a^b \big(f(x) + g(x)\big) \big(f(x) + g(x)\big) dx \\ &\leq \ \underbrace{\int_a^b |f(x)|^2 dx}_{=1} \ + \underbrace{\int_a^b |g(x)|^2 dx}_{=1} \ + 2 \int_a^b f(x) g(x) dx. \end{aligned}$$

Aus der Cauchy-Schwarzschen Ungleichung ergibt sich

$$2\int_a^b f(x)g(x)dx \leq 2\,\|f\|_2\,\|g\|_2 = 2.$$

Gleichheit gilt aber nur, falls f, g linear abhängig sind, d.h. $f = \lambda\, g$ für $\lambda \geq 0$ und ist daher nicht möglich.

Satz 2.5. *Die Lösung eines linearen Approximationsproblems ist bei streng konvexer Norm eindeutig bestimmt.*

Beweis. Es seien $h_0, \tilde{h}_0 \in U$ mit $h_0 \neq \tilde{h}_0$ zwei beste Approximationen an $f \in V$. Nach Satz 2.3 gilt dann auch

$$\|f - \frac{1}{2}\,(h_0 + \tilde{h}_0)\| = \frac{1}{2}\,\|(f - h_0) + (f - \tilde{h}_0)\| = E_U(f).$$

Division durch $E_U(f)$ ergibt wegen

$$\Big\|\frac{f - h_0}{E_U(f)}\Big\| = \Big\|\frac{f - \tilde{h}_0}{E_U(f)}\Big\| = 1$$

die Ungleichung

$$1 = \frac{1}{2}\Big\|\frac{f - h_0}{E_U(f)} + \frac{f - \tilde{h}_0}{E_U(f)}\Big\| \leq \frac{1}{2}\,\Big(\Big\|\frac{f - h_0}{E_U(f)}\Big\| + \Big\|\frac{f - \tilde{h}_0}{E_U(f)}\Big\|\Big) < 1$$

da die Norm streng konvex ist, und somit erhalten wir einen Widerspruch. □

Bemerkung: Die Umkehrung dieses Satzes gilt nicht!

2.2 Skalarprodukte und unitäre Vektorräume

Wir wiederholen einige wichtige Begriffe und Sätze aus der Linearen Algebra.

Definition 2.6. *Sei V ein Vektorraum über $K \in \{\mathbb{R}, \mathbb{C}\}$.*
Eine Abbildung $\langle .,. \rangle : V \times V \to K$ mit den folgenden Eigenschaften

(i) $\langle \alpha x + \beta y, z\rangle = \alpha\langle x, z\rangle + \beta\langle y, z\rangle \quad \forall\, \alpha, \beta \in K,\ x, y, z \in V,$
(d.h., $\langle .,. \rangle$ ist bilinear, sesquilinear)

(ii) $\langle x, y\rangle = \overline{\langle y, x\rangle} \quad \forall\, x, y \in V$ *(d.h., $\langle .,. \rangle$ ist symmetrisch bzw. hermitisch)*

(iii) $\langle x, x\rangle \geq 0 \quad \forall\, x \in V \setminus \{0\}$ *(d.h.,* $\langle ., .\rangle$ *ist positiv definit)*

heißt **Skalarprodukt** *(inneres Produkt) in* V*, und* V *heißt euklidisch (unitär).*

Beispiel: Sei $V = C[a, b]$ über $\mathbb{R}$. Dann ist

$$\langle f, g\rangle := \int_a^b f(x)g(x)dx$$

ein Skalarprodukt in V.

Satz 2.7. *(Cauchy-Schwarzsche Ungleichung)*
Sei V *ein unitärer Vektorraum. Für* $x, y \in V$ *gilt*

$$|\langle x, y\rangle|^2 \leq \langle x, x\rangle\langle y, y\rangle.$$

Gleichheit gilt genau dann, wenn $x = \lambda y \neq 0$ *mit* $\lambda \in K$ *oder* $x = 0$ *oder* $y = 0$.

Satz 2.8. *In einem unitären Vektorraum ist*

$$\|x\|_2 := \sqrt{\langle x, x\rangle}, \qquad x \in V$$

eine Norm.

Beispiel: In $V = C[a, b]$ mit dem Skalarprodukt $\langle f, g\rangle = \int_a^b f(x)g(x)dx$ ist

$$\|f\|_2 = (\int_a^b f(x)^2 dx)^{\frac{1}{2}}$$

eine Norm in V.

Satz 2.9. *Die durch ein Skalarprodukt induzierte Norm*

$$\|x\|_2 = (\langle x, x\rangle)^{\frac{1}{2}}, \qquad x \in V$$

ist streng konvex.

Beweis. Es gilt die Parallelogrammgleichung

$$\|x + y\|_2^2 + \|x - y\|_2^2 = 2(\|x\|_2^2 + \|y\|_2^2).$$

Für $\|x\|_2 = \|y\|_2 = 1$ und $x \neq y$ folgt daraus wegen $\|x - y\|_2^2 > 0$

$$\|x + y\|_2^2 < 2(\|x\|_2 + \|y\|_2) = 4.$$

□

Definition 2.10. *Zwei Elemente x, y eines unitären Vektorraums heißen* **orthogonal**, *falls $\langle x, y\rangle = 0$ gilt. Eine endliche oder abzählbare Menge von Elementen $\{x_1, x_2, \ldots, x_n, \ldots\}$ heißt* **orthonormal (Orthonormalsystem)**, *wenn*

$$\langle x_i, x_k\rangle = \delta_{ik} = \begin{cases} 1 & i = k, \\ 0 & i \neq k. \end{cases}$$

Satz 2.11. *(Schmidtsches Orthonormierungsverfahren)*
Aus jedem System linear unabhängiger Elemente $\{a_1, a_2, \ldots, a_n\}$ des unitären Vektorraums V lässt sich ein orthonormales System $\{u_1, u_2, \ldots, u_n\}$ gewinnen, und zwar rekursiv durch

$$\begin{aligned} u_1 &= \frac{a_1}{\|a_1\|}\,, \\ u_k &= \frac{a_k - \sum_{j=1}^{k-1}\langle a_k, u_j\rangle u_j}{\|a_k - \sum_{j=1}^{k-1}\langle a_k, u_j\rangle u_j\|} \qquad k = 2, 3, \ldots. \end{aligned}$$

Approximation in unitären Vektorräumen

Problem: Sei V ein unitärer Vektorraum und U_n ein Untervektorraum von V mit $\dim U_n = n$. Zu $f \in V$ finde man ein $h_0 \in U_n$ mit

$$\|f - h_0\|_2 \leq \|f - h\|_2 \qquad \forall\, h \in U_n.$$

Da die durch das Skalarprodukt induzierte Norm streng konvex ist, existiert zu jedem $f \in V$ genau eine beste Approximation $h_0 \in U_n$.

Satz 2.12. *Sei U_n ein Untervektorraum des unitären Vektorraums V und $\dim U_n = n$. Das Element $h_0 \in U_n$ ist genau dann beste Approximation eines Elements $f \in V$, wenn*

$$\langle f - h_0, h\rangle = 0 \quad \forall\, h \in U_n$$

gilt. Das Element h_0 heißt **orthogonale Projektion** *von f auf U_n.*
Ist $\{u_1, \ldots, u_n\}$ eine Orthonormalbasis von U_n, so hat h_0 die Form

$$h_0 = \sum_{j=1}^{n} \langle f, u_j\rangle\, u_j.$$

Beweis. 1. Existenz: Sei $\{u_1, \ldots, u_n\}$ eine Orthonormalbasis von U_n. Dann ist $\langle f - h_0, h\rangle = 0 \quad \forall\, h \in U_n$ genau dann erfüllt wenn

$$\langle f - h_0, u_j\rangle = 0 \qquad j = 1, \ldots, n,$$

d.h., wenn

$$\langle f, u_j\rangle = \langle h_0, u_j\rangle \qquad j = 1, \ldots, n.$$

Wegen $h_0 \in U_n$, existieren Konstanten $a_1, a_2, \ldots, a_n \in \mathbb{C}$ mit

$$h_0 = a_1 u_1 + a_2 u_2 + \ldots + a_n u_n.$$

Also folgt

$$\langle h_0, u_j \rangle = \langle a_1 u_1 + \ldots + a_n u_n, u_j \rangle = a_j.$$

Das Element $h_0 \in U$ mit

$$h_0 = \sum_{j=1}^{n} \langle f, u_j \rangle \, u_j$$

erfüllt also die Bedingung

$$\langle f - h_0, h \rangle = 0 \qquad \forall \, h \in U_n.$$

2. Außerdem gilt für alle $h \in U_n$

$$\begin{aligned}
\|f - h\|_2^2 &= \|f - h_0 + h_0 - h\|_2^2 \\
&= \langle (f - h_0) + (h_0 - h), (f - h_0) + (h_0 - h) \rangle \\
&= \|f - h_0\|_2^2 + \underbrace{\langle f - h_0, \underbrace{h_0 - h}_{\in U_n} \rangle}_{=0} + \underbrace{\overline{\langle f - h_0, \underbrace{h_0 - h}_{\in U_n} \rangle}}_{=0} + \|h_0 - h\|_2^2 \\
&= \|f - h_0\|_2 + \underbrace{\|h_0 - h\|^2}_{\geq 0} \\
&\geq \|f - h_0\|_2^2.
\end{aligned}$$

Gleichheit gilt genau dann, wenn $h = h_0$. □

2.3 Fourierreihen

Sei $f : \mathbb{R} \to \mathbb{R}$ eine 2π-periodische, stückweise stetige Funktion. Wir wollen f in $\mathcal{T}_n := \text{span}\{1, \cos x, \sin x, \ldots, \cos nx, \sin nx\}$ approximieren.
Sei $p \in \mathcal{T}_n$. Betrachte die Norm

$$\|f\|_2 := (\int_0^{2\pi} |f(x)|^2 dx)^{\frac{1}{2}} \quad (L^2\text{−Norm}).$$

Gesucht ist $p_0 \in \mathcal{T}_n$, so dass der Approximationsfehler

$$\|p_0 - f\|_2 := \left(\int_0^{2\pi} (p_0(x) - f(x))^2 dx \right)^{\frac{1}{2}}$$

minimal wird. Wir betrachten das Skalarprodukt in $L^2([0,2\pi))$,

$$\langle f,g\rangle := \int_0^{2\pi} f(x)\overline{g(x)}dx \qquad f,g \in L^2([0,2\pi)).$$

Satz 2.13. *Die trigonometrischen Funktionen* $\{1,\cos x,\ldots,\cos nx,\sin nx\}$ *bilden in* $[0,2\pi]$ *ein orthogonales System von* $\mathcal{T}_n$*. Es gilt*

$$\begin{aligned}
\int_0^{2\pi}\cos(jx)\cos(kx)dx &= \begin{cases} 0 & j\neq k,\\ 2\pi & j=k=0,\\ \pi & j=k\neq 0,\end{cases} \qquad \forall\, j,k\in\mathbb{N}_0,\\
\int_0^{2\pi}\sin(jx)\sin(kx)dx &= \begin{cases} 0 & j\neq k,\\ \pi & j=k>0,\end{cases} \qquad \forall\, j,k\in\mathbb{N},\\
\int_0^{2\pi}\sin(jx)\cos(kx)dx &= 0 \qquad \forall\, j\in\mathbb{N},\ k\in\mathbb{N}_0.
\end{aligned}$$

Beweis. Es gilt

$$I_1 := \int_0^{2\pi}\cos(jx)\cos(kx)dx = \frac{1}{2}\int_0^{2\pi}[\cos(j+k)x+\cos(j-k)x]\,dx.$$

Für $j\neq k$ folgt

$$I_1 = \frac{1}{2}\left[\frac{1}{j+k}\sin(j+k)x+\frac{1}{(j-k)}\sin(j-k)x\right]_0^{2\pi} = 0.$$

Für $j=k>0$ folgt

$$I_1 = \underbrace{\frac{1}{2}\left[\frac{1}{2j}\sin(2jx)\right]_0^{2\pi}}_{0} + \frac{1}{2}\int_0^{2\pi}\cos\,0\,dx = \pi.$$

Für $j=k=0$ folgt

$$I_1 = \frac{1}{2}\int_0^{2\pi}\underbrace{\cos 0+\cos 0}_{2}\,dx = 2\pi.$$

Die anderen Identitäten folgen analog. □

Wir wenden nun Satz 2.12 auf den Vektorraum $L^2([0,2\pi))$ und den Untervektorraum $\mathcal{T}_n$ der trigonometrischen Polynome an.

Satz 2.14. *Das trigonometrische Polynom p_0 ist beste Approximation von $f \in L^2([0,2\pi))$ in $\mathcal{T}_n$, wenn*

$$p_0(x) = \frac{a_0}{2} + \sum_{k=1}^{n} (a_k \cos(kx) + b_k \sin(kx))$$

mit

$$\begin{aligned} a_k(f) &= a_k = \frac{1}{\pi}\int_0^{2\pi} f(x)\cos(kx)dx \quad (k = 0,\ldots,n), \\ b_k(f) &= b_k = \frac{1}{\pi}\int_0^{2\pi} f(x)\sin(kx)dx \quad (k = 1,\ldots,n). \end{aligned}$$

Die Koeffizienten a_k, b_k heißen **Fourierkoeffizienten** *von f.*

Beweis. Wegen Satz 2.13 ist das System

$$\{\ \frac{1}{\sqrt{2\pi}}, \frac{1}{\sqrt{\pi}}\cos x, \frac{1}{\sqrt{\pi}}\sin x, \ldots, \frac{1}{\sqrt{\pi}}\cos(nx), \frac{1}{\sqrt{\pi}}\sin(nx)\}$$

orthonormal. Nach Satz 2.12 hat die orthogonale Projektion von f auf $\mathcal{T}_n$ die Form

$$\begin{aligned} p_0(x) &= \langle f, \frac{1}{\sqrt{2\pi}}\rangle \frac{1}{\sqrt{2\pi}} + \sum_{j=1}^{n} \Big(\langle f, \frac{1}{\sqrt{\pi}}\ \cos(jx)\rangle \frac{1}{\sqrt{\pi}}\ \cos(jx) \\ &\quad + \langle f, \frac{1}{\sqrt{\pi}}\ \sin(jx)\rangle \frac{1}{\sqrt{\pi}}\ \sin(jx) \Big), \end{aligned}$$

wobei

$$\begin{aligned} \frac{1}{2\pi}\ \langle f, 1\rangle &= \frac{1}{2\pi}\ \int_0^{2\pi} f(x)\,dx = \frac{a_0}{2}, \\ \frac{1}{\pi}\langle f, \cos(jx)\rangle &= \frac{1}{\pi}\ \int_0^{2\pi} f(x)\cos(jx)\,dx = a_j \qquad j = 1,\ldots,n, \\ \frac{1}{\pi}\langle f, \sin(jx)\rangle &= \frac{1}{\pi}\ \int_0^{2\pi} f(x)\sin(jx)\,dx = b_j \qquad j = 1,\ldots,n. \end{aligned}$$

□

Numerische Berechnung der Fourierkoeffizienten

Die komplexe Form des Fourier-Polynoms ist gegeben durch

$$p_0(x) = \sum_{k=-n}^{n} \underbrace{c_k(f)}_{=:c_k}\ e^{ixk}$$

mit

$$\begin{aligned} c_k(f) &:= \frac{1}{2}\,(a_k - ib_k) = \frac{1}{2\pi}\int_0^{2\pi} f(x)\underbrace{(\cos(kx) - i\sin(kx))}_{e^{-ikx}}\,dx \quad k = 1,\ldots,n, \\ c_{-k}\,(f) &:= \overline{c_k(f)} = \frac{1}{2\pi}\int_0^{2\pi} f(x)e^{ikx}\,dx \quad k = 1,\ldots,n, \\ c_0(f) &:= \frac{a_0}{2} = \frac{1}{2\pi}\int_0^{2\pi} f(x)\,dx. \end{aligned}$$

Also erhalten wir

$$c_k(f) = \frac{1}{2\pi}\int_0^{2\pi} f(x)e^{-ikx}\,dx = \langle f, e^{ik\cdot}\rangle, \quad k = -n,\ldots,n.$$

" Grobe" numerische Berechnung von $c_k(f)$: Wir zerlegen das Intervall $[0, 2\pi]$ in Teilintervalle der Länge $\frac{2\pi}{M}$ und wenden zur näherungsweisen Berechnung des Integrals die Rechteckregel an. Wir erhalten mit $w_M := e^{-2\pi i/M}$

$$\begin{aligned} c_k^* &= \frac{1}{2\pi}\,\frac{2\pi}{M}\sum_{j=0}^{M-1} f(\frac{2\pi j}{M})\,e^{-ik\cdot\frac{2\pi j}{M}} \\ &= \frac{1}{M}\sum_{j=0}^{M-1} f(\frac{2\pi j}{M})w_M^{jk}. \end{aligned}$$

Die Koeffizienten c_k^* lassen sich also mit Hilfe einer DFT(M) berechnen.

Beachte: M und n stehen nicht in Zusammenhang, aber c_k^* ist eine M-periodische Folge!

Die Folge der Fourierkoeffizienten $c_k = c_k(f)$ ist jedoch im Allgemeinen eine Nullfolge. Es gilt:

Satz 2.15. *Die Funktion $f \in C([0, 2\pi])$ besitze eine absolut konvergente Fourierreihe der Form*

$$f(x) = \sum_{j=-\infty}^{\infty} c_j(f)e^{ijx} = \frac{a_0}{2} + \sum_{j=1}^{\infty}\Big(a_j(\cos(jx)) + b_j(\sin(jx))\Big).$$

(Dies ist z.B. für $f \in C^1([0, 2\pi))$ der Fall). Dann gilt die **Aliasing-Formel**

$$c_k^* = \sum_{l=-\infty}^{\infty} c_{k+lM}(f) \qquad (k \in \mathbb{Z}).$$

Beweis. Aus $f(x) = \sum_{l=-\infty}^{\infty} c_l(f) e^{ilx}$ folgt für $x = x_j = \frac{2\pi j}{M}$, $j = 0, \ldots, M-1$,

$$f(x_j) = \sum_{l=-\infty}^{\infty} c_l(f) w_M^{-lj} \ .$$

Wegen

$$c_k^* = \frac{1}{M} \sum_{j=0}^{M-1} f(x_j)\, w_M^{jk}, \qquad k = 0, \ldots, M-1,$$

ergibt sich nach Lemma 1.29

$$\begin{aligned} c_k^* &= \frac{1}{M} \sum_{j=0}^{M-1} \Big(\sum_{l=-\infty}^{\infty} c_l(f)\, w_M^{-lj} \Big) \\ &= \sum_{l=-\infty}^{\infty} c_l(f)\, \frac{1}{M} \underbrace{\sum_{j=0}^{M-1} w_M^{jk}\, w_M^{-lj}}_{\begin{cases} 0 & k \not\equiv l \bmod M \\ M & k = l \bmod M \end{cases}} = \sum_{r=-\infty}^{\infty} c_{k+rM}\ (f). \end{aligned}$$

□

Nach der Aliasing-Formel gilt also für alle $k \in \mathbb{Z}$

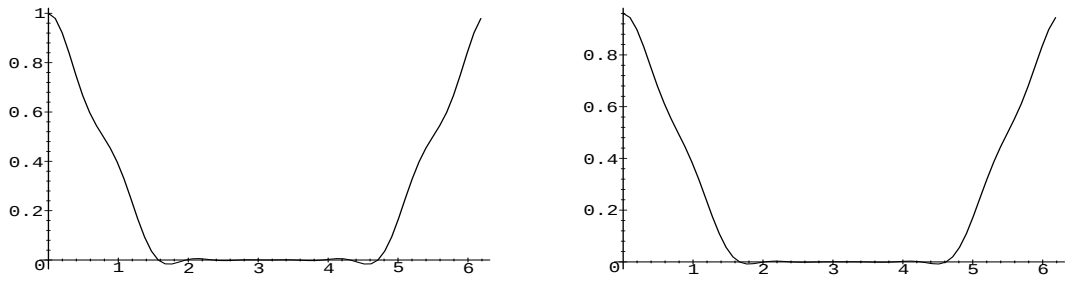
$$c_k^* - c_k(f) = \sum_{l=1}^{\infty} (c_{k+lM}\ (f) + c_{k-lM}(f)).$$

Für $|k| > \frac{M}{2}$ kann der absolute Fehler $|c_k^* - c_k(f)|$ also größer als $|c_k(f)|$ werden! Damit c_k^* eine gute Näherung von $c_k(f)$ ist, muss M also im Vergleich zu n möglichst groß gewählt werden, mindestens $M > 2n$.

Algorithmus:
Eingabe: Wähle $M > 2n$, $M = 2^t$, $L = 2^{t'}$.

1. Berechne
$$c_k^* = \frac{1}{M} \sum_{j=0}^{M-1} f(\frac{2\pi j}{M}) w_M^{jk}$$
mittels eines FFT-Algorithmus (siehe Abschnitt 1.5).

2. Werte das Polynom $p_0(x)$ an den äquidistanten Stützstellen $\frac{2\pi j}{L}$, $j = 0, \ldots, L-1$, mittels eines FFT-Algorithmus aus.
$$p_0(\frac{2\pi j}{L}) = \sum_{k=-n}^{n} c_k^* \underbrace{e^{\frac{2\pi i k j}{L}}}_{w_L^{-jk}} = \sum_{k=0}^{L-1} \tilde{c}_k w_L^{-jk}$$



Beste Approximation aus $\mathcal{T}_8$, berechnet mit obigem Algorithmus (mit $M = 16$ (links) und $M = 32$ (rechts)).

mit $\tilde{\mathbf{c}} = (c_0^*, c_1^*, \ldots, c_n^*, 0, \ldots, 0, c_{-n}^*, \ldots, c_{-1}^*)^T \in \mathbb{C}^L$.

Ausgabe: $p_0(\frac{2\pi j}{L})$, $j = 0, \ldots, L-1$.

Beispiel: Wir approximieren die 2π-periodische Funktion

$$f(t) = \begin{cases} 1 - \frac{2}{\pi}t & 0 \le t \le \frac{\pi}{2}, \\ 0 & \frac{\pi}{2} < t \le \frac{3\pi}{2}, \\ -3 + \frac{2}{\pi}t & \frac{3\pi}{2} < t < 2\pi. \end{cases}$$

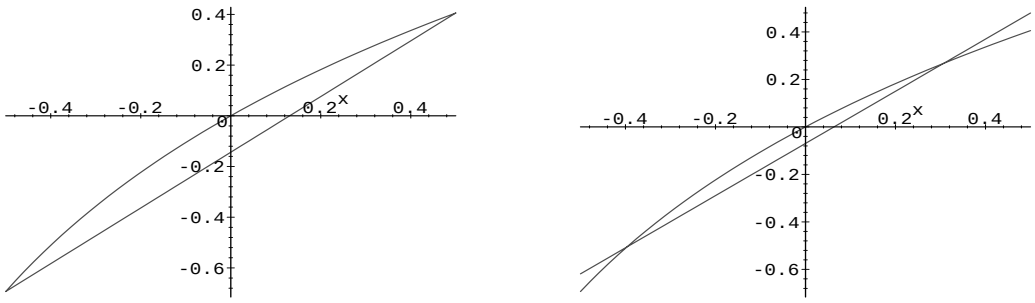
durch ihre beste Approximation in $\mathcal{T}_8$, d.h., durch ihr Fourierpolynom $p_0(t) \in \mathcal{T}_8$ und wenden zur Berechnung des Fourierpolynoms den obigen Algorithmus an. Wir verwenden zur Berechnung der 17 Fourierkoeffizienten zunächst $M = 16$ (links) und dann $M = 32$ (rechts). Für $M = 16$ ist die Berechnung nach obigen Überlegungen noch sehr ungenau. Wir erhalten die Fourierkoeffizienten

	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
$M = 16$	0.25	0.2053	0.1067	0.0253	0	0.0113	0.01831	0.0081	0
$M = 32$	0.25	0.2033	0.1026	0.0232	0	0.0088	0.01266	0.0049	0
exakt	0.25	0.2026	0.1013	0.0225	0	0.0081	0.01126	0.0041	0

2.4 Gleichmäßige Approximation

In diesem Abschnitt sei nun speziell $V = C[a, b]$ mit der Tschebyscheff-Norm

$$\|f\|_\infty := \max_{x \in [a,b]} |f(x)| \qquad (\text{Maximum} - \text{Norm}).$$



Interpolierende Gerade $q_1(x)$ an $f(x) = \ln(1+x)$ (links), und beste Approximation durch $p_1^* \in \Pi_1$.

Wir suchen nun ein Polynom $p_n^* \in \Pi_n$ (höchstens $n-$ten Grades), das den Abstand $f - p_n$ in der Maximum-Norm minimiert

$$\|f - p_n\|_\infty = \max_{x\in[a,b]} |f(x) - p_n(x)| \ \rightarrow \text{Min!}$$

Dann heißt p_n^* Polynom bester gleichmäßiger Approximation (bester Approximation im Tschebyscheffschen Sinne).

Beispiel 1: Die Funktion $f(x) = \ln(1 + x)$ soll in $[-\frac{1}{2}, \frac{1}{2}]$ durch $p_1^* \in \Pi_1$ (eine Gerade) gleichmäßig approximiert werden. Das Polynom p_1^* habe die Form $p_1^*(x) = c_0 + c_1\, x$. Betrachte die Differenz $\delta(x) = p_1^*(x) - \ln(1 + x)$.

Wir betrachten zunächst die interpolierende Gerade $q_1(x)$ mit $q_1(\frac{1}{2}) = f(\frac{1}{2})$ und $q_1(-\frac{1}{2}) = f(-\frac{1}{2})$. Dann entsteht der größte Abstand für $x \approx 0$. Eine Verschiebung der Geraden "nach oben" bewirkt eine Verkleinerung des Abstandes in 0 und eine Vergrößerung in $-\frac{1}{2}, \frac{1}{2}$. Also hat $|\delta(x)|$ drei Extremwerte, an $x_0 = -\frac{1}{2}$, an $x_2 = \frac{1}{2}$ und an einer Zwischenstelle $x_1 \in (-\frac{1}{2}, \frac{1}{2})$.
Wir berechnen zunächst x_1 folgendermaßen:

$$\delta'(x) = c_1 - \frac{1}{1+x} = 0 \qquad \Leftrightarrow \quad x_1 = \frac{1}{c_1} - 1.$$

Für p_1^* muss also gelten (siehe Abbildung):

$$\begin{aligned}
\delta(x_0) &= c_0 + c_1 x_0 - \ln(1 + x_0) = \Delta, \\
\delta(x_1) &= c_0 + c_1 x_1 - \ln(1 + x_1) = -\Delta, \\
\delta(x_2) &= c_0 + c_1 x_2 - \ln(1 + x_2) = \Delta.
\end{aligned}$$

Aus

$$\begin{aligned} c_0 - \frac{1}{2}\, c_1 - \ln(\frac{1}{2}) &= \Delta, \\ c_0 + \frac{1}{2}\, c_1 - \ln(\frac{3}{2}) &= \Delta, \end{aligned}$$

folgt $c_1 = \ln(\frac{3}{2}) - \ln(\frac{1}{2}) = 1.0986$ und damit $x_1 = \frac{1}{c_1} - 1 = -0.08976$. Aus

$$\begin{aligned} c_0 - \frac{1}{2}\, c_1 - \ln(\frac{1}{2}) &= \Delta \\ c_0 + x_1\; c_1 - \ln(1 + x_1) &= -\Delta \end{aligned}$$

folgt $c_0 = \frac{1}{2}(\ln(\frac{1}{2}) + \ln(1+x_1) + c_1(\frac{1}{2} - x_1)) = -0.06964$ und $\Delta = c_0 - \frac{1}{2}c_1 - \ln(\frac{1}{2}) = 0.07420$. Wir erhalten die Gerade bester Approximation

$$p_1^*(x) = -0.06964 + 1.0986\, x.$$

Beispiel 2: Die Funktion $f(x) \equiv 0$ soll in $[-1, 1]$ durch ein Polynom $p_{n+1}^* \in \Pi_{n+1}$ mit Höchstkoeffizient 1 gleichmäßig approximiert werden.
Nach Satz 1.19 war $2^{-n} T_{n+1}(x)$ ein Polynom mit der Eigenschaft

$$2^{-n} = \max_{x \in [-1,1]} \frac{|T_{n+1}(x)|}{2^n} \leq \max_{x \in [-1,1]} |w_{n+1}(x)|$$

für jedes Polynom $w_{n+1}(x) \in \Pi_{n+1}$ mit Höchstkoeffizient 1.

Beispiel 3: Das Monom $f(x) = x^{n+1}$ soll durch $p_n^* \in \Pi_n$ in $[-1, 1]$ gleichmäßig approximiert werden. Betrachte

$$\delta(x) = p_n^*(x) - f(x) = p_n^*(x) - x^{n+1}\ .$$

Dann ist $\delta(x)$ ein Polynom von Grad $n+1$ mit Höchstkoeffizient (-1). Folglich ist $|\delta(x)|$ nach Beispiel 2 minimal, wenn

$$-\delta(x) = x^{n+1} - p_n^*(x) = \frac{1}{2^n}\, T_{n+1}(x)$$

ein Tschebyscheffpolynom vom Grad $n+1$ ist, d.h.,

$$p_n^*(x) = x^{n+1} - \frac{1}{2^n}\, T_{n+1}(x).$$

Wir erhalten:

n	x^{n+1}	$T_{n+1}(x)$	$p_n^*(x)$	Δ
1	x^2	$2x^2-1$	$\frac{1}{2}$	$\frac{1}{2}$
2	x^3	$4x^3-3x$	$\frac{3}{4}x$	$\frac{1}{4}$
3	x^4	$8x^4-8x^2+1$	$x^2-\frac{1}{8}$	$\frac{1}{8}$
4	x^5	$16x^5-20x^3+5x$	$\frac{5}{4}\ x^3-\frac{5}{16}\ x$	$\frac{1}{16}$

Satz 2.16. *(Alternantensatz)*
Existiert zu einer Funktion $f \in C[a,b]$ und einem Polynom $p_n \in \Pi_n$ eine Folge (Referenz) von $n+2$ Punkten

$$a \le x_0 < x_1 < \ldots < x_{n+1} \ \le b,$$

in denen der Approximationsfehler $\delta(x) = p_n(x) - f(x)$ seinen Maximalwert

$$\Delta_n(f) := \|p_n - f\|_\infty$$

mit alternierenden Vorzeichen annimmt,

$$\begin{aligned} |\delta(x_k)| &= |p_n(x_k) - f(x_k)| = \Delta_n(f), \qquad k = 0,\ldots,n+1, \\ \delta(x_k) &= -\delta(x_{k+1}), \qquad k = 0,\ldots,n, \end{aligned}$$

so ist p_n ein Polynom bester gleichmäßiger Approximation (also $p_n = p_n^*$) an die Funktion f und $\Delta_n(f)$ gleich dem Abstand der Funktion f von Π_n.

Beweis. Angenommen, es existiert ein $p_n^* \in \Pi_n$ mit kleinerem Approximationsfehler, d.h.

$$\delta^*(x) = p_n^*(x) - f(x), \qquad \|\delta^*\|_\infty := \Delta^* < \Delta.$$

Betrachte nun $q_n(x) := p_n(x) - p_n^*(x)$. Dann wechselt $q_n(x)$ mindestens $n+1$ mal das Vorzeichen in $[a,b]$ denn

$$\operatorname{sign}(q_n(x_k)) = \operatorname{sign}(p_n(x_k)) \quad k = 0,\ldots,n+1.$$

Also hat $q_n \in \Pi_n$ mindestens $n+1$ Nullstellen und ist daher das Nullpolynom im Widerspruch zur Annahme. □

Die in Satz 2.15 definierte Folge $x_0 < \ldots < x_{n+1}$ heißt **Alternante**.

Bemerkung: Man kann zeigen, dass zu jeder Funktion $f \in C[a,b]$ ein eindeutig bestimmtes Polynom gleichmäßiger Approximation existiert.

Der Remez-Algorithmus

Wir wollen zu festgelegtem $f \in C[a,b]$ ein Polynom gleichmäßiger Approximation numerisch bestimmen.

Idee: Wir suchen zunächst ein Polynom $p_n \in \Pi_n$ mit

$$p_n(x_k) - f(x_k) = (-1)^k h \qquad k = 0, \ldots, n+1 \tag{2.2}$$

mit einer gewissen Startfolge (Startreferenz) $a \le x_0 < x_1 < \ldots < x_{n+1} \le b$.

Wir ändern nun die Stützstellen schrittweise ab, so dass die Folge der zugehörigen $|h|$ monoton zunimmt und gegen $\Delta = \Delta_n(f)$ konvergiert. Erfüllt p_n die Bedingung (2.2), so nennen wir p_n das Referenzpolynom und $|h|$ seine Referenzabweichung.

Satz 2.17. *(Existenz u. Eindeutigkeit des Referenzpolynoms)*
Für jede Referenz $a \le x_0 < x_1 \ldots < x_{n+1} \le b$ und jede Funktion $f \in C[a,b]$ gibt es genau ein Polynom $p_n(x)$ und genau eine Zahl h, die (2.2) erfüllen.

Beweis. 1) Existenz: Nach Satz 1.2 ist durch $n+2$ Stützstellen $x_0 < \ldots < x_{n+1}$ und passende Stützwerte $y_0, \ldots, y_{n+1}$ ein Interpolationspolynom $p_{n+1} \in \Pi_{n+1}$ mit $p_{n+1}(x_k) = y_k$, $k = 0, \ldots, n+1$ eindeutig festgelegt.
Wir betrachten die zwei Interpolationspolynome zu den Stützwerten $y_k = (-1)^k$, $k = 0, \ldots, n+1$ und zu $y_k = f(x_k)$, $k = 0, \ldots, n+1$ in Lagrangedarstellung,

$$\begin{aligned} q_{n+1}(x) &= \sum_{k=0}^{n+1} (-1)^k l_k^{n+1}(x) \\ &= \sum_{j=0}^{n+1} a_j x^j \qquad (\text{mit } \; l_k^{n+1}(x) = \prod_{\substack{j=0 \\ j\neq k}}^{n+1} (\frac{x - x_j}{x_k - x_j})), \end{aligned} \tag{2.3}$$

$$r_{n+1}(x) = \sum_{k=0}^{n+1} f(x_k) l_k^{n+1}(x) = \sum_{j=0}^{n+1} b_j x^j. \tag{2.4}$$

Bilde nun

$$p_n(x) := r_{n+1}(x) - \frac{b_{n+1}}{a_{n+1}} \; q_{n+1}(x).$$

Dann ist $p_n(x) \in \Pi_n$, denn $r_{n+1}(x)$ und $\frac{b_{n+1}}{a_{n+1}} q_{n+1}(x)$ haben beide den Höchstkoeffizienten b_{n+1}. (Beachte, dass $a_{n+1} \neq 0$, denn wegen der $n+2$ alternierenden Stützwerte hat das Polynom q_{n+1} genau $n+1$ Nullstellen.)
Weiter gilt

$$p_n(x_k) = r_{n+1}(x_k) - \frac{b_{n+1}}{a_{n+1}} \; q_{n+1}(x_k) = f(x_k) - \frac{b_{n+1}}{a_{n+1}} \; (-1)^k.$$

Setze nun $h := -\frac{b_{n+1}}{a_{n+1}}$. Dann erfüllt p_n die Bedingungen (2.2).

2) Eindeutigkeit: Angenommen, es existiert ein $\overline{p_n} \in \Pi_n$ mit $\overline{p_n}(x_k) - f(x_k) = (-1)^k\overline{h}$, $k = 0, \ldots, n+1$ und $\overline{p_n} \neq p_n$. Dann erhalten wir

$$\overline{p_n}\,(x_k) - p_n(x_k) = (f(x_k) + (-1)^k\overline{h}) - (f(x_k) + (-1)^k h).$$

Falls $\overline{h} = h$ ist, folgt $\overline{p_n} = p_n$ wegen Satz 1.2. Falls $\overline{h} \neq h$ ist, hat $\overline{p_n} - p_n$ mindestens $n+2$ Vorzeichenwechsel und daher mindestens $n+1$ Nullstellen. Also ist $\overline{p_n} - p_n$ ein Nullpolynom im Widerspruch zur Annahme. □

Explizite Berechnung von h

Wenn h schon durch die Referenz $\{x_k\}_{k=0}^{n+1}$ und $f(x)$ bestimmt ist, wäre eine explizite Darstellung hilfreich, da dann (2.2) in ein Interpolationsproblem übergeht. Im Beweis von Satz 2.17 hatten wir

$$h := -\frac{b_{n+1}}{a_{n+1}}$$

erhalten, wobei a_{n+1} und b_{n+1} die Höchstkoeffizienten der Interpolationspolynome q_{n+1} und r_{n+1} waren.

Wir betrachten den Höchstkoeffizient von $l_k^{n+1}(x) = \prod\limits_{\substack{j=0\\ j\neq k}}^{n+1} \frac{(x-x_j)}{(x_k-x_j)}$ und finden für $k = 0, \ldots, n$

$$d_k := \prod_{\substack{j=0\\ j\neq k}}^{n+1} \frac{1}{x_k - x_j} = (-1)^{n+1-k}|d_k|$$

da $x_0 < x_1 < \ldots < x_{n+1}$. Aus (2.3) und (2.4) ergibt sich damit

$$a_{n+1} = \sum_{k=0}^{n+1}(-1)^k\,(-1)^{n+1-k}|d_k|, \qquad b_{n+1} = \sum_{k=0}^{n+1} f(x_k)\,(-1)^{n+1-k}|d_k|,$$

so dass wir

$$\begin{aligned} h &= -\frac{\sum_{k=0}^{n+1} f(x_k)\,(-1)^{n+1-k}|d_k|}{\sum_{k=0}^{n+1}(-1)^k(-1)^{n+1-k}|d_k|} = -\frac{\sum_{k=0}^{n+1} f(x_k)(-1)^k|d_k|}{\sum_{k=0}^{n+1}|d_k|} \\ &= -\sum_{k=0}^{n+1} f(x_k)(-1)^k\,\lambda_k \quad \text{mit} \quad \lambda_k := \frac{|d_k|}{\sum_{j=0}^{n+1}|d_j|} \end{aligned} \tag{2.5}$$

erhalten. Beachte, dass $\sum_{k=0}^{n+1} \lambda_k = 1$.

Wir leiten noch eine weitere Darstellung für h her. Stellt man ein beliebiges $p \in \Pi_n$ als Interpolationspolynom $(n+1)-$ten Grades dar,

$$p(x) = \sum_{k=0}^{n+1} p(x_k)\, l_k^{n+1}(x) = \sum_{j=0}^{n+1} c_j x^j,$$

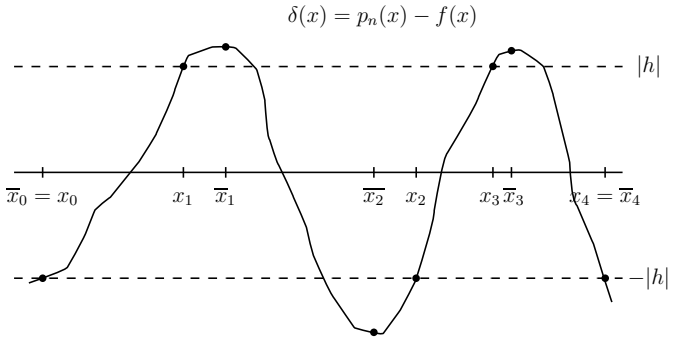
so ergibt sich für c_{n+1}

$$c_{n+1} \ = 0 = \sum_{k=0}^{n+1} p(x_k)\,(-1)^{n+1-k}|d_k| = (-1)^{n+1} \sum_{k=0}^{n+1} p(x_k)(-1)^k |d_k|.$$

Also erhalten wir mit (2.5)

$$h = \sum_{k=0}^{n+1} \lambda_k (-1)^k \,(p(x_k) - f(x_k)). \tag{2.6}$$

Jetzt kann das Referenzpolynom $p_n(x)$ mittels Polynom-Interpolation von beliebigen $n+1$ der $n+2$ Stützstellen/Stützwerte $(x_k, f(x_k) + (-1)^k h)$ berechnet werden. Anschließend muss der Verlauf von $\delta(x) = p_n(x) - f(x)$ untersucht werden. Gilt $|\delta(x)| \ \leq |h|$ für alle $x \in [a,b]$, so ist $\{x_k\}_{k=0}^{n+1}$ eine Alternante und $p_n(x)$ das gesuchte Polynom bester Approximation.



Gilt in einigen Teilintervallen $|\delta(x)| > |h|$, so muss die Referenz geändert werden. Ist im Intervall (x_k, x_{k+1}) $\max|\delta(x)| > |h|$, so wähle einen Punkt $\overline{x} \in (x_k, x_{k+1})$, so dass $|\delta(\overline{x})| \ > |h|$, und tausche $\overline{x}$ gegen x_k aus, falls $\delta(x_k)$ und $\delta(\overline{x})$ das gleiche Vorzeichen haben, ansonsten gegen x_{k+1} (siehe Skizze).
So erhalten wir eine neue Referenz $\overline{x}_0 < \ldots < \overline{x}_{n+1}$.

Satz 2.18. *Tauscht man die Referenz $a \leq x_0 < \ldots < x_{n+1} \leq b$ gegen eine Referenz $a \leq \overline{x}_0 < \ldots < \overline{x}_{n+1} \leq b$ aus, so dass*

$$|\delta(\overline{x}_k)| \geq |\delta(x_k)|, \qquad k = 0, \ldots, n+1,$$

und

$$\text{sgn } \delta(\overline{x}_k) = -\text{sgn } \delta(\overline{x}_{k+1}\ ), \qquad k = 0, \ldots, n,$$

und für mindestens ein j

$$|\delta(\overline{x}_j)| > |\delta(x_j)| = |h|$$

gilt, wobei δ *die zum alten Referenzpolynom* $p_n(x)$ *gehörende Fehlerfunktion* $\delta(x) = p_n(x) - f(x)$ *ist, so hat das neue Referenzpolynom* $\overline{p}_n(x)$ *eine größere Referenzabweichung* $|\overline{h}| > |h|$.

Beweis. Wir stellen $\overline{h}$ mit Hilfe von (2.6) dar und finden mit $p = p_n$ und der Referenz $\{\overline{x}_k\}_{k=0}^{n+1}$

$$|\overline{h}| \ = |\sum_{k=0}^{n+1} \overline{\lambda}_k (-1)^k (p_n(\overline{x}_k) - f(\overline{x}_k))| \ = \sum_{k=0}^{n+1} \ \overline{\lambda}_k |\delta(\overline{x}_k)|,$$

wobei wir ausgenutzt haben, dass $\delta(\overline{x}_k)$ alternierendes Vorzeichen besitzen. Andererseits gilt $|h| \ = |\delta(x_k)|$ für alle $k = 0, \ldots, n+1$ und damit

$$\sum_{k=0}^{n+1} \overline{\lambda}_k |\ \delta(x_k)| \ = |h| \ \sum_{k=0}^{n+1} \overline{\lambda}_k = |h|.$$

Da $|\delta(\overline{x}_j)| > |\delta(x_j)|$ für mindestens ein j, folgt $|\overline{h}| > |h|$. □

Mit der neuen Referenz $a \leq \overline{x}_0 < \ldots < \overline{x}_{n+1} \leq b$ wird der Prozess wiederholt, usw. Wir erhalten eine monoton wachsend Folge von Referenzabweichungen. Die Folge ist konvergent, denn es gilt

Satz 2.19. *(Beschränktheit der Referenzabweichung)*
Ist $p_n(x)$ *ein Referenzpolynom mit der Referenzabweichung* $|h|$ *und* $\delta(x)$ *die zugehörige Fehlerfunktion, so gilt* $|h| \ \leq \Delta_n(f) \leq \max_{x\in[a,b]} \ |\delta(x)|$ *mit* $\Delta_n(f)$ *aus Satz 2.16 .*

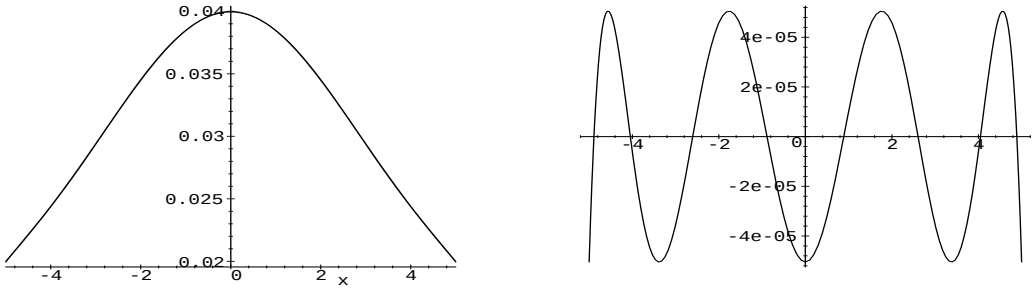
Beweis. Es war $\Delta_n(f) = \max_{x\in[a,b]} \ |(p_n^*(x) \ - f(x)|$, wobei p_n^* das (gesuchte) Polynom gleichmäßiger Approximation zu f ist. Folglich gilt

$$\Delta_n(f) \leq \max_{x\in[a,b]} \ \ |\delta(x)|.$$

Weiter gilt nach (2.6) mit $p = p_n^*$ für die Referenzabweichung $|h|$ des Polynoms $p_n(x)$

$$|h| \ = |\sum_{k=0}^{n+1} \lambda_k (-1)^k (p_n^*(x_k) - f(x_k))| \leq \underbrace{\max_{x\in[a,b]} |p_n^*(x) - f(x)|}_{\Delta_n(f)} \underbrace{(\sum_{k=0}^{n+1} \lambda_k)}_{=1} = \Delta_n(f).$$

□



Beste Approximation aus Π_7 an f, berechnet mit Remez-Algorithmus (links) und der entstehende Approximationsfehler (rechts).

Problem: Wie sollte die Startreferenz $a \leq x_0 < \ldots < x_{n+1} \leq b$ gewählt werden?

Wähle im Intervall $[a, b]$ zum Beispiel die Tschebyscheff-Knoten

$$x_k = \frac{a+b}{2} + (\frac{a-b}{2}) \cos \frac{k\pi}{n+1} \qquad k = 0, \ldots, n+1.$$

Algorithmus: siehe z.B. G. Maeß: Vorlesungen über numerische Mathematik, Band II: Analysis, Birkhäuser, Basel, 2. Aufl., 1988.

Beachte: Das Maple-Programm findet keine sinnvolle Lösung, wenn sich für die Startreferenz die Referenzabweichung 0 ergibt.

Beispiel: Wir wollen eine gleichmäßige Approximation an die Funktion $f(x) = \frac{1}{x^2+25}$ auf dem Intervall $[-5, 5]$ finden (Rungebeispiel). Wir suchen das Polynom bester Approximation an $f(x)$ aus Π_7.
Als Startreferenz verwenden wir die Tschebyscheff-Knoten

$$[-5.0, -4.61940, -3.53553, -1.91342, 0.0, 1.91342, 3.53553, 4.61940, 5.0]).$$

Nach 4 Iterationen mit dem Algorithmus erhalten wir die neue Referenz

$$[-5.0, -4.58094, -3.39400, -1.73059, 0.0, 1.73059, 3.39400, 4.58094, 5.0])$$

und $h = 0.00005$.

3 CAGD

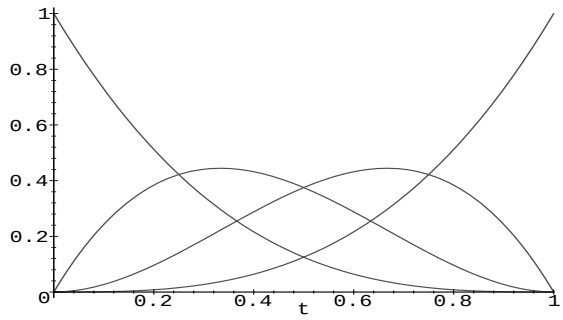
CAGD (Computer Aided Geometric Design) beschäftigt sich mit mathematischen und numerischen Methoden zur Beschreibung geometrischer Objekte, die in den Bereichen von CAD/CAM bis zur Robotik und in der wissenschaftlichen Visualisierung vorkommen. Die Hauptobjekte des mathematischen Interesses sind Kurven und Flächen, die durch Splines erzeugt werden.

3.1 Bernstein-Polynome

Definition 3.1. *Die Polynome*

$$b_k^n(t) := \binom{n}{k} t^k (1-t)^{n-k} \quad k = 0, \ldots, n, \quad n \in \mathbb{N}_0$$

heißen **Bernstein-Polynome** *vom Grad n bezüglich des Intervalls $[0,1]$. Ferner sei $b_k^n \equiv 0$ für $k < 0, k > n$.*



Bernstein-Polynome $b_k^3(t)$ für $k = 0, 1, 2, 3$.

Satz 3.2. *(Eigenschaften der Bernstein-Polynome)*
Für $n \in \mathbb{N}_0,\ 0 \leq k \leq n$ gilt

(i) *Das Polynom $b_k^n(t)$ hat eine k-fache Nullstelle für $t = 0$.*

(ii) *Das Polynom $b_k^n(t)$ hat eine $(n-k)$-fache Nullstelle für $t = 1$.*

(iii) *Es gilt $b_k^n(t) > 0$ für $0 < t < 1$ und $b_k^n(t)$ hat genau ein Maximum in $[0,1]$, nämlich an der Stelle $t = \frac{k}{n}$.*

Beweis. Die Eigenschaften (i), (ii) folgen aus der Definition.
(iii) Es gilt $b_k^n(t) > 0$ für $t \in (0,1)$ nach Definition. Insbesondere hat b_k^n keine Nullstelle in $(0,1)$ da bereits alle Nullstellen entweder in 0 oder in 1 liegen. Wir erhalten für die Ableitung

$$\begin{aligned}(b_k^n(t))' &= \binom{n}{k}\left(k\,t^{k-1}(1-t)^{n-k} - (n-k)\,t^k(1-t)^{n-k-1}\right)\\ &= \begin{cases} -n(1-t)^{n-1} & \text{für } k=0,\\ nt^{n-1} & \text{für } k=n,\\ \binom{n}{k}t^{k-1}(1-t)^{n-k-1}(k-nt) & \text{für } 0<k<n.\end{cases}\end{aligned}$$

Folglich ist $b_0^n(t)$ monoton fallend und besitzt ein Maximum in 0, $b_n^n(t)$ ist monoton wachsend und besitzt ein Maximum in 1. Für $b_k^n(t),\ 1 \le k \le n-1$ erhalten wir ein Maximum falls $k - nt = 0$, d.h., für $t = \frac{k}{n}$. □

Satz 3.3. *Für $t \in \mathbb{R}$ gilt*

(i) $\sum_{k=0}^{n} b_k^n(t) = 1, \qquad n \in \mathbb{N}_0,$

(ii) $\sum_{k=0}^{n} \frac{k}{n}\, b_k^n(t) = t, \qquad n \in \mathbb{N},\ n \ge 1,$

(iii) $\sum_{k=0}^{n} (\frac{k}{n})^2\, b_k^n(t) = \frac{n-1}{n}t^2 + \frac{t}{n}, \qquad n \in \mathbb{N},\ n \ge 2.$

Beweis. (i) Es gilt

$$(t + (1-t))^n = 1 = \sum_{k=0}^{n}\binom{n}{k}t^k(1-t)^{n-k} = \sum_{k=0}^{n} b_k^n(t).$$

(ii) Wegen $\frac{k}{n}\binom{n}{k} = \frac{(n-1)!}{(k-1)!(n-k)!} = \binom{n-1}{k-1}$ folgt mit $k' = k-1$

$$\begin{aligned}\sum_{k=0}^{n} \frac{k}{n}\, b_k^n(t) &= \sum_{k=0}^{n}\frac{k}{n}\binom{n}{k}t^k(1-t)^{n-k} = t\sum_{k=1}^{n}\binom{n-1}{k-1}t^{k-1}(1-t)^{n-k}\\ &= t\sum_{k'=0}^{n-1}\binom{n-1}{k'}t^{k'}(1-t)^{(n-1)-k'} = t\sum_{k=0}^{n-1} b_k^{n-1}(t) \overset{(i)}{=} t.\end{aligned}$$

(iii) Übungsaufgabe. □

Bemerkung: Die Bernstein-Polynome erfüllen die Rekursionsformel

$$b_k^n(t) = (1-t)\, b_k^{n-1}(t) + t\, b_{k-1}^{n-1}(t)$$

mit $b_n^{n-1}(t) := 0$ und $b_{-1}^{n-1}(t) := 0$, denn

$$\begin{aligned} & (1-t)\, b_k^{n-1}(t) + t\, b_{k-1}^{n-1}(t) \\ = \;& (1-t)\binom{n-1}{k} t^k (1-t)^{n-1-k} + t\binom{n-1}{k-1} t^{k-1}(1-t)^{n-k} \\ = \;& \left(\binom{n-1}{k} + \binom{n-1}{k-1}\right) t^k (1-t)^{n-k} = \binom{n}{k} t^k (1-t)^{n-k} = b_k^n(t). \end{aligned}$$

Dabei wurde die Rekursion $\binom{n-1}{k} + \binom{n-1}{k-1} = \binom{n}{k}$ ausgenutzt.

3.2 Bézier-Kurven

Wir verwenden Bernstein-Polynome zur Konstruktion von Kurven.

Definition 3.4. *Sind $\boldsymbol{\beta}_0, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_n \in \mathbb{R}^d$ gegeben, so ist das vektorwertige Polynom*

$$p(t) = \sum_{k=0}^{n} \boldsymbol{\beta}_k b_k^n(t)$$

eine polynomiale Kurve im $\mathbb{R}^d$ in **Bézier-Darstellung** *auf $[0,1]$. Die Koeffizienten $\boldsymbol{\beta}_0, \ldots, \boldsymbol{\beta}_n$ heißen* **Kontroll-** *oder* **Bézier-Punkte** *von p, der durch sie bestimmte Streckenzug heißt* **Bézier-Polygon**.

Eigenschaften von Bézier-Kurven

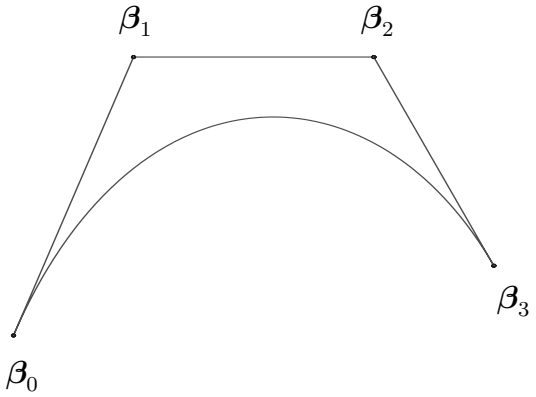
Satz 3.5. *Für $t \in [0,1]$ liegt der Graph des Bézier-Polynoms in der konvexen Hülle seiner Bézier-Punkte, d.h.*

$$p(t) \in \text{conv}\ (\boldsymbol{\beta}_0, \ldots, \boldsymbol{\beta}_n) := \{x \in \mathbb{R}^d : x = \sum_{i=0}^{n} \lambda_i \boldsymbol{\beta}_i,\ 0 \le \lambda_i \le 1,\ \sum_{i=0}^{n} \lambda_i = 1\}\ .$$

Beweis. Sei $p(t) = \sum_{k=0}^{n} \boldsymbol{\beta}_k b_k^n(t)$. Die Behauptung folgt nun direkt aus

$$\sum_{k=0}^{n} b_k^n(t) = 1 \text{ und } \ b_k^n(t) \ge 0 \text{ für } \ t \in [0,1].$$

□



Kontrollpolygon und kubische Bézier-Kurve.

Satz 3.6. *Für die Bézier-Darstellung* $p(t) = \sum_{k=0}^{n} \boldsymbol{\beta}_k b_k^n(t)$ *gilt die Differentiationsformel*

$$p^{(k)}(t) = \frac{n!}{(n-k)!} \sum_{j=0}^{n-k} (\Delta^k \boldsymbol{\beta}_j)\, b_j^{n-k}(t),$$

wobei $\Delta^k \boldsymbol{\beta}_j$ *die Vorwärtsdifferenzen sind,*

$$\Delta^0 \boldsymbol{\beta}_j := \boldsymbol{\beta}_j, \quad \Delta^1 \boldsymbol{\beta}_j := \boldsymbol{\beta}_{j+1} - \boldsymbol{\beta}_j, \quad \Delta^k \boldsymbol{\beta}_j := \Delta^{k-1} \boldsymbol{\beta}_{j+1} - \Delta^{k-1} \boldsymbol{\beta}_j.$$

Beweis. Für die erste Ableitung von $p(t)$ erhalten wir

$$\begin{aligned} \frac{d}{dt}\, b_j^n(t) &= \underbrace{\binom{n}{j} j}_{n\binom{n-1}{j-1}}\, t^{j-1}(1-t)^{n-j} - \underbrace{\binom{n}{j}(n-j)}_{n\binom{n-1}{j}}\, t^j (1-t)^{n-1-j} \\ &= n[b_{j-1}^{n-1}\ (t) - b_j^{n-1}(t)]. \end{aligned}$$

Insbesondere ist $\frac{d}{dt} b_0^n(t) = -n b_0^{n-1}(t)$ und $\frac{d}{dt} b_n^n(t) = n b_{n-1}^{n-1}(t)$. Daraus folgt

$$\begin{aligned} p'(t) &= \sum_{k=0}^{n} \boldsymbol{\beta}_k\, (b_k^n(t))' = \sum_{k=0}^{n} \boldsymbol{\beta}_k\, n\, (b_{k-1}^{n-1}(t) - b_k^{n-1}(t)) \\ &= \sum_{k=1}^{n} n\boldsymbol{\beta}_k\, b_{k-1}^{n-1}(t) - \sum_{k=0}^{n-1} n\boldsymbol{\beta}_k\, b_k^{n-1}(t) \\ &= \sum_{k'=0}^{n-1} n\boldsymbol{\beta}_{k'+1}\ \ b_{k'}^{n-1}\ (t) - \sum_{k=0}^{n-1} n\boldsymbol{\beta}_k\, b_k^{n-1}(t) \\ &= \sum_{k=0}^{n-1} n(\underbrace{\boldsymbol{\beta}_{k+1} - \boldsymbol{\beta}_k}_{\Delta^1 \boldsymbol{\beta}_k})\, b_k^{n-1}(t). \end{aligned}$$

Die allgemeine Formel folgt nun durch vollständige Induktion. □

Folgerung 3.7. *Für die Randpunkte $t = 0$ und $t = 1$ erhält man die Werte*

$$p^{(k)}(0) = \frac{n!}{(n-k)!}\Delta^k \boldsymbol{\beta}_0, \qquad p^{(k)}(1) = \frac{n!}{(n-k)!}\Delta^k \boldsymbol{\beta}_{n-k}.$$

Insbesondere gilt

$$\begin{aligned} p(0) &= \boldsymbol{\beta}_0, & p(1) &= \boldsymbol{\beta}_n, \\ p'(0) &= n(\boldsymbol{\beta}_1 - \boldsymbol{\beta}_0), & p'(1) &= n(\boldsymbol{\beta}_n - \boldsymbol{\beta}_{n-1}), \\ p''(0) &= n(n-1)(\boldsymbol{\beta}_2 - 2\boldsymbol{\beta}_1 + \boldsymbol{\beta}_0), & p''(1) &= n(n-1)(\boldsymbol{\beta}_n - 2\boldsymbol{\beta}_{n-1} + \boldsymbol{\beta}_{n-2}). \end{aligned}$$

Satz 3.8. *(Gradanhebung)*
Gegeben seien $n+1$ Bézier-Punkte $\boldsymbol{\beta}_0, \ldots, \boldsymbol{\beta}_n$. Definiert man

$$\hat{\boldsymbol{\beta}}_k := \frac{k}{n+1}\boldsymbol{\beta}_{k-1} + \Big(1 - \frac{k}{n+1}\Big)\,\boldsymbol{\beta}_k, \qquad k = 0, 1, \ldots, n+1,$$

so gilt

$$\sum_{k=0}^{n} \boldsymbol{\beta}_k\; b_k^n(t) = \sum_{k=0}^{n+1} \hat{\boldsymbol{\beta}}_k\; b_k^{n+1}(t).$$

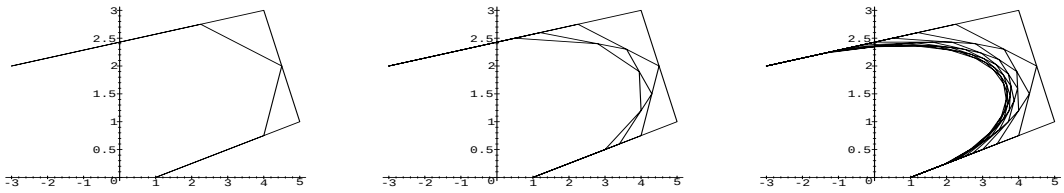
Beweis. Es gilt

$$\begin{aligned} & \frac{(n+1-k)}{(n+1)}\, b_k^{n+1}(t) + \frac{(k+1)}{(n+1)}\, b_{k+1}^{n+1}(t) \\ = \; & \underbrace{\frac{(n+1-k)}{(n+1)}\binom{n+1}{k}}_{\frac{n!}{k!(n-k)!}=\binom{n}{k}}\, t^k(1-t)^{n+1-k} + \underbrace{\frac{(k+1)}{(n+1)}\binom{n+1}{k+1}}_{\binom{n}{k}}\, t^{k+1}(1-t)^{n-k} \\ = \; & \binom{n}{k} t^k (1-t)^{n-k}((1-t)+t) = b_k^n(t). \end{aligned}$$

Daher ist

$$\begin{aligned} & \sum_{k=0}^{n} \boldsymbol{\beta}_k\; b_k^n(t) = \sum_{k=0}^{n} \boldsymbol{\beta}_k \Big(\frac{(n+1-k)}{(n+1)}\, b_k^{n+1}(t) + \frac{(k+1)}{(n+1)}\; b_{k+1}^{n+1}(t) \Big) \\ = \; & \sum_{k=0}^{n} \boldsymbol{\beta}_k \Big(1 - \frac{k}{n+1}\Big)\, b_k^{n+1}(t) + \sum_{k=1}^{n+1}\; \boldsymbol{\beta}_{k-1} \frac{k}{n+1}\, b_k^{n+1}(t) \\ = \; & \boldsymbol{\beta}_0\, b_0^{n+1}(t) + \sum_{k=1}^{n} \Big(\frac{k}{n+1}\boldsymbol{\beta}_{k-1} + \Big(1 - \frac{k}{n+1}\Big)\boldsymbol{\beta}_k \Big)\; b_k^{n+1}(t) + \boldsymbol{\beta}_n\, b_{n+1}^{n+1}(t) \\ = \; & \sum_{k=0}^{n+1} \hat{\boldsymbol{\beta}}_k\, b_k^{n+1}(t). \end{aligned}$$

□



Gradanhebung um 1 (links), um 3 (Mitte) und um 10 (rechts). Kontrollpunkte zu Beginn sind $\boldsymbol{\beta}_0 = (1.0, 0.0)$, $\boldsymbol{\beta}_1 = (5.0, 1.0)$, $\boldsymbol{\beta}_2 = (4.0, 3.0)$, $\boldsymbol{\beta}_3 = (-3.0, 2.0)$.

Algorithmus von de Casteljau

Problem: Gesucht ist ein effektiver Algorithmus zur Berechnung des Bézier-Polynoms $p(t) = \sum_{k=0}^{n} \boldsymbol{\beta}_k b_k^n(t)$.

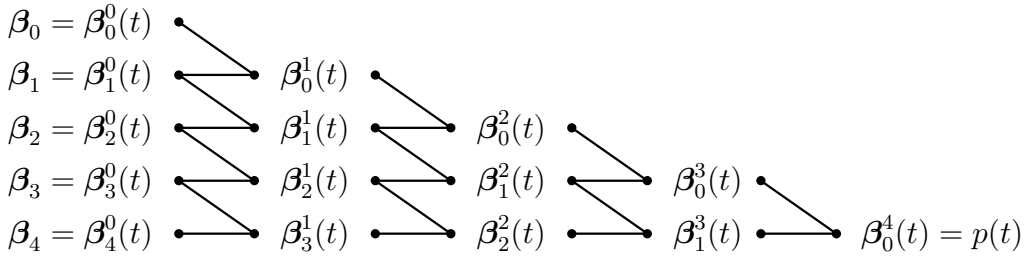
Der Algorithmus von de Casteljau beruht auf der Rekursionsformel für Bernstein-Polynome.

Gegeben: $\boldsymbol{\beta}_k =: \boldsymbol{\beta}_k^0(t), \quad k = 0, \ldots, n.$
Wir nutzen $b_k^n(t) = (1-t)b_k^{n-1}(t) + t b_{k-1}^{n-1}(t)$ und erhalten

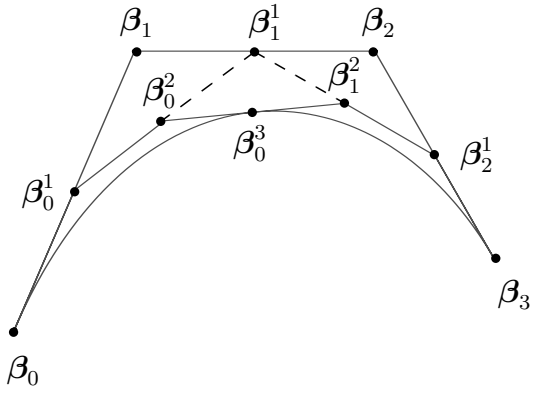
$$\begin{aligned}
p(t) &= \sum_{k=0}^{n} \boldsymbol{\beta}_k^0(t) b_k^n(t) \\
&= \sum_{k=0}^{n} \boldsymbol{\beta}_k^0(t) \left[(1-t)b_k^{n-1}(t) + t b_{k-1}^{n-1}(t)\right] \\
&= \sum_{k=0}^{n-1} \underbrace{\left[\boldsymbol{\beta}_k^0(t)(1-t) + t\,\boldsymbol{\beta}_{k+1}^0(t)\right]}_{\boldsymbol{\beta}_k^1(t)} b_k^{n-1} \\
&\;\;\vdots \\
&= \sum_{k=0}^{1} \boldsymbol{\beta}_k^{n-1}(t)\, b_k^1(t) = \boldsymbol{\beta}_0^{n-1}(t)\,(1-t) + \boldsymbol{\beta}_1^{n-1}(t)\, t = \boldsymbol{\beta}_0^n(t).
\end{aligned}$$

Also ist $\boldsymbol{\beta}_0^n(t)$ der Wert des Polynoms p an der Stelle t. Dabei ist $\boldsymbol{\beta}_k^j(t)$ jeweils eine konvexe Linearkombination von $\boldsymbol{\beta}_k^{j-1}(t)$ und $\boldsymbol{\beta}_{k+1}^{j-1}(t)$.

Berechnung mit einem Dreiecksschema

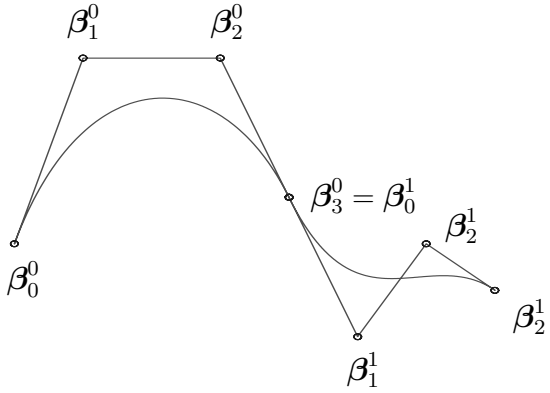


Beispiel: Sei $n = 3$. Die Werte $\boldsymbol{\beta}_k^j$ liegen immer auf der Verbindungsstrecke von $\boldsymbol{\beta}_k^{j-1}$ und $\boldsymbol{\beta}_{k+1}^{j-1}$. Für $t = \frac{1}{2}$ erhalten wir die folgende graphische Darstellung.



Segmentierung einer kubischen Bézier-Kurve.

Bemerkung: Der de Casteljau-Algorithmus erlaubt eine Segmentierung in zwei Bézierpolynome. Im obigen Beispiel besitzt das erste Bézierpolynom die Kontrollpunkte $\boldsymbol{\beta}_0$, $\boldsymbol{\beta}_0^1$, $\boldsymbol{\beta}_0^2$, $\boldsymbol{\beta}_0^3$ und das zweite Bézierpolynom die Kontrollpunkte $\boldsymbol{\beta}_0^3$, $\boldsymbol{\beta}_1^2$, $\boldsymbol{\beta}_2^1$, $\boldsymbol{\beta}_3$. Zur graphischen Darstellung wird der Prozess einfach einige Male wiederholt. Die erhaltenem Bézierpolygone sind dann gute Approximationen an das Bezierpolynom p.



Zusammengesetzte kubische Bézier-Kurve.

Zusammensetzen von Bezier-Kurven

Satz 3.9. *Eine zusammengesetzte Bézier-Kurve s mit*

$$s(t) = \begin{cases} \sum_{k=0}^{n} \boldsymbol{\beta}_k^0 \, b_k^n(t) & t \in [0,1], \\ \sum_{k=0}^{n} \boldsymbol{\beta}_k^1 \, b_k^n(t-1) & t \in [1,2], \end{cases}$$

ist genau dann C^r-stetig, wenn $\Delta^j \boldsymbol{\beta}^0_{n-j} = \Delta^j \boldsymbol{\beta}^1_0$ für $j = 0, \ldots, r$ gilt.
Speziell: Die Kurve $s(t)$ ist stetig, falls $\boldsymbol{\beta}^0_n = \boldsymbol{\beta}^1_0$ gilt.
Die Kurve $s(t)$ ist stetig differenzierbar, falls $\boldsymbol{\beta}^0_n = \boldsymbol{\beta}^1_0$ und $\boldsymbol{\beta}^0_n - \boldsymbol{\beta}^0_{n-1} = \boldsymbol{\beta}^1_1 - \boldsymbol{\beta}^1_0$.

Beweis. Der Beweis folgt direkt aus Satz 3.6 bzw. Folgerung 3.7. □

3.3 B-Spline-Kurven

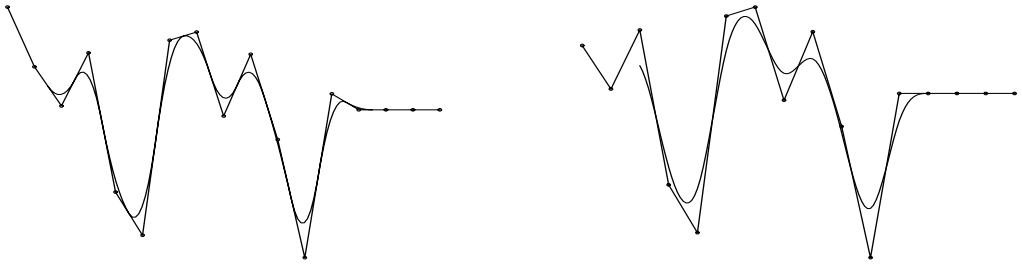
Definition 3.10. *Seien $m \in \mathbb{N}$ und eine Knotenfolge $X = (x_i)_{i\in\mathbb{Z}}$ mit $x_i < x_{i+1}$ für alle $i \in \mathbb{Z}$ gegeben. Mit $N_{i,m}$ seien die zugehörigen B-Splines bezeichnet. Dann heißt*

$$s(t) = \sum_{i\in\mathbb{Z}} \mathbf{d}_i N_{i,m}(t) \tag{3.1}$$

eine **B-Spline-Kurve der Ordnung** *m. Die Koeffizienten $\mathbf{d}_i \in \mathbb{R}^d$ heißen* **de Boor-Punkte** *oder* **Kontrollpunkte**.

Da B-Splines $N_{i,m}$ außerhalb des Intervalls $[x_i, x_{i+m}]$ verschwinden, ist für festes $t \in \mathbb{R}$ immer nur eine endliche Summe zu berechnen. Weiterhin ergeben sich folgende Eigenschaften der B-Spline-Kurve.

Satz 3.11. *Verändert man in einer B-Spline-Kurve $s(t)$ der Form (3.1) den Kontrollpunkt $\mathbf{d}_i$, so ändert sich die Kurve nur lokal in (x_i, x_{i+m}).*
Wegen $\sum_{i\in\mathbb{Z}} N_{i,m}(t) = 1$ (siehe Satz 1.39), sind B-Spline-Kurven (analog wie Bézier-Polynome) in der konvexen Hülle ihrer Kontrollpunkte enthalten.



Quadratische B-Spline-Kurve (links) und kubische B-Spline-Kurve (rechts) mit Kontrollpolygon.

Falls $d_i \in \mathbb{R}$, wählen wir die Kontrollpunkte $(\frac{x_{i+1}+\ldots+x_{i+m-1}}{m-1}, d_i)$ da für $m > 1$

$$\sum_{i\in\mathbb{Z}} \Big(\frac{x_{i+1} + \ldots + x_{i+m-1}}{m-1}\Big) N_{i,m}(t) = t$$

gilt. Diese Formel folgt mit Hilfe der Rekursion in Satz 1.38 durch vollständige Induktion.

Der de Boor-Algorithmus

Zur Auswertung von $s(t)$ an einer bestimmten Stelle t, wollen wie einen Algorithmus herleiten (analog zum de Casteljau-Algorithmus). Dabei verwenden wir die Rekursionsformel (Satz 1.38)

$$N_{i,m}(t) = w_{i,m}(t)\, N_{i,m-1}(t) + (1 - w_{i+1,m}(t))\, N_{i+1,m-1}(t)$$

mit

$$w_{i,m}\ (t) := \frac{t - x_i}{x_{i+m-1} - x_i} \quad (\Rightarrow 1 - w_{i+1,m}(t) = \frac{x_{i+m} - t}{x_{i+m} - x_{i+1}}).$$

Wir erhalten

$$\begin{aligned}
s(t) &= \sum_{i\in\mathbb{Z}} \mathbf{d}_i N_{i,m}(t) \\
&= \sum_{i\in\mathbb{Z}} \mathbf{d}_i [w_{i,m}(t)\, N_{i,m-1}(t) + (1 - w_{i+1,m}(t))\, N_{i+1,m-1}(t)] \\
&= \sum_{i\in\mathbb{Z}} \underbrace{[w_{i,m}(t)\mathbf{d}_i + (1 - w_{i,m}(t))\mathbf{d}_{i-1}]}_{:=\mathbf{d}_i^1(t)}\ N_{i,m-1}(t) \\
&\ \ \vdots \\
&= \sum_{i\in\mathbb{Z}} \mathbf{d}_i^{m-1}(t)\, N_{i,1}(t).
\end{aligned}$$

Wegen

$$N_{j,1}(t) = \begin{cases} 1 & t \in [x_j, x_{j+1}) \\ 0 & \text{sonst} \end{cases}$$

folgt für B-Spline-Kurven der Ordnung 1 $s(t) = \mathbf{d}_i^{m-1}\ (t)$ für $t \in [x_i, x_{i+1})$.

Algorithmus

Gegeben: Knotenfolge $X = \{x_i\}_{i\in\mathbb{Z}}$,
$t \in [x_j, x_{j+1})$,
Kontrollpunkte $\mathbf{d}_i \in \mathbb{R}^d$ für $i = j - m + 1, \ldots, j$,

Gesucht: $s(t) = \sum_{i\in\mathbb{Z}}\ \mathbf{d}_i\, N_{i,m}(t)$.

1. Setze $\mathbf{d}_i^0 := \mathbf{d}_i$, $i = j - m + 1, \ldots, j$.
2. Berechne für $k = 1, \ldots, m - 1$
   Berechne für $i = j, \ldots, j - m + k + 1$

$$\mathbf{d}_i^k := w_{i,m-k+1}(t)\, \mathbf{d}_i^{k-1}\ + (1 - w_{i,m-k+1}(t))\mathbf{d}_{i-1}^{k-1}$$

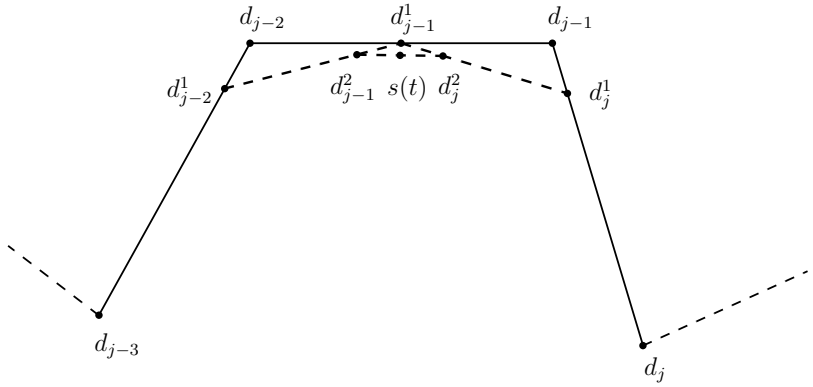
mit $w_{i,m-k+1}(t) := \frac{t-x_i}{x_{i+m-k}-x_i}$.
Ausgabe: $s(t) = \mathbf{d}_j^{m-1}(t)$.

Beispiel: $m = 4$ (kubische Splines)
Sei $x_i = i$ für $i \in \mathbb{Z}$ und sei $t = \frac{1}{2}(x_j + x_{j+1})$ für ein festes j. Für gegebene $\mathbf{d}_i^0 = \mathbf{d}_i,\ i = j-3, j-2, j-1, j$ berechnen wir $s(t) = \sum_{i\in\mathbb{Z}} \mathbf{d}_i\, N_{i,4}(t)$.

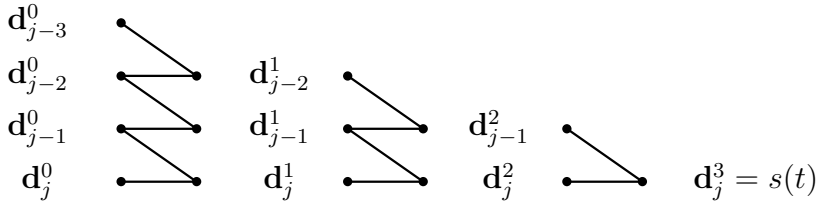
Wir erhalten

$$w_{i,4}(t) = \frac{t-i}{i+m-1-i} = \frac{t-i}{m-1} = \frac{t-i}{3}.$$

und verwenden zur Berechnung von $\mathbf{d}_i^k$ ein Dreiecksschema.



Berechnung eines Funktionswertes der kubischen B-Spline-Kurve.



wobei

$$\begin{aligned}
\mathbf{d}_{j-2}^1 &= w_{j-2,4}\,\mathbf{d}_{j-2}^0 + (1 - w_{j-2,4})\,\mathbf{d}_{j-3}^0 \\
&= \frac{t-(j-2)}{3}\ \mathbf{d}_{j-2}^0\ + (1 - \frac{t-(j-2)}{3})\,\mathbf{d}_{j-3}^0 = \frac{5}{6}\,\mathbf{d}_{j-2}^0 + \ \frac{1}{6}\,\mathbf{d}_{j-3}^0, \\
\mathbf{d}_{j-1}^1 &= w_{j-1,4}\ \mathbf{d}_{j-1}^0\ + (1 - w_{j-1,4})\,\mathbf{d}_{j-2}^0\ = \frac{3}{6}\ \mathbf{d}_{j-1}^0\ + \frac{3}{6}\ \mathbf{d}_{j-2}^0, \\
\mathbf{d}_j^1 &= \frac{1}{6}\ \mathbf{d}_j^0 + \frac{5}{6}\ \mathbf{d}_{j-1}^0.
\end{aligned}$$

Analog folgt

$$\begin{aligned}
\mathbf{d}^2_{j-1} &= w_{j-1,3}\,\mathbf{d}^1_{j-1} + (1-w_{j-1,3})\mathbf{d}^1_{j-2} = \frac{3}{4}\,\mathbf{d}^1_{j-1} + \frac{1}{4}\,\mathbf{d}^1_{j-2},\\
\mathbf{d}^2_{j} &= \frac{1}{4}\,\mathbf{d}^1_{j} + \frac{3}{4}\,\mathbf{d}^1_{j-1},\\
\mathbf{d}^3_{j} &= \frac{1}{2}\,\mathbf{d}^2_{j} + \frac{1}{2}\,\mathbf{d}^2_{j-1}.
\end{aligned}$$

Satz 3.12. *(Knoteneinfügung) (Böhm)*
Sei $m \in \mathbb{N}$ und $X = (x_i)_{i\in\mathbb{Z}}$ eine Knotenfolge mit $x_i < x_{i+1}$ für alle $i \in \mathbb{Z}$ und seien $N_{i,m}$ die zugehörigen B-Splines. Gegeben sei ein $\hat{x} \in (x_j, x_{j+1})$. Eine verfeinerte Knotenfolge $\hat{X} := (\hat{x}_i)_{i\in\mathbb{Z}}$ sei durch

$$\hat{x}_i := \begin{cases} x_i & i \le j,\\ \hat{x} & i = j+1,\\ x_{i-1} & i \ge j+2, \end{cases}$$

definiert. Mit $\hat{N}_{i,m}$ seien die zur Knotenfolge $\hat{X}$ gehörende B-Splines bezeichnet. Zu einer Folge von Kontrollpunkten $(\mathbf{d}_i)_{i\in\mathbb{Z}}$ definiere die Folge $(\hat{\mathbf{d}}_i)_{i\in\mathbb{Z}}$ durch

$$\hat{\mathbf{d}}_i := \begin{cases} \mathbf{d}_i & i \le j-m+1,\\ \frac{\hat{x}-x_i}{x_{i+m-1}-x_i}\mathbf{d}_i + \frac{x_{i+m-1}-\hat{x}}{x_{i+m-1}-x_i}\mathbf{d}_{i-1} & j-m+2 \le i \le j,\\ \mathbf{d}_{i-1} & i \ge j+1. \end{cases}$$

Dann ist $\sum_{i\in\mathbb{Z}} \mathbf{d}_i\, N_{i,m} = \sum_{i\in\mathbb{Z}} \hat{\mathbf{d}}_i\, \hat{N}_{i,m}$.

Beweis. Nach Definition 1.36 war

$$N_{i,m}(t) := (x_{i+m} - x_i)\, u_{t,m-1}[x_i, \ldots, x_{i+m}]$$

mit $u_{t,m-1}(x) := (x-t)^{m-1}_+$ definiert. Es gilt nun wegen der Rekursionsformel für dividierte Differenzen

$$\begin{aligned}
& (x_i - x_{i+m})\, u_{t,m-1}[x_i, \ldots, x_{i+m}] + (\hat{x} - x_i)\, u_{t,m-1}[x_i, \ldots, x_{i+m-1}, \hat{x}]\\
& +(x_{i+m} - \hat{x})\, u_{t,m-1}[\hat{x}, x_{i+1}, \ldots, x_{i+m}]\\
= \;& (x_i - x_{i+m})\Big(\frac{u_{t,m-1}[x_i, \ldots, x_{i+m-1}] - u_{t,m-1}[x_{i+1}, \ldots, x_{i+m}]}{x_i - x_{i+m}}\Big)\\
& +(\hat{x} - x_i)\Big(\frac{u_{t,m-1}[x_{i+1}, \ldots, x_{i+m-1}, \hat{x}] - u_{t,m-1}[x_i, \ldots, x_{i+m-1}]}{\hat{x} - x_i}\Big)\\
& +(x_{i+m} - \hat{x})\Big(\frac{u_{t,m-1}[x_{i+1}, \ldots, x_{i+m}] - u_{t,m-1}[\hat{x}, x_{i+1}, \ldots, x_{i+m-1}]}{x_{i+m} - \hat{x}}\Big)\\
= \;& 0.
\end{aligned}$$

Daraus folgt für $i = j-m+1,\ldots,j$ wegen $\hat{x} \in \text{supp}N_{i,m} = [i, i+m]$

$$\begin{aligned} N_{i,m}(t) &= (\hat{x}-x_i)\,\frac{\hat{N}_{i,m}(t)}{(\hat{x}_{i+m}-x_i)} + (x_{i+m} - \hat{x})\,\frac{\hat{N}_{i+1,m}(t)}{(x_{i+m}-\hat{x}_{i+1})} \\ &= (\hat{x}_{j+1}-\hat{x}_i)\,\frac{\hat{N}_{i,m}(t)}{(\hat{x}_{i+m}-\hat{x}_i)} + (\hat{x}_{i+m+1}-\hat{x}_{j+1})\,\frac{\hat{N}_{i+1,m}(t)}{(\hat{x}_{i+m+1}-\hat{x}_{i+1})}, \end{aligned}$$

denn $\hat{x} = \hat{x}_{j+1}$, $x_{i+m} = \hat{x}_{i+m+1}$ und $x_i = \hat{x}_1$ für $i = j-m+1,\ldots,j$. Für $i < j-m+1$ bzw. $i > j$ ergibt sich keine Änderung, d.h.,

$$\begin{aligned} N_{i,m}(t) &= \hat{N}_{i,m}(t) \quad \text{für } i < j-m+1, \\ N_{i,m}(t) &= \hat{N}_{i+1,m}(t) \quad \text{für } i > j. \end{aligned}$$

Damit folgt

$$\begin{aligned} \sum_{i\in\mathbb{Z}} \mathbf{d}_i N_{i,m}(t) &= \sum_{i<j-m+1} \mathbf{d}_i \hat{N}_{i,m}(t) + \sum_{i>j} \mathbf{d}_i \hat{N}_{i+1,m}(t) \\ &\quad + \sum_{i=j-m+1}^{j} \mathbf{d}_i \left(\frac{\hat{x}-\hat{x}_i}{\hat{x}_{i+m}-\hat{x}_i}\hat{N}_{i,m}(t) + \frac{\hat{x}_{i+m+1}-\hat{x}}{\hat{x}_{i+m+1}-\hat{x}_{i+1}} N_{i+1,m}(t) \right). \end{aligned}$$

Für den letzten Term ergibt sich durch Indexverschiebung

$$\begin{aligned} &\sum_{i=j-m+1}^{j} \frac{\hat{x}-\hat{x}_i}{\hat{x}_{i+m}-\hat{x}_i}\,\mathbf{d}_i\,\hat{N}_{i,m}(t) + \sum_{i=j-m+2}^{j+1} \frac{\hat{x}_{i+m}-\hat{x}}{\hat{x}_{i+m}-\hat{x}_i}\;\mathbf{d}_{i-1}\;\hat{N}_{i,m}(t) \\ = \;& \mathbf{d}_{j-m+1}\hat{N}_{j-m+1}(t) + \mathbf{d}_j \hat{N}_{j+1}(t) \\ &+ \sum_{i=j-m+2}^{j} \Big(\frac{\hat{x}-\hat{x}_i}{\hat{x}_{i+m}-\hat{x}_i}\mathbf{d}_i + \frac{\hat{x}_{i+m}-\hat{x}}{\hat{x}_{i+m}-\hat{x}_i}\mathbf{d}_{i-1} \Big) \hat{N}_{i,m}(t). \end{aligned}$$

Damit folgt die Behauptung. □

3.4 Subdivision-Algorithmus

Sei nun die Knotenfolge $x_i = i \quad i \in \mathbb{Z}$ gegeben, und $N_{i,m}$ seien die zugehörigen B-Splines der Ordnung m (**kardinale B-Splines**).

Satz 3.13. *Es gilt*

(i) *Für $m = 1$ ist der kardinale B-Spline definiert durch*

$$N_{0,1}\,(t) = \begin{cases} 1 & t \in (0,1], \\ 0 & \text{sonst.} \end{cases}$$

(ii) *Es gilt die Rekursion*

$$N_{i,m}(t) = \frac{t-i}{m-1}\ N_{i,m-1}(t) + \frac{i+m-t}{m-1} N_{i+1,m-1}\ (t).$$

(iii) *Für den Träger des kardinalen B-Splines gilt* $\operatorname{supp} N_{i,m} = [i,\, i+m]$. *Insbesondere ist*

$$N_{i,m}(t) = N_{0,m}(t-i).$$

Wir verwenden daher die Schreibweise $N_m(t) := N_{0,m}(t)$.

(iv) *Der kardinale B-Spline kann durch Faltung berechnet werden,*

$$\begin{aligned} N_m\ (t) &= (N_{m-1} * N_1)\ (t) \\ &:= \int_{-\infty}^{\infty} N_{m-1}(t-s)\, N_1(s)\, ds. \end{aligned}$$

(v) *Für die Fouriertransformierte des kardinalen B-Splines gilt*

$$\hat{N}_m(w) := \int_{-\infty}^{\infty} N_m(t)\, e^{-iwt}\ dt = \Big(\frac{1-e^{-iw}}{iw}\Big)^m.$$

Beweis. Eigenschaft (i) folgt direkt aus der Definition 1.36 mit $x_i = i$. Die Rekursion (ii) folgt aus Satz 1.38 mit $x_i = i$ und (iii) aus Satz 1.39.
(iv) Es gilt

$$\begin{aligned} \int_{-\infty}^{\infty} N_{m-1}(t-s)\, N_1(s)\, ds &= \int_0^1 N_{m-1}(t-s) ds \\ &= \int_{t-1}^{t} N_{m-1}(s)\, ds. \end{aligned}$$

Weiterhin finden wir (siehe Übungsaufgabe)

$$N_m'(s) = (m-1)\Big(\frac{N_{m-1}(s)}{m-1} - \frac{N_{m-1}(s-1)}{m-1}\Big) = N_{m-1}(s) - N_{m-1}(s-1).$$

Damit folgt

$$\begin{aligned} \sum_{j=0}^{m-1} N_m'(s-j) &= N_{m-1}(s) - N_{m-1}(s-m) \\ &= N_{m-1}(s) \end{aligned}$$

für $s \in (-\infty, m]$. Wir erhalten also für $t \in [0, m]$

$$\begin{aligned}
\int_{t-1}^{t} N_{m-1}(s)\,ds &= \int_{t-1}^{t} \sum_{j=0}^{m-1} N'_m(s-j)\,ds = \sum_{j=0}^{m-1} N_m(s-j)\Big|_{t-1}^{t} \\
&= \sum_{j=0}^{m-1} (N_m(t-j) - N_m(t-1-j)) \\
&= N_m(t) - \underbrace{N_m(t-m)}_{0} = N_m(t),
\end{aligned}$$

und für $t \notin [0, m]$ ist $\int_{t-1}^{t} N_{m-1}(s)\,ds = 0$ wegen supp $N_{m-1} = [0, m-1]$.
(v) Für den B-Spline erster Ordnung ergibt sich die Fouriertransformierte

$$\begin{aligned}
\hat{N}_1(w) &= \int_{-\infty}^{\infty} N_1(t)\, e^{-iwt}\ dt = \int_0^1 e^{-iwt}\ dt \\
&= \frac{1}{-iw}\ e^{-iwt}\ \Big|_0^1 = \frac{e^{-iw}-1}{-iw}\ = \frac{1-e^{-iw}}{iw}\ .
\end{aligned}$$

Für $m > 1$ gilt mit vollständiger Induktion

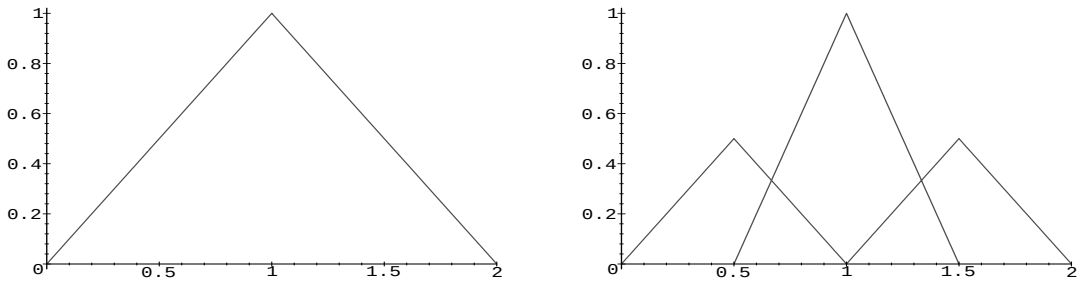
$$\begin{aligned}
\hat{N}_m\ (w) &= \int_{-\infty}^{\infty} (N_{m-1}\ *N_1\ )(t)\, e^{-iwt} dt \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} N_{m-1}(t-s)\, N_1(s)\, ds\, e^{-iwt}\, dt \\
&= \int_{-\infty}^{\infty} N_1(s)\, e^{-iws} \int_{-\infty}^{\infty} N_{m-1}(t-s)\, \underbrace{e^{-iwt}\, e^{iws}}_{e^{-iw(t-s)}}\ dt\ ds \\
&= \int_{-\infty}^{\infty} N_1(s)\, e^{-iws} ds \int_{-\infty}^{\infty} N_{m-1}(t')\, e^{-iwt'}\ dt' \\
&= \hat{N}_1(w)\, \hat{N}_{m-1}(w)\ = \Big(\frac{1-e^{-iw}}{iw}\Big)^m.
\end{aligned}$$

□

Wir betrachten nun die B-Spline-Kurve

$$s(t) = \sum_{i \in \mathbb{Z}} \mathbf{d}_i N_m(t-i)$$

und suchen nach einem schnellen Algorithmus zur graphischen Darstellung von $s(t)$ durch Berechnung eines "besseren" Kontrollpolygons.



Geometrische Darstellung der Zwei-Skalen-Relation für lineare Splines.

Idee: Stelle $N_m(t)$ durch B-Splines auf dem Gitter $\mathbb{Z}/2$ dar:
Beispiel: $m = 2$

Der lineare kardinale B-Spline lässt sich als Linearkombination linearer Splines auf dem Gitter $\mathbb{Z}/2$ auffassen,

$$N_2(t) = \frac{1}{2}\ N_2(2t) + N_2(2t-1) + \frac{1}{2}\ N_2(2t-2).$$

Allgemein gilt:

Satz 3.14. *Die kardinalen B-Splines erfüllen die Zwei-Skalen-Gleichung*

$$N_m(t) = \frac{1}{2^{m-1}} \sum_{j=0}^{m} \binom{m}{j}\ N_m(2t-j).$$

Beweis. Es gilt nach Satz 3.13

$$\begin{aligned} \hat{N}_m(w) &= \Big(\frac{1-e^{-iw}}{iw}\Big)^m = \Big(\frac{1+e^{-iw/2}}{2}\Big)^m \Big(\frac{1-e^{-iw/2}}{iw/2}\Big)^m \\ &= \Big(\frac{1+e^{-iw/2}}{2}\Big)^m \hat{N}_m\ (\frac{w}{2}). \end{aligned}$$

Durch Anwendung der inversen Fouriertransformation ($f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{f}(w)\, e^{iwt}\, dw$ für $f \in L^2(\mathbb{R}) \cap L^1(\mathbb{R})$) erhalten wir

$$
\begin{aligned}
N_m(t) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \Big(\frac{1+e^{-iw/2}}{2}\Big)^m \hat{N}_m\Big(\frac{w}{2}\Big)\ e^{iwt}\ dw \\
&= \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{1}{2^m} \sum_{j=0}^{m} \binom{m}{j} e^{-iw'j}\, \hat{N}_m(w')\, e^{2iw't}\, 2dw' \\
&= \frac{1}{2^{m-1}} \sum_{j=0}^{m} \binom{m}{j} \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{N}_m(w')\, e^{iw(2t-j)}\, dw \\
&= \frac{1}{2^{m-1}} \sum_{j=0}^{m} \binom{m}{j} N_m(2t-j).
\end{aligned}
$$

□

Wir erhalten somit folgende neue Darstellung für die B-Spline-Kurve $s(t)$,

$$
\begin{aligned}
s(t) &= \sum_{i\in\mathbb{Z}} \mathbf{d}_i\, N_m(t-i) \\
&= \sum_{i\in\mathbb{Z}} \mathbf{d}_i \frac{1}{2^{m-1}} \sum_{j=0}^{m} \binom{m}{j}\ N_m(2(t-i)-j).
\end{aligned}
$$

Sei nun

$$
a_j := \begin{cases} \frac{1}{2^{m-1}} \binom{m}{j} & j = 0,\ldots,m \\ 0 & j<0 \text{ oder } j>m. \end{cases}
$$

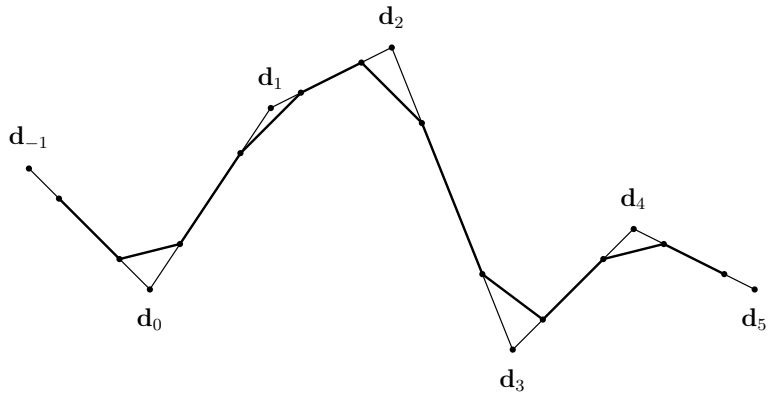
Dann folgt

$$
\begin{aligned}
s(t) &= \sum_{j\in\mathbb{Z}} \sum_{i\in\mathbb{Z}} a_j\, \mathbf{d}_i\, N_m(2t-2i-j) \\
&= \sum_{j'\in\mathbb{Z}} \underbrace{\Big(\sum_{i\in\mathbb{Z}} a_{j'-2i}\, \mathbf{d}_i\Big)}_{:=\mathbf{d}^{(1)}_{j'}} N_m(2t-j').
\end{aligned}
$$

Beispiel: Für $m=3$ (quadratische Splines) erhalten wir

$$
a_0 = \frac{1}{4}\ , \quad a_1 = \frac{3}{4}, \quad a_2 = \frac{3}{4}, \quad a_3 = \frac{1}{4}\ .
$$

Die neuen Kontrollpunkte $\mathbf{d}^{(1)}_j$ ergeben sich als konvexe Linearkombinationen aus den gegebenen Kontrollpunkten. Der Algorithmus (corner cutting) ist auch unter dem Namen Chaikins Algorithmus in der Literatur bekannt.



Erste Iteration des Chaikins Algorithmus (corner cutting).

Wir erhalten

$$\begin{aligned}
\mathbf{d}_0^{(1)} &= \sum_{i\in\mathbb{Z}} a_{-2i}\mathbf{d}_i = a_0\,\mathbf{d}_0 + a_2\,\mathbf{d}_{-1} = \frac{1}{4}\,\mathbf{d}_0 + \frac{3}{4}\mathbf{d}_{-1},\\
\mathbf{d}_1^{(1)} &= \sum_{i\in\mathbb{Z}} a_{1-2i}\mathbf{d}_i = a_1\,\mathbf{d}_0 + a_3\,\mathbf{d}_{-1} = \frac{3}{4}\,\mathbf{d}_0 + \frac{1}{4}\,\mathbf{d}_{-1},\\
\mathbf{d}_2^{(1)} &= a_2\,\mathbf{d}_0 + a_0\,\mathbf{d}_1 = \frac{3}{4}\,\mathbf{d}_0 + \frac{1}{4}\,\mathbf{d}_1,\\
\mathbf{d}_3^{(1)} &= a_3\,\mathbf{d}_0 + a_1\,\mathbf{d}_1 = \frac{1}{4}\,\mathbf{d}_0 + \frac{3}{4}\,\mathbf{d}_1, \quad \ldots
\end{aligned}$$

Die Prozedur kann nun suksessive fortgesetzt werden:

$$\begin{aligned}
s(t) &= \sum_{j\in\mathbb{Z}} \mathbf{d}_j^{(1)}\, N_m(2t-j) = \sum_{j\in\mathbb{Z}} \mathbf{d}_j^{(2)}\, N_m(4t-j)\\
\ldots &= \sum_{j\in\mathbb{Z}} \mathbf{d}_j^{(k)}\; N_m(2^k t-j),
\end{aligned}$$

wobei $\mathbf{d}_j^{(k)} := \sum_{i\in\mathbb{Z}}\; a_{j-2i}\,\mathbf{d}_j^{(k-1)}$.
Wir zeigen: Das Kontrollpolygon der $\mathbf{d}_j^{(k)}$ konvergiert für $k\to\infty$ gegen die B-Spline-Kurve $s(t)$.

Satz 3.15. *Gegeben sei die Folge $(a_j)_{j\in\mathbb{Z}}$ mit $a_j = \frac{1}{2^{m-1}}\binom{m}{j}$ für $j = 0,\ldots,m$ ($m\in\mathbb{N}, m\geq 2$) und $a_j = 0$ für $j > m$ bzw. $j < 0$. Dann konvergiert das Kontrollpolygon $p_k(t) = \sum_{j\in\mathbb{Z}}\; \mathbf{d}_j^{(k)}\, N_2\;(2^k t - j + 1 - \frac{m}{2}\;)$ gegen die B-Spline-Kurve $s(t) = \sum_{j\in\mathbb{Z}} \mathbf{d}_j N_m(t-j)$ für $k\to\infty$, wobei $(\mathbf{d}_j^0) = (\mathbf{d}_j)$ und*

$$\mathbf{d}_j^{(k)} := \sum_{i\in\mathbb{Z}} a_{j-2i}\,\mathbf{d}_i^{(k-1)}\;.$$

Bemerkung: Die Verschiebung des linearen B-Splines N_2 in obigem Satz um $\frac{m}{2}-1$ ist notwendig um die richtige Zuordnung der Kontrollpunkte im Vergleich zur B-Spline-Kurve $s(t)$ zu gewährleisten, denn N_m und $N_2(\cdot + 1 - \frac{m}{2})$ haben beide ihr Maximum in $\frac{m}{2}$.

Zum Beweis benötigen wir einige Lemmata.

Lemma 3.16. *Es existiert eine Konstante K_m mit $0 < K_m < \infty$, so dass für beliebige beschränkte Folgen $\mathbf{c} = (c_j)_{j\in\mathbb{Z}}$ gilt*

$$\|\mathbf{c}\|_\infty = \sup_{j\in\mathbb{Z}} |c_j| \leq K_m \,\|\sum_{j\in\mathbb{Z}} c_j N_m(\cdot - j)\|_\infty.$$

Beweis. Ein erster Beweis dieser Aussage geht zurück auf Carl de Boor, 1968. Die bisher beste Schranke $K_m \leq m\,2^{m-1}$ wurde von Karl Scherer und A. Yu. Shadrin gefunden (siehe: "New upper bound for the B-spline basis condition number II. A proof of de Boor's 2^k-conjecture", Journal of Approximation Theory 99 (1999), S. 217–229. □

Lemma 3.17. *Sei f eine stetige Funktion und*

$$w(f,h) := \sup_{|t|<h} \|f - f(\cdot - t)\|_\infty$$

das Stetigkeitsmodul von f. Dann gilt

$$I_1 := \|f - \sum_{j\in\mathbb{Z}} f(jh) N_m(\frac{\cdot}{h} - j)\|_\infty \leq m\; w(f,h),$$

wobei N_m der B-Spline der Ordnung $m \geq 2$ ist.

Beweis. Wegen $\sum_{j\in\mathbb{Z}} N_m(t-j) = 1$ folgt

$$\begin{aligned} I_1 &= \|f - \sum_{j\in\mathbb{Z}} f(jh) N_m(\frac{\cdot}{h} - j)\|_\infty \\ &= \sup_{x\in\mathbb{R}} |\sum_{j\in\mathbb{Z}} (f(x) - f(jh))\, N_m(\frac{x}{h} - j)|. \end{aligned}$$

Außerdem gilt $\mathrm{supp} N_m(\frac{\cdot}{h} - j) = [jh, (j+m)h]$. Also ist für festes $x \in \mathbb{R}$ die obige Summe über j endlich, denn nur für $x \in [jh, (j+m)h]$ ist $N_m(\frac{x}{h} - j) \neq 0$. Wir wählen nun $k \in \mathbb{Z}$, so dass $kh \leq x < (k+1)h$, und erhalten

$$\sum_{j\in\mathbb{Z}} (f(x) - f(jh))\, N_m(\frac{x}{h} - j)) = \sum_{j=k-m+1}^{k} (f(x) - f(jh))\, N_m(\frac{x}{h} - j).$$

Aus

$$|x - jh| < (k+1)\,h - (k-m+1)\,h = mh$$

folgt daher

$$\begin{aligned} I_1 &\leq \sup_{x\in\mathbb{R}} \sum_{j\in\mathbb{R}} \underbrace{|f(x) - f(jh)|}_{\leq w(f,mh)} \underbrace{N_m(\frac{x}{h} - j)}_{>0} \\ &\leq w(f,mh) \sup_{x\in\mathbb{R}} \sum_{j\in\mathbb{Z}} N_m(\frac{x}{h} - j) = w(f,mh) \leq m\, w(f,h). \end{aligned}$$

□

Lemma 3.18. *Gegeben sei die Folge $(a_j)_{j\in\mathbb{Z}}$ mit*

$$a_j = \begin{cases} \frac{1}{2^{m-1}}\binom{m}{j} & j = 0,\ldots,m, \\ 0 & j < 0 \text{ oder } j > m. \end{cases}$$

Weiter sei $(a_j^{(0)}) := (\delta_{j,0})$ und $a_j^{(k)}$ für alle $j \in \mathbb{Z}$ definiert durch

$$\begin{aligned} a_j^{(1)} &:= \sum_{i\in\mathbb{Z}} a_i^{(0)}\; a_{j-2i} = a_j, \\ a_j^{(k)} &:= \sum_{i\in\mathbb{Z}} a_i^{(k-1)}\; a_{j-2i} \qquad k = 2,3,4,\ldots. \end{aligned}$$

Dann gilt

$$\lim_{k\to\infty} \|N_m - \sum_{j\in\mathbb{Z}} a_j^{(k)}\; N_2(2^k\cdot - j + 1 - m/2)\|_\infty = 0.$$

Beispiel: $m = 3$ (quadratische Splines). Wir finden $a_0 = \frac{1}{4},\ a_1 = \frac{3}{4},\ a_2 = \frac{3}{4},\ a_3 = \frac{1}{4}$. Das 0. Kontrollpolygon lautet $N_2(x - \frac{1}{2})$. Für das erste Kontrollpolygon erhalten wir mit $a_j^{(1)} = a_j,\ j = 0,1,2,3,$

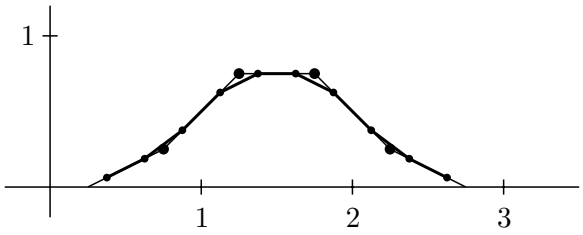
$$\frac{1}{4}\; N_2(2x - \frac{1}{2}\,) + \frac{3}{4}\; N_2(2x - \frac{3}{2}\,) + \frac{3}{4}\; N_2(2x - \frac{5}{2}\,) + \frac{1}{4}\; N_2(2x - \frac{7}{2}\,).$$

In der zweiten Iteration ergeben sich die Koeffizienten

$$\begin{aligned} a_{2j}^{(2)} &= \sum_{i\in\mathbb{Z}} a_i^{(1)}\, a_{2j-2i} = \sum_{i'\in\mathbb{Z}} a_{j-i'}^{(1)}\; a_{2i'} = a_0\, a_j^{(1)} + a_2\, a_{j-1}^{(1)}, \\ a_{2j+1}^{(2)} &= \sum_{i\in\mathbb{Z}} a_i^{(1)}\, a_{2j+1-2i} = \sum_{i'\in\mathbb{Z}} a_{j_i'}^{(1)}\, a_{2i'+1} = a_1\, a_j^{(1)} + a_3\, a_{j-1}^{(1)}, \end{aligned}$$

und damit

$$\begin{aligned}
a_0^{(2)} &= a_0^2 = \tfrac{1}{16}, & a_1^{(2)} &= a_1 a_0 = \tfrac{3}{16}, & a_2^{(2)} &= a_0 a_1 + a_2 a_0 = \tfrac{6}{16},\\
a_3^{(2)} &= a_1^2 + a_3 a_0 = \tfrac{10}{16}, & a_4^{(2)} &= a_0 a_2 + a_2 a_1 = \tfrac{12}{16}, & a_5^{(2)} &= a_1 a_2 + a_3 a_1 = \tfrac{12}{16},\\
a_6^{(2)} &= a_0 a_3 + a_2^2 = \tfrac{10}{16}, & a_7^{(2)} &= a_1 a_3 + a_3 a_2 = \tfrac{6}{16}, & a_8^{(2)} &= a_2 a_3 = \tfrac{3}{16},\\
a_9^{(2)} &= a_3^2 = \tfrac{1}{16}.
\end{aligned}$$



Erste und zweite Iteration des Chaikins Algorithmus zur Berechnung des kardinalen quadratischen B-Splines.

Beweis von Lemma 3.18. Wir setzen $b_j^{(k)} := N_m(\frac{j}{2^k})$. Sei nun

$$\begin{aligned}
f_k(x) &:= \sum_{j\in\mathbb{Z}} N_m\Big(\frac{j}{2^k}\Big)\, N_m(2^k x - j) = \sum_{j\in\mathbb{Z}} b_j^{(k)}\, N_m(2^k x - j),\\
g_k(x) &:= \sum_{j\in\mathbb{Z}} N_m\Big(\frac{j}{2^k}\Big)\, N_2(2^k x - j + 1 - \frac{m}{2}\,) = \sum_{j\in\mathbb{Z}} b_j^{(k)}\, N_2(2^k x - j + 1 - \frac{m}{2}\,)
\end{aligned}$$

Dann folgt aus Lemma 3.17 mit $h = \frac{1}{2^k}$

$$\|f_k - N_m\|_\infty \le m\, w(N_m, \frac{1}{2^k}),$$

und in analoger Weise

$$\|g_k - N_m\|_\infty \le (1 + \frac{m}{2})\, w(N_m, \frac{1}{2^k})\ .$$

Nach Satz 3.14 erhalten wir

$$\begin{aligned}
N_m(x) &= \sum_{j\in\mathbb{Z}} a_j N_m(2x - j) = \sum_{j\in\mathbb{Z}} a_j^{(1)} \sum_{l\in\mathbb{Z}} a_l N_m(4x - 2j - l)\\
&= \sum_{l'\in\mathbb{Z}} \underbrace{\Big(\sum_{j\in\mathbb{Z}} a_j^{(1)}\ a_{l'-2j}\Big)}_{a_{l'}^{(2)}} N_m(4x - l') = \ldots = \sum_{j\in\mathbb{Z}} a_j^{(k)} N_m(2^k x - j)
\end{aligned}$$

und damit

$$\begin{aligned}\|N_m - f_k\|_\infty &= \|\sum_{j\in\mathbb{Z}} (a_j^{(k)} N_m(2^k\cdot - j) - b_j^{(k)} N_m(2^k\cdot - j))\|_\infty \\ &= \|\sum_{j\in\mathbb{Z}} (a_j^{(k)} - b_j^{(k)})\, N_m(2^k\cdot - j)\|_\infty.\end{aligned}$$

Wegen Lemma 3.16 existiert eine Konstante $K_m < \infty$ mit

$$\sup_{j\in\mathbb{Z}} |a_j^{(k)} - b_j^{(k)}| \le K_m \|\sum_{j\in\mathbb{Z}} (a_j^{(k)} - b_j^{(k)})\, N_m(2^k\cdot - j)\|_\infty.$$

Andererseits ist wegen $\sum_{j\in\mathbb{Z}} N_2(2^k x + 1 - \frac{m}{2} - j) = 1$ für alle $x \in \mathbb{R}$,

$$\begin{aligned}&\|g_k - \sum_{j\in\mathbb{Z}} a_j^{(k)}\, N_2(2^k\cdot - j + 1 - \frac{m}{2}\|_\infty \\ =\ &\|\sum_{j\in\mathbb{Z}} (a_j^{(k)} - b_j^{(k)})\, N_2(2^k\cdot - j + 1 - \frac{m}{2})\|_\infty \\ \le\ &\sup_{x\in\mathbb{R}} \sum_{j\in\mathbb{Z}} |a_j^{(k)} - b_j^{(k)}|\, N_2(2^k x - j + 1 - \frac{m}{2}) \le \sup_j |a_j^{(k)} - b_j^{(k)}|.\end{aligned}$$

Damit ergibt sich

$$\|g_k - \sum_{j\in\mathbb{Z}} a_j^{(k)}\, N_2(2^k\cdot - j + 1 - \frac{m}{2})\|_\infty \le \sup_j |a_j^{(k)} - b_j^{(k)}| \le K_m \|N_m - f_k\|_\infty,$$

und wir erhalten

$$\begin{aligned}&\|N_m - \sum_{j\in\mathbb{Z}} a_j^{(k)} N_2(2^k\cdot - j + 1 - \frac{m}{2})\|_\infty \\ \le\ &\|g_k - \sum_{j\in\mathbb{Z}} a_j^{(k)}\, N_2(2^k\cdot - j + 1 - \frac{m}{2})\|_\infty + \|N_m - g_k\|_\infty \\ \le\ &K_m \|N_m - f_k\|_\infty + \|N_m - g_k\|_\infty \\ \le\ &K_m\, m\, w(N_m, \frac{1}{2^k}) + (1 + \frac{m}{2})\, w(N_m, \frac{1}{2^k}) \overset{k\to\infty}{\longrightarrow} 0.\end{aligned}$$

□

Beweis von Satz 3.15. Sei wieder $a_j^{(1)} = a_j$ mit a_j wie in Satz 3.15 angegeben und

$$a_j^{(k)} = \sum_{i\in\mathbb{Z}} a_i^{(k-1)}\, a_{j-2i}, \qquad k = 2, 3, \ldots.$$

Weiterhin setzen wir $(\mathbf{d}_j^0) := (\mathbf{d}_j)$ und

$$\mathbf{d}_j^{(k)} := \sum_{i\in\mathbb{Z}} \mathbf{d}_i^{(k-1)}\, a_{j-2i}, \qquad k = 1, 2, \ldots.$$

Wir zeigen durch vollständige Induktion, dass für $k \geq 1$

$$\mathbf{d}_j^{(k)} = \sum_{i\in\mathbb{Z}} \mathbf{d}_i a_{j-2^k i}^{(k)}$$

gilt.
Für $k = 1$ ist die Behauptung richtig.
Sei nun $\mathbf{d}_j^{(k)} = \sum_{i\in\mathbb{Z}} \mathbf{d}_i\, a_{j-2^k i}^{(k)}$. Dann folgt mit der Substitution $l' = l - 2^k i$

$$\begin{aligned}
\mathbf{d}_j^{(k+1)} &= \sum_{l\in\mathbb{Z}} \mathbf{d}_l^{(k)}\ a_{j-2l} = \sum_{l\in\mathbb{Z}} (\sum_{i\in\mathbb{Z}}\ \mathbf{d}_i\, a_{l-2^k i}^{(k)})\, a_{j-2l} \\
&= \sum_{i\in\mathbb{Z}} \mathbf{d}_i \sum_{l\in\mathbb{Z}} a_{l-2^k i}^{(k)}\ a_{j-2l} \\
&= \sum_{i\in\mathbb{Z}}\ \mathbf{d}_i \sum_{l'\in\mathbb{Z}}\ a_{l'}^{(k)}\ a_{j-2l'-2^{k+1}i} = \sum_{i\in\mathbb{Z}} \mathbf{d}_i\, a_{j-2^{k+1}i}^{(k+1)}.
\end{aligned}$$

Nun erhalten wir wegen $N_m(t-i) = 0$ für $t \notin [i,\, i+m]$,

$$\begin{aligned}
&\| \sum_{i\in\mathbb{Z}}\ \mathbf{d}_i N_m(\cdot - i) - \sum_{j\in\mathbb{Z}} \mathbf{d}_j^k N_2(2^k \cdot - j + 1 - \frac{m}{2})\ \|_\infty \\
= &\sum_{i\in\mathbb{Z}} \mathbf{d}_i \Big(N_m(\cdot - i) - \sum_{j\in\mathbb{Z}} a_{j-2^k i}^{(k)}\ N_2(2^k \cdot - j + 1 - \frac{m}{2}) \Big) \|_\infty \\
\leq &\underbrace{\sup_{k\in\mathbb{Z}} \sum_{i=k}^{k+m} \|\mathbf{d}_i\|_\infty}_{<\infty}\ \|N_m(\cdot - i) - \sum_{j'\in\mathbb{Z}}\ a_{j'}^{(k)} N_2(2^k(\cdot - i) - j' + 1 - \frac{m}{2})\|_\infty.
\end{aligned}$$

Da die Kontrollpunkte $\mathbf{d}_i$ eine beschränkte Norm haben, konvergiert dieser Term nach Lemma 3.18 für $k \to \infty$ gegen Null. □

Der Subdivision-Algorithmus ist die effizienteste Möglichkeit der Berechnung von B-Spline-Kurven.

Bemerkung: Wegen

$$\sum_{k\in\mathbb{Z}} (k + \frac{m}{2})\, N_m(t-k) = t$$

für all $t \in \mathbb{R}$ ist für skalare Werte d_k das Kontrollpolygon durch die Kontrollpunkte $(k + \frac{m}{2},\, d_k)^T$ bestimmt.

4 Numerische Integration

4.1 Einführung

Bei der Berechnung von bestimmten Integralen ist man im Allgemeinen auf numerische Näherungen angewiesen, wenn keine analytische, geschlossene Darstellung dieser Integrale möglich ist.

Definition 4.1. *Unter einer* **Quadraturformel** Q *zur Berechnung des bestimmten Integrals* $I(f) := \int_a^b f(x)\,dx$ *für* $f \in C[a,b]$ *verstehen wir die Summe*

$$Q(f) := \sum_{j=0}^{n} a_j f(x_j), \qquad a_j \in \mathbb{R}, x_j \in [a,b],$$

mit den **Knoten** $x_0, \ldots, x_n$ *und den* **Gewichten** $a_0, \ldots, a_n$. *Die Differenz*

$$R(f) := \int_a^b f(x)\,dx - Q(f)$$

bezeichnet man als **Quadraturfehler**. *Wenn* $R(p) = 0$ *für alle Polynome* $p \in \Pi_d$ *gilt, so heißt* Q **exakt** *auf der Menge der Polynome* Π_d *vom Höchstgrad* d.

Beispiel:

$n = 0:$	$Q(f) = f(a)\,(b-a)$	Rechteckregel
	$Q(f) = f(\frac{a+b}{2})\,(b-a)$	Mittelpunktregel
$n = 1:$	$Q(f) = (f(a) + f(b))\,\frac{b-a}{2}$	Trapezregel

Idee: Zur Berechnung von $\int_A^B f(x)\,dx$ wende man eine einfache Quadraturformel auf möglichst vielen Teilintervallen von $[A,B]$ an:
Wähle eine Zerlegung $A = z_0 < z_1 < z_2 < \ldots < z_n = B$ und wende die Quadraturformel auf jedem Teilintervall $[z_j, z_{j+1}], j = 0, \ldots, n-1$, an:

a) zusammengesetzte Rechteckregel: $I(f) \approx \sum_{j=0}^{n-1} f(z_j)\,(z_{j+1} - z_j)$

b) zusammengesetzte Mittelpunktregel: $I(f) \approx \sum_{j=0}^{n-1} f(\frac{z_j + z_{j+1}}{2})\,(z_{j+1} - z_j)$

c) zusammengesetzte Trapezregel:

$$\begin{aligned} I(f) &\approx \sum_{j=0}^{n-1} \frac{1}{2}(f(z_j) + f(z_{j+1}))\,(z_{j+1} - z_j) \\ &= f(z_0)\,(\frac{z_1 - z_0}{2}) + \sum_{j=1}^{n-1} f(z_j)\,(\frac{z_{j+1} - z_{j-1}}{2}) + f(z_n)\,(\frac{z_n - z_{n-1}}{2}). \end{aligned}$$

Wählt man die Zerlegung des Intervalls $[A, B]$ äquidistant, d.h. $z_j = A + j\,h$, mit $h = \frac{B-A}{n}, j = 0, \ldots, n$, so folgt z.B. für die zusammengesetzte Mittelpunktregel:

$$I(f) \approx \sum_{j-0}^{n-1} f(\frac{A + hj + A + h(j+1)}{2})\, h = h \sum_{j=0}^{n-1} f(A + \frac{2j+1}{2}h).$$

Um verschiedene Quadraturformeln miteinander vergleichen zu können, benötigen wir eine Darstellung des Quadraturfehlers $R(f) = I(f) - Q(f)$. Dazu betrachten wir im Folgenden ein "kleines" Teilintervall $[a, b]$.

Satz 4.2. *Der Fehler $R(f)$ einer $(n+1)-$punktigen Quadraturformel vom Exaktheitsgrad d*

$$\int_a^b f(x)\, dx = \sum_{j=0}^{n} a_j f(x_j) + R(f), \quad a \le x_0 < \ldots < x_n \le b,$$

hat für $f \in C^{m+1}[a, b]$ und $0 \le m \le d$ die Darstellung

$$R(f) = \int_a^b f^{(m+1)}(t) G_m(t)\, dt,$$

wobei

$$G_m(t) := \frac{1}{m!} R_x[(x-t)_+^m] := \frac{1}{m!}(\int_a^b (x-t)_+^m dx - \sum_{j=0}^{n} a_j (x_j - t)_+^m).$$

Hierbei bedeutet $R_x((x-t)_+^m)$, dass R auf die Funktion $(x-t)_+^m$ als Funktion in x anzuwenden ist. G_m heißt **Peano-Kern** *von R.*

Beweis. Sei $f \in C^{m+1}[a, b]$ gegeben. Eine Taylor-Entwicklung von f in a ergibt

$$f(x) = \underbrace{\sum_{j=0}^{m} \frac{1}{j!} f^{(j)}(a)\,(x-a)^j}_{\text{Polynom vom Grad } m \le d} + r_m(x)$$

mit dem Restglied

$$r_m(x) = \frac{1}{m!} \int_a^x f^{(m+1)}(t)(x-t)^m\, dt = \frac{1}{m!} \int_a^b f^{(m+1)}(t)(x-t)_+^m\, dt.$$

Wegen $R(p) = 0$ für alle $p \in \Pi_d$ und $m \le d$ folgt

$$R(f) = R(r_m).$$

Nun ist

$$\begin{aligned} R(r_m) &= \int_a^b r_m(x)\,dx - \sum_{j=0}^{n} a_j r_m(x_j) \\ &= \frac{1}{m!}\int_a^b \int_a^b f^{(m+1)}(t)(x-t)_+^m\,dtdx - \sum_{j=0}^{n}\frac{a_j}{m!}\int_a^b f^{(m+1)}(t)(x_j-t)_+^m\,dt \\ &= \frac{1}{m!}\int_a^b f^{(m+1)}(t)\cdot\left(\int_a^b (x-t)_+^m dx - \sum_{j=0}^{n} a_j(x_j-t)_+^m\right)dt \\ &= \frac{1}{m!}\int_a^b f^{(m+1)}(t)\cdot R_x[(x-t)_+^m]\,dt. \end{aligned}$$

□

Folgerung 4.3. *Ist $f \in C^{m+1}[a,b]$, so ist*

$$|R(f)| \le \max_{x\in[a,b]} |f^{(m+1)}(x)| \int_a^b |G_m(t)|\,dt.$$

Beispiel: Die Mittelpunktregel: $\int_a^b f(x)dx \approx (b-a)f(\frac{a+b}{2}) = Q(f)$ hat den Exaktheitsgrad $d=1$, denn für $p(x) = mx+n$ ist

$$\begin{aligned} I(p) = \int_a^b (mx+n)dx &= \left[\frac{mx^2}{2}+nx\right]_a^b = m\frac{(b^2-a^2)}{2} + n(b-a) \\ &= (b-a)(m\Big(\frac{b+a}{2}\Big)+n) = (b-a)p(\frac{a+b}{2}) = Q(p). \end{aligned}$$

Wähle $m=0$. Dann erhalten wir den Peano-Kern

$$\begin{aligned} G_0(t) &= \int_a^b (x-t)_+^0\,dx - (b-a)(\frac{a+b}{2}-t)_+^0 \\ &= \begin{cases} (b-t)-(b-a) = a-t, & a \le t \le \frac{a+b}{2} \\ (b-t)-0, & \frac{a+b}{2} < t \le b \end{cases}. \end{aligned}$$

Für $f \in C^1[a,b]$ folgt

$$|R(f)| \le \int_a^b |f'(t)||G_0(t)|\,dt \le \max_{x\in[a,b]} |f'(x)| \underbrace{\int_a^b |G_0(t)|\,dt}_{\frac{(b-a)^2}{4}}.$$

Wähle nun $m = 1$:

$$\begin{aligned} G_1(t) &= \int_a^b (x-t)_+^1 dx - (b-a)(\frac{a+b}{2} - t)_+^1 \\ &= \begin{cases} \int_t^b (x-t)dx - (b-a)(\frac{a+b}{2} - t), & a \le t \le \frac{a+b}{2} \\ \int_t^b (x-t)dx, & \frac{a+b}{2} < t \le b \end{cases} \\ &= \begin{cases} \frac{(a-t)^2}{2}, & a \le t \le \frac{a+b}{2} \\ \frac{(b-t)^2}{2}, & \frac{a+b}{2} < t \le b \end{cases} \end{aligned}$$

da

$$\int_t^b (x-t)dx = \left[\frac{x^2}{2} - xt\right]_t^b = \frac{b^2}{2} - bt - \frac{t^2}{2} + t^2 = \frac{b^2}{2} - bt + \frac{t^2}{2} = \frac{(b-t)^2}{2}.$$

Dann gilt aus Symmetriegründen

$$\int_a^b |G_1(t)|\, dt = 2\int_a^{\frac{a+b}{2}} \frac{(t-a)^2}{2} dt = 2\left[\frac{(t-a)^3}{6}\right]_a^{\frac{a+b}{2}} = 2\frac{(b-a)^3}{48} = \frac{(b-a)^3}{24},$$

und damit folgt für $f \in C^2[a,b]$ also

$$|R(f)| \le \max_{x\in[a,b]} |f''(x)| \int_a^b |G_1(t)|\, dt = \frac{(b-a)^3}{24} \max_{x\in[a,b]} |f''(x)|.$$

□

Für die zusammengesetzte Mittelpunktregel folgt daraus für $f \in C^2[A,B]$

$$\begin{aligned} |I(f) - Q(f)| &= |\int_A^B f(x)\, dx - \sum_{j=0}^{n-1} f(A + \frac{2j+1}{2}h)\, h| \\ &\le \sum_{j=0}^{n-1} |\int_{A+jh}^{A+(j+1)h} f(x)\, dx - f(A + \frac{2j+1}{2}h)\, h| \\ &\le \sum_{j=0}^{n-1} \max_{x\in[A+jh, A+(j+1)h]} |f''(x)|\, \frac{h^3}{24} \\ &\le \max_{x\in[A,B]} |f''(x)|\, n\, \frac{(B-A)^3}{24n^3} = \mathcal{O}(\frac{1}{n^2}). \end{aligned}$$

4.2 Interpolatorische Quadraturformeln

Definition 4.4. *Eine $(n+1)-$punktige Quadraturformel $Q(f) := \sum_{j=0}^n a_j f(x_j)$ heißt* **interpolatorisch**, *wenn Q auf der Menge der Polynome Π_n exakt ist.*

Satz 4.5. *Zu beliebig vorgegebenen Stützstellen $a \le x_0 < x_1 < \ldots < x_n \le b$ existiert genau eine interpolatorische Quadraturformel $Q(f) = \sum_{j=0}^{n} a_j f(x_j)$. Ihre Gewichte erhält man durch Integration der Lagrange-Grundpolynome $l_j, j = 0, \ldots, n$, d.h.*

$$a_j = \int_a^b l_j(x)\, dx \quad mit \quad l_j(x) := \prod_{\substack{k=0 \\ k \neq j}}^{n} \frac{(x - x_k)}{(x_j - x_k)}.$$

(Idee: Interpoliere f durch $p_n \in \Pi_n$ und integriere dann p_n exakt statt f.)

Beweis. 1) Existenz: Da jedes Polynom $p \in \Pi_n$ eindeutig in der Lagrangeform $p(x) = \sum_{j=0}^{n} p(x_j) l_j(x)$ darstellbar ist (siehe Satz 1.2, Satz 1.6), folgt

$$\int_a^b p(x)\, dx = \int_a^b \sum_{j=0}^{n} p(x_j) l_j(x)\, dx = \sum_{j=0}^{n} p(x_j) \underbrace{\int_a^b l_j(x) dx}_{=a_j} = Q(p).$$

2) Eindeutigkeit: Seien $Q(f)$ und $\tilde{Q}(f) = \sum_{j=0}^{n} b_j f(x_j)$ zu den Stützstellen $a \le x_0 < \ldots < x_n \le b$ interpolatorisch. Dann gilt für alle $p \in \Pi_n$: $Q(p) - \tilde{Q}(p) = 0$. Das bedeutet $\sum_{j=0}^{n} (a_j - b_j) p(x_j) = 0$ für alle $p \in \Pi_n$. Wählen wir $p_k(x) := x^k, k = 0, \ldots, n$, so erhalten wir das LGS

$$\begin{bmatrix} 1 & 1 & \ldots & 1 \\ x_0 & x_1 & \ldots & x_n \\ x_0^2 & x_1^2 & \ldots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_0^n & x_1^n & \ldots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 - b_0 \\ a_1 - b_1 \\ \vdots \\ a_n - b_n \end{bmatrix} = \mathbf{0}.$$

Da die Koeffizientenmatrix invertierbar ist, folgt $a_j = b_j,\ j = 0, \ldots, n$. □

Bemerkung: Für äquidistante Stützstellen $x_j = a + h\, j,\ j = 0, \ldots, n,\ h = \frac{b-a}{n}$, heißen die interpolatorischen Quadraturformeln **Newton-Cotes-Formeln**. Wir erhalten in diesem Fall mit der Substitution $t = \frac{x-a}{h}$

$$\begin{aligned} a_j &= a_{j,n} = \int_a^b \prod_{\substack{k=0 \\ k \neq j}}^{n} \frac{x - x_k}{x_j - x_k}\, dx = \int_a^b \prod_{\substack{k=0 \\ k \neq j}}^{n} \left(\frac{x - a - hk}{hj - hk}\right) dx \\ &= \int_0^n \prod_{\substack{k=0 \\ k \neq j}}^{n} \frac{h(t-k)}{h(j-k)} h\, dt = \frac{h}{\prod_{\substack{k=0 \\ k \neq j}}^{n} (j-k)} \int_0^n \prod_{\substack{k=0 \\ k \neq j}}^{n} (t-k)\, dt \\ &= \frac{h(-1)^{n-j}}{j!(n-j)!} \int_0^n \prod_{\substack{k=0 \\ k \neq j}}^{n} (t-k)\, dt. \end{aligned} \tag{4.1}$$

Beispiel: Für $n = 1$ erhält man die Trapezregel: $x_0 = a, x_1 = b$,

$$Q(f) = a_0 f(x_0) + a_1 f(x_1) = a_0 f(a) + a_1 f(b)$$

mit

$$\begin{aligned} a_0 &= \int_a^b l_0(x)\, dx = \int_a^b \frac{(x-b)}{(a-b)}\, dx = \left[\frac{\frac{x^2}{2} - bx}{a-b}\right]_a^b = \frac{b-a}{2}, \\ a_1 &= \int_a^b l_1(x)\, dx = \frac{b-a}{2}. \end{aligned}$$

□

Satz 4.6. *Der Quadraturfehler für die $(n+1)-$punktige interpolatorische Quadraturformel hat die Form*

$$R(f) = \int_a^b f(x)\, dx - \int_a^b p_n(x)\, dx = \int_a^b f[x_0, \ldots, x_n, x]\, w_{n+1}(x)\, dx$$

mit $w_{n+1}(x) = (x - x_0)(x - x_1) \cdot \ldots \cdot (x - x_n)$, wobei $p_n \in \Pi_n$ das Interpolationspolynom von f zu den Stützstellen $x_0 < x_1 < \ldots < x_n$ ist.

Beweis. Für die Polynom-Interpolation war nach Satz 1.16:

$$f(x) - p_n(x) = f[x_0, \ldots, x_n, x] w_{n+1}(x),$$

woraus die Behauptung folgt. □

Beispiel:

1. Für $n = 1$ folgt wegen $(x-a)(x-b) \leq 0 \quad \forall\, x \in [a,b]$

$$\begin{aligned} |R(f)| &= \left| \int_a^b f[a,b,x](x-a)(x-b)\, dx \right| \\ &\leq \max_{\eta \in [a,b]} |f[a,b,\eta]| \int_a^b |(x-a)(x-b)|\, dx = \max_{\eta \in [a,b]} f[a,b,\eta]\, \frac{(b-a)^3}{6}. \end{aligned}$$

   Für $f \in C^2[a,b]$ war $f[a,b,\eta] = \frac{1}{2} f''(\xi)$ für eine Zwischenstelle $\xi \in [a,b]$ (siehe Folgerung 1.17). Damit erhalten wir $|R(f)| \leq \frac{(b-a)^3}{12} \max\limits_{\xi \in [a,b]} |f''(\xi)|$.

2. Für $n = 2$ nennt man die entsprechende Newton-Cotes-Formel **Simpson-Regel** oder **Keplersche Fassregel.** Wir erhalten

$$Q(f) = a_0 f(a) + a_1 f(\frac{a+b}{2}) + a_2 f(b)$$

und mit (4.1)

$$\begin{aligned}
a_0 &= \int_a^b l_0(x)\,dx = \frac{(\frac{b-a}{2})\cdot(-1)^{2-0}}{0!(2-0)!}\int_0^2 (t-1)(t-2)\,dt \\
&= \frac{b-a}{4}\left[\frac{t^3}{3}-\frac{3t^2}{2}+2t\right]_0^2 = (\frac{b-a}{4})\,(\frac{8}{3}-6+4) = \frac{b-a}{6}, \\
a_1 &= \int_a^b l_1(x)\,dx = \frac{(\frac{b-a}{2})\cdot(-1)}{1!1!}\int_0^2 t\,(t-2)\,dt = \frac{4(b-a)}{6}, \\
a_2 &= \int_a^b l_2(x)\,dx = \frac{b-a}{6}.
\end{aligned}$$

Also folgt

$$Q(f) = \frac{b-a}{6}[f(a)+4f(\frac{a+b}{2})+f(b)].$$

Quadraturfehler
Die Simpson-Regel ist sogar für Polynome vom Grad 3 exakt. Wir nutzen die Abschätzung aus Folgerung 4.3 (bzw. Satz 4.2) mit $m = d = 3$ und erhalten

$$|R(f)| \le \max_{x\in[a,b]} |f^{(4)}(x)| \int_a^b |G_3(t)|\,dt$$

für $f \in C^4[a,b]$, wobei für $t \in [a,b]$,

$$\begin{aligned}
G_3(t) &= \frac{1}{3!}\left\{\int_a^b (x-t)_+^3\,dx - (\frac{b-a}{6})\left[(a-t)_+^3 + 4(\frac{a+b}{2}-t)_+^3 + (b-t)_+^3\right]\right\} \\
&= \frac{1}{6}\left\{\int_t^b (x-t)^3\,dx - \frac{b-a}{6}\left[4(\frac{a+b}{2}-t)_+^3 + (b-t)^3\right]\right\} \\
&= \begin{cases} \frac{1}{6}\left\{\frac{(x-t)^4}{4}|_t^b - \frac{b-a}{6}\left[4(\frac{a+b}{2}-t)^3+(b-t)^3\right]\right\}, & a \le t \le \frac{a+b}{2}, \\ \frac{1}{6}\left\{\frac{(b-t)^4}{4} - \frac{b-a}{6}\cdot(b-t)^3\right\}, & \frac{a+b}{2} < t \le b, \end{cases} \\
&= \begin{cases} \frac{(t-a)^3}{72}(3t-a-2b), & a \le t \le \frac{a+b}{2}, \\ \frac{(b-t)^3}{72}(2a+b-3t), & \frac{a+b}{2} < t \le b. \end{cases}
\end{aligned}$$

Also folgt insgesamt

$$\int_a^b |G_3(t)|\,dt = 2\int_a^{\frac{a+b}{2}} \frac{(t-a)^3(a+2b-3t)}{72}\,dt = \frac{1}{2880}(b-a)^5.$$

Wir erhalten für $f \in C^4[a,b]$:

$$|R(f)| \le \frac{1}{2880}(b-a)^5 \max_{x\in[a,b]} |f^{(4)}(x)|.$$

Newton-Cotes-Formeln höherer Ordnung werden kaum benutzt. Besser ist es, Formeln niedriger Ordnung zusammenzusetzen.

Zusammengesetzte Simpson-Regel
Für die Zerlegung $A = z_0 < z_1 < \ldots < z_n = B$ in äquidistante Stützstellen $z_j = A + j\,h,\ j = 0, \ldots, n,\ h = \frac{B-A}{n}$, erhalten wir

$$\begin{aligned}\int_a^b f(x)\,dx &\approx \sum_{j=0}^{n-1} \frac{h}{6}\left[f(z_j) + 4f(\frac{z_j + z_{j+1}}{2}) + f(z_{j+1})\right] \\ &= \frac{h}{3}\left[\frac{f(A)}{2} + 2\sum_{j=0}^{n-1} f(\frac{z_j + z_{j+1}}{2}) + \sum_{j=1}^{n-1} f(z_j) + \frac{f(B)}{2}\right].\end{aligned}$$

Fehler für zusammengesetzte Regel:

$$\begin{aligned}|R^z(f)| &\le \sum_{j=0}^{n-1} \frac{1}{2880}\,h^5 \max_{\xi\in[z_j,z_{j+1}]} |f^{(4)}(\xi)| \\ &\le \max_{x\in[A,B]} |f^{(4)}(x)|\,\frac{(B-A)^5}{n^5}\,\frac{n}{2880} = \mathcal{O}\left(\frac{1}{n^4}\right).\end{aligned}$$

4.3 Das Romberg-Verfahren

Teilt man das Intervall $[A, B]$ in n Teilintervalle gleicher Länge und wendet die Trapezregel an, so erhält man mit $h = \frac{B-A}{n}$

$$T(h) := h\left\{\frac{f(A)}{2} + \sum_{j=1}^{n-1} f(A + jh) + \frac{f(B)}{2}\right\}.$$

Wir wollen die zusammengesetzte Trapezregel auf einer Folge von Gittern iterativ anwenden und daraus ein neues verbessertes Quadraturverfahren konstruieren. Dazu untersuchen wir zunächst das asymptotische Verhalten von $T(h)$ für $h \to 0$.

Definition 4.7. *Die* **Bernoulli-Polynome** $B_j \in \Pi_j, j \in \mathbb{N}_0$, *sind rekursiv definiert durch*

(i) $B_0(x) := 1$,

(ii) $B_j'(x) = j\,B_{j-1}(x), \quad j = 1, 2, \ldots,$

(iii) $\int_0^1 B_j(x)\,dx = 0, \quad j = 1, 2, \ldots.$

Es ist also $B_1(x) = x - \frac{1}{2},\ B_2(x) = x^2 - x + \frac{1}{6},\ B_3(x) = x^3 - \frac{3}{2}x^2 + \frac{1}{2}x$. usw.

Satz 4.8. *Die Bernoulli-Polynome haben die folgenden Eigenschaften:*

(a) $B_j(0) = B_j(1), \quad j = 2, 3, 4, \ldots$

(b) $B_j(x) = (-1)^j B_j(1-x), \quad j = 0, 1, \ldots$

(c) $B_{2j+1}(0) = B_{2j+1}(1) = 0, \quad j = 1, 2, \ldots$

(d) $B_{2j+1}(\frac{1}{2}) = 0, \quad j = 0, 1, \ldots.$

Die Zahlen $B_j := B_j(0),\ j \in \mathbb{N}_0$, *die auch in der Potenzreihe*

$$\frac{z}{e^z - 1} = \sum_{j=0}^{\infty} \frac{B_j}{j!} z^j, \quad |z| < 2\pi,$$

auftreten, heißen **Bernoulli-Zahlen**.

Beweis. 1) Nach Definition der Bernoulli-Polynome B_j folgt für $j \geq 2$

$$B_j(1) - B_j(0) = \int_0^1 B_j'(x)\, dx \overset{(ii)}{=} j \int_0^1 B_{j-1}(x)\, dx \overset{(iii)}{=} 0.$$

2) Setze $C_j(x) := (-1)^j B_j(1-x),\ j \in \mathbb{N}_0$. Dann folgt $C_0(x) = B_0(1-x) = 1$. Weiter gilt für $j = 1, 2, \ldots$

$$\begin{aligned} C_j'(x) &= (-1)^j [B_j(1-x)]' = (-1)^{j-1} B_j'(1-x) = (-1)^{j-1}(j-1) B_{j-1}(1-x) \\ &= (j-1) C_{j-1}(x), \end{aligned}$$

und

$$\int_0^1 C_j(x)\, dx = (-1)^j \int_0^1 B_j(1-x)\, dx = (-1)^j \int_0^1 B_j(x)\, dx = 0.$$

Also erfüllt $C_j(x)$ die Definition der Bernoulli-Polynome, d.h., $C_j(x) = B_j(x),\ j \in \mathbb{N}_0$.
3) Aus (a) und (b) folgt

$$B_{2j+1}(1) = (-1)^{2j+1} B_{2j+1}(0) = -B_{2j+1}(0)$$

und $B_{2j+1}(1) = B_{2j+1}(0)$, also

$$B_{2j+1}(1) = B_{2j+1}(0) = 0, \qquad j = 1, 2, \ldots.$$

4) Aus (b) folgt $B_{2j+1}(\frac{1}{2}) = (-1)^{2j+1} B_{2j+1}(\frac{1}{2}) = -B_{2j+1}(\frac{1}{2})$ und damit die Behauptung. □

Lemma 4.9. *Das Bernoulli-Polynom $B_{2j+1}(x)$ besitzt für $j \in \mathbb{N}$ im Intervall $[0,1]$ genau die einfachen Nullstellen $0, 1, \frac{1}{2}$.*

Beweis. Wegen Satz 4.8 (c),(d) sind $0, 1, \frac{1}{2}$ Nullstellen von B_{2j+1}. Wir zeigen mittels vollständiger Induktion, dass B_{2j+1} in $[0,1]$ keine weitere Nullstelle hat. Für $j = 1$ folgt: $B_3(x) = x^3 - \frac{3}{2}x^2 + \frac{1}{2}x$ hat genau die Nullstellen $0, \frac{1}{2}, 1$. Besitzt nun B_{2j+1} nur die einfachen Nullstellen $0, \frac{1}{2}, 1$ in $[0,1]$, so hat B_{2j+2} lokale Extrema an diesen Stellen und nur je eine einfache Nullstelle in $(0, \frac{1}{2})$ und $(\frac{1}{2}, 1)$. Also besitzt B_{2j+3} nur 2 lokale Extrema in $[0,1]$ und wieder nur die Nullstellen $0, \frac{1}{2}, 1$. □

Lemma 4.10. *Für die geraden Bernoulli-Polynome B_{2j}, $j \in \mathbb{N}$, gilt für alle $x \in (0,1)$*

$$B_{2j}(x) \neq B_{2j}(0)\ (= B_{2j}(1))$$

und $B_{2j}(0) \neq 0$.

Beweis. Aus Lemma 4.9 folgt, dass B_{2j} für $j \geq 1$ in $[0,1]$ nur je eine einfache Nullstelle in $(0, \frac{1}{2})$ und $(\frac{1}{2}, 1)$ besitzt. Betrachte $\tilde{B}_{2j}(x) := B_{2j}(x) - B_{2j}(0)$. Dann besitzt $\tilde{B}_{2j}(x)$ die Nullstellen $0, 1$. Hätte B_{2j} eine weitere Nullstelle in $(0,1)$, dann folgt aus dem Satz von Rolle, dass B_{2j-1} zwei Nullstellen in $(0,1)$ besitzt im Widerspruch zu Lemma 4.9. □

Mit Hilfe der Bernoulli-Polynome folgt nun

Satz 4.11 (Euler-Maclaurinsche-Summenformel)**.** *Für $f \in C^{2m+1}[0,n]$ ist*

$$\int_0^n f(x)\,dx = [\frac{f(0)}{2} + \sum_{j=1}^{n-1} f(j) + \frac{f(n)}{2}] - \sum_{k=1}^{m} \frac{B_{2k}}{(2k)!}[f^{(2k-1)}(n) - f^{(2k-1)}(0)] + R_m$$

mit dem Restglied

$$R_m := -\frac{1}{(2m+1)!} \int_0^n \overline{B}_{2m+1}(x) f^{(2m+1)}(x)\,dx$$

wobei $\overline{B}_j : \mathbb{R} \to \mathbb{R}$ die periodisch fortgesetzten Bernoulli-Polynome sind, d.h.

$$\overline{B}_j(x) := B_j(x - \lfloor x \rfloor) \quad \textit{mit } \lfloor x \rfloor := \max\{a \in \mathbb{Z} : a \leq x\}.$$

Beweis. Wegen $B_j(0) = B_j(1)$ ist $\overline{B}_j$ für $j \geq 2$ eine stetige Funktion. Betrachte nun zunächst $m = 0$. Die Funktion $\overline{B}_1$ ist stückweise aus $B_1(x) = x - \frac{1}{2}$ zusammengesetzt. Mittels partieller Integration folgt

$$\begin{aligned} \int_0^1 \overline{B}_1(x) f'(x) dx &= \int_0^1 B_1(x) f'(x)\,dx = [B_1(x) f(x)]_0^1 - \int_0^1 B_0(x) f(x)\,dx \\ &= \frac{1}{2}(f(1) + f(0)) - \int_0^1 f(x)\,dx. \end{aligned}$$

Analog findet man für $i = 0, 1, \ldots, n-1$

$$\begin{aligned}\int_i^{i+1} \overline{B}_1(x) f'(x)\, dx &= \int_0^1 B_1(x) f'(x+i)\, dx \\ &= \frac{1}{2}(f(i+1)+f(i)) - \int_i^{i+1} f(x)\, dx.\end{aligned}$$

Daraus folgt

$$-R_0 := \int_0^n \overline{B}_1(x) f'(x)\, dx = \sum_{j=0}^{n-1} \frac{1}{2}(f(j+1)+f(j)) - \int_0^n f(x)\, dx$$

bzw.

$$\int_0^n f(x)\, dx = \underbrace{\frac{1}{2}f(0) + \sum_{j=1}^{n-1} f(j) + \frac{1}{2}f(n)}_{T(1)} + R_0.$$

Die Beziehung gilt also für $m = 0$. Das Restglied R_0 kann nun durch partielle Integration weiter umgeformt werden,

$$\begin{aligned}R_0 &= -\int_0^n \overline{B}_1(x) f'(x)\, dx = \left[-\frac{1}{2}\overline{B}_2(x) f'(x)\right]_0^n + \frac{1}{2}\int_0^n \overline{B}_2(x) f''(x)\, dx \\ &= -\frac{1}{2}B_2(0)\,[f'(n) - f'(0)] + \frac{1}{2}\int_0^n \overline{B}_2(x) f''(x)\, dx \\ &= -\frac{B_2}{2!}[f'(n) - f'(0)] + \underbrace{\left[\frac{1}{6}\overline{B}_3(x) f''(x)\right]_0^n}_{0} \underbrace{-\frac{1}{6}\int_0^n \overline{B}_3(x) f'''(x) dx}_{R_1} \\ &= -\frac{B_2}{2!}[f'(n) - f'(0)] + R_1.\end{aligned}$$

Durch $2m$-malige partielle Integration folgt dann schließlich

$$\int_0^n f(x)\, dx = T(1) + R_0 = T(1) - \sum_{j=1}^{m} \frac{B_{2j}}{(2j)!}[f^{(2j-1)}(n) - f^{(2j-1)}(0)] + R_m.$$

□

Damit erhalten wir nun

Satz 4.12. *Sei* $m \in \mathbb{N},\ f \in C^{2m+1}[A, B]$ *und*

$$T(h) := h\left(\frac{f(A)}{2} + \sum_{j=1}^{n-1} f(A+jh) + \frac{f(B)}{2}\right), \quad h := \frac{B-A}{n}$$

mit $n \in \mathbb{N}$ die zusammengesetzte Trapezregel. Dann gilt

$$\int_A^B f(x)\,dx = T(h) - \sum_{j=1}^{m-1} \frac{B_{2j}\cdot h^{2j}}{(2j)!}[f^{(2j-1)}(B) - f^{(2j-1)}(A)] - \\ - \frac{(B-A)\,B_{2m}\,h^{2m}}{(2m)!} f^{(2m)}(\xi)$$

mit einer von m und n abhängigen Zwischenstelle $\xi \in (a,b)$.

Beweis. Sei $g(x) := f(A+xh)$, dann ist

$$\int_0^n g(x)\,dx = \int_0^n f(\underbrace{A+xh}_{:=y})\,dx = \frac{1}{h}\int_A^B f(y)\,dy$$

und $g^{(j)}(x) = h^j f^{(j)}(A+xh),\ j \in \mathbb{N}_0$, sowie

$$\frac{1}{2}g(0) + \sum_{j=1}^{n-1} g(j) + \frac{1}{2}g(n) = \frac{f(A)}{2} + \sum_{j=1}^{n-1} f(A+jh) + \frac{1}{2}f(B) = \frac{1}{h}T(h).$$

Aus Satz 4.11 folgt daher mit der Substitution $y = A+xh$

$$\begin{aligned}
&\int_A^B f(x)\,dx = h\int_0^n g(x)\,dx \\
= \;& h\Big\{\underbrace{(\frac{1}{2}g(0) + \sum_{j=1}^{n-1} g(j) + \frac{1}{2}g(n))}_{\frac{1}{h}T(h)} - \underbrace{\sum_{j=1}^{m} \frac{B_{2j}}{(2j)!}[g^{(2j-1)}(n) - g^{(2j-1)}(0)]}_{\sum_{j=1}^{m}\frac{B_{2j}\,h^{2j-1}}{(2j)!}[f^{(2j-1)}(B)-f^{(2j-1)}(A)]} \\
&-\frac{1}{(2m+1)!}\int_0^n \overline{B}_{2m+1}(x)\ \underbrace{g^{(2m+1)}(x)}_{h^{2m+1}f^{(2m+1)}(Ax+h)}\ dx\Big\} \\
= \;& T(h) - \sum_{j=1}^{m-1}\frac{B_{2j}\,h^{2j}}{(2j)!}[f^{(2j-1)}(B) - f^{(2j-1)}(A)] \\
&-\frac{B_{2m}\,h^{2m}}{(2m)!}[f^{(2m-1)}(B) - f^{(2m-1)}(A)] \\
&-\frac{h^{2m+2}}{(2m+1)!}\,\frac{1}{h}\int_A^B \overline{B}_{2m+1}(\frac{y-A}{h})\,f^{(2m+1)}(y)\,dy.
\end{aligned}$$

Fasst man die letzten beiden Glieder zusammen, finden wir durch partielle Integration

$$\begin{aligned} I &:= \frac{-B_{2m}h^{2m}}{(2m)!}[f^{(2m-1)}(B) - f^{(2m-1)}(A)] \\ &\quad -\frac{h^{2m+1}}{(2m+1)!}\int_A^B \overline{B}_{2m+1}(\frac{y-A}{h}) f^{(2m+1)}(y)dy \\ &= \frac{-B_{2m}h^{2m}}{(2m)!}[f^{(2m-1)}(B) - f^{(2m-1)}(A)] + \frac{h^{2m}}{(2m)!}\int_A^B \overline{B}_{2m}(\frac{x-A}{h}) f^{(2m)}(x)\,dx, \end{aligned}$$

Da $\overline{B}_{2m}(\frac{x-A}{h}) - B_{2m}(0)$ nach Lemma 4.10 in $(0,1)$ keine Nullstellen hat, folgt nach dem verallgemeinerten Mittelwertsatz

$$\begin{aligned} I &= \frac{h^{2m}}{(2m)!}\int_A^B [\overline{B}_{2m}(\frac{x-A}{h}) - B_{2m}(0)] f^{(2m)}(x)\,dx \\ &= \frac{h^{2m}}{(2m)!} f^{(2m)}(\xi) \quad \underbrace{\int_A^B [\overline{B}_{2m}(\frac{x-A}{h}) - B_{2m}(0)]\,dx}_{-(B-A)\,B_{2m} + \underbrace{\frac{h}{2m+1}\overline{B}_{2m+1}(\frac{x-A}{h})}_{=0}\Big|_A^B} \\ &= -\frac{h^{2m}}{(2m!)} f^{(2m)}(\xi)\,(B-A)\,B_{2m}. \end{aligned}$$

wobei $\xi \in (A,B)$. □

Bei der Anwendung der zusammengesetzten Trapezregel ist der Fehler also besonders klein, wenn $f(A) = f(B)$ gilt. Insbesondere erhalten wir

Folgerung 4.13. *Es sei $f \in C^{2m+1}[A,B]$ periodisch mit der Periode $B-A$. Dann gilt*

$$\int_A^B f(x)\,dx = T(h) - \frac{(B-A)B_{2m}h^{2m}}{(2m)!} f^{(2m)}(\xi)$$

für ein $\xi \in (A,B)$.

Aus Satz 4.12 folgt, dass der Quadraturfehler der zusammengesetzten Trapezregel $T(h)$ zur Maschenweite $h = (B-A)/n$ eine Darstellung der Form

$$\int_A^B f(x)\,dx - T(h) = \alpha_2 h^2 + \alpha_4 h^4 + \ldots + \alpha_{2m-2}h^{2m-2} + \mathcal{O}(h^{2m})$$

hat, wobei $\alpha_{2j} := \frac{B_2 j}{(2j)!}\left[f^{(2j-1)}(B) - f^{(2j-1)}(A)\right]$. Folglich gilt mit $0 < q < 1$

$$\int_A^B f(x)\,dx - T(qh) = \alpha_2(qh)^2 + \alpha_4(qh)^4 + \ldots + \alpha_{2m-2}(qh)^{2m-2} + \mathcal{O}(h^{2m}).$$

Wähle nun die Linearkombination $T^{(1)}(h) := \frac{T(qh)-q^2T(h)}{1-q^2}$ aus $T(h)$ und $T(qh)$. Dann folgt

$$\begin{aligned}
&\int_A^B f(x)\,dx - T^{(1)}(h)\\
= \;& \frac{1}{1-q^2}\Big(\int_A^B f(x)\,dx - T(qh)\Big) - \frac{q^2}{1-q^2}\Big(\int_A^B f(x)\,dx - T(h)\Big)\\
= \;& \frac{1}{1-q^2}[\alpha_2(qh)^2 + \alpha_4(qh)^4 + \ldots + \alpha_{2m-2}(qh)^{2m-2} + \mathcal{O}(h^{2m})]\\
& -\frac{q^2}{1-q^2}[\alpha_2 h^2 + \alpha_4 h^4 + \ldots + \alpha_{2m-2}h^{2m-2} + \mathcal{O}(h^{2m})]\\
= \;& \frac{1}{1-q^2}[\underbrace{(\alpha_2(qh)^2 - q^2\alpha_2 h^2)}_{0} + (\alpha_4(qh)^4 - q^2\alpha_4 h^4) + \ldots\\
& + (\alpha_{2m-2}(qh)^{2m-2} - q^2\alpha_{2m-2}h^{2m-2}) + \mathcal{O}(h^{2m})]\\
= \;& \underbrace{\frac{\alpha_4(q^4-q^2)}{1-q^2}}_{:=\hat\alpha_4=-q^2\alpha_4} h^4 + \ldots + \underbrace{\alpha_{2m-2}\frac{(q^{2m-2}-q^2)}{1-q^2}}_{:=\hat\alpha_{2m-2}} h^{2m-2} + \mathcal{O}(h^{2m})
\end{aligned}$$

mit $\hat\alpha_{2j+2} = q^2\frac{(q^{2j}-1)}{1-q^2}\,\alpha_{2j+2}$ für $1 \le j \le m-2$.
Die neue Quadraturformel $T^{(1)}(h)$ verhält sich bzgl. der Ordnung des Quadraturfehlers in h wesentlich günstiger als $T(h)$. Speziell für $q = 1/2$ ergibt sich

$$\begin{aligned}
& T^1(h) = \frac{T(\frac{1}{2}h) - \frac{1}{4}T(h)}{\frac{3}{4}} = \frac{4T(\frac{1}{2}h) - T(h)}{3}\\
= \;& \frac{4h}{6}\Big(\frac{f(A)}{2} + \sum_{j=1}^{2n-1} f(A + j\frac{h}{2}) + \frac{f(B)}{2}\Big) - \frac{h}{3}\Big(\frac{f(A)}{2} + \sum_{j=1}^{n-1} f(A+jh) + \frac{f(B)}{2}\Big)\\
= \;& \frac{h}{2}\left[\frac{f(A)}{3} + \sum_{j=1}^{n-1}\Big[\frac{4}{3}f(A + \frac{(2j-1)h}{2}) + \frac{2}{3}f(A+jh)\Big] + \frac{4}{3}f(B - \frac{h}{2}) + \frac{f(B)}{3}\right]
\end{aligned}$$

Nach Berechnung von $T(\frac{h}{4})$ ergibt sich

$$T^{(1)}(\frac{h}{2}) = \frac{4\,T(\frac{h}{4}) - T(\frac{h}{2})}{3}.$$

Setze nun

$$T^{(2)}(h) := \frac{16\ T^{(1)}(\frac{h}{2}) - T^{(1)}(h)}{15}.$$

Dann folgt für den Quadraturfehler

$$\begin{aligned}
& \int_A^B f(x)dx - T^{(2)}(h) \\
= & \int_A^B f(x)dx - \frac{16}{15}\left(\frac{4}{3}T(\frac{h}{4}) - \frac{1}{3}T(\frac{h}{2})\right) + \frac{1}{15}\left(\frac{4}{3}T(\frac{h}{2}) - \frac{1}{3}T(h)\right) \\
= & \underbrace{\alpha_2\,(\frac{64}{45}\frac{h^2}{16} - \frac{20}{45}\frac{h^2}{4} + \frac{1}{45}h^2)}_{h^2(\frac{4}{45}-\frac{5}{45}+\frac{1}{45})=0} + \underbrace{\alpha_4\,(\frac{64}{45}\frac{h^4}{256} - \frac{20}{45}\frac{h^4}{16} + \frac{1}{45}h^4)}_{\frac{h^4}{45}(\frac{1}{4}-\frac{5}{4}+1)=0} + \mathcal{O}(h^6).
\end{aligned}$$

Diese Idee lässt sich fortsetzen. Mit $T_i^{(k)} := T^{(k)}(\frac{h}{2^i})$ ergibt sich die Bildungsvorschrift

$$T_i^{(k)} := \frac{4^k T_{i+1}^{(k-1)} - T_i^{(k-1)}}{4^k - 1}.$$

Dieses Halbierungsverfahren wurde 1955 von Romberg vorgeschlagen.

Algorithmus:

Gegeben: $f(\frac{B-A}{2^m}\,j) \quad j = 0, \ldots, 2^m$.

1. Für $i = 0, \ldots, m$ setze $h_i := \frac{B-A}{2^i}$ mit $n_i := 2^i$.

2. Berechne

$$T_i^{(0)} := T(h_i) = h_i\,\{\frac{f(A)}{2} + \sum_{j=1}^{n_i-1} f(A + jh_i) + \frac{f(B)}{2}\}.$$

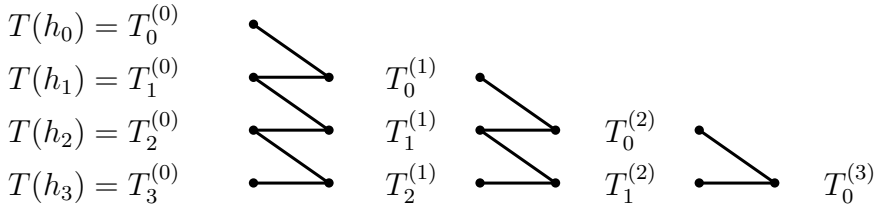
3. Für $k = 1, \ldots m$ setze

$$T_i^{(k)} := T_{i+1}^{(k-1)} + \frac{T_{i+1}^{(k-1)} - T_i^{(k-1)}}{4^k - 1} \qquad i = 0, \ldots, m-k.$$

Ausgabe: $T_0^{(m)} \approx \int_A^B f(x)dx$.

Das Romberg-Verfahren kann mit Hilfe eines Dreieckschemas berechnet werden.

Beispiel: $m = 3$.



Bemerkung. Die Berechnung der Startwerte lässt sich noch vereinfachen (siehe Schritt 2 des Algorithmus)

$$\begin{aligned} T_i^{(0)} &= \frac{h_i}{2}[f(A) + 2f(A+h_i) + \ldots + 2f(B-h_i) + f(B)] \\ &= \frac{h_{i-1}}{4}[f(A) + 2f(A+\underbrace{2h_i}_{h_{i-1}}) + \ldots + 2f(B-\underbrace{2h_i}_{h_{i-1}}) + f(B)] \\ &+ h_i[f(A+h_i) + f(A+3h_i) + \ldots + f(B-h_i)] \\ &= \frac{1}{2}T_{i-1}^{(0)} + h_i \sum_{j=1}^{2^{i-1}} f(A+(2j-1)h_i). \end{aligned}$$

Dadurch werden Funktionswerte nicht doppelt berechnet.

Satz 4.14. *Bei gegebener Funktion $f \in C^{2k+2}[A,B]$ sei $T_i^{(k)}$ eine durch das Romberg-Verfahren berechnete Näherung für $\int_A^B f(x)dx$. Dann gilt*

$$\int_A^B f(x)\,dx - T_i^{(k)} = (B-A)^{2k+3}\,\frac{(-1)^{k+1}}{2^{(k+1)(k+2i)}}\,\frac{B_{2k+2}}{(2k+2)!}f^{(2k+2)}(\xi)$$

mit einer von k und i abhängigen Zwischenstelle $\xi \in (A,B)$.

Beweis. Jede Spalte des Romberg-Schemas entsteht durch Linearkombination der Werte der vorangehenden Spalte. Damit sind die Werte $T_i^{(k)}$ schließlich Linearkombinationen der ersten Spalte,

$$T_i^{(k)} = \sum_{j=i}^{i+k} c_{ij}^{(k)} T(h_j) \text{ mit } c_{ij}^{(k)} = \prod_{\substack{l=i \\ l\neq j}}^{i+k} \frac{h_l^2}{h_l^2 - h_j^2}.$$

Dies folgt durch vollständige Induktion aus

$$T_i^{(k)} = T_{i+1}^{(k-1)} + \frac{T_{i+1}^{(k-1)} - T_i^{(k-1)}}{4^k - 1}.$$

Wir betrachten das Polynom $f(x) = x^r$ in Lagrange-Darstellung mit den Stützstellen h_j^2, $j = i, \ldots, i+k$. Dann gilt nach Satz 1.6 für beliebige $x \in \mathbb{R}$

$$\sum_{j=i}^{i+k} (h_j^2)^r \prod_{\substack{l=i \\ l\neq j}}^{i+k} \frac{x - h_l^2}{h_j^2 - h_l^2} = x^r \qquad (r = 0, 1, \ldots, k).$$

Setzen wir $x = 0$, so folgt

$$\sum_{j=i}^{i+k} c_{ij}^{(k)}\, h_j^{2r} = \begin{cases} 1 & \text{für} \quad r = 0 \\ 0 & \text{für} \quad r = 1, \ldots, k. \end{cases} \tag{4.2}$$

Im Beweis von Satz 4.13 war

$$\begin{aligned} & \int_A^B f(x)dx \\ = \;& T(h_j) + \sum_{l=1}^{k} \alpha_{2l} h_j^{2l} + \frac{h_j^{2k+2}}{(2k+2)!} \int_B^A [\overline{B}_{2k+2}(\frac{x-A}{h_j}) - B_{2k+2}(0)] f^{(2k+2)}(x)\, dx \end{aligned}$$

mit gewissen Konstanten $\alpha_2, \alpha_4 \ldots \alpha_{2k}$. Damit folgt

$$\begin{aligned} & \underbrace{\sum_{j=i}^{i+k} c_{ij}^{(k)}}_{\overset{(4.2)}{=} 1} \int_A^B f(x)dx = \underbrace{\sum_{j=i}^{i+k} c_{ij}^{(k)} T(h_j)}_{T_i^{(k)}} + \sum_{l=1}^{k} \alpha_{2l} \underbrace{\sum_{j=i}^{i+k} c_{ij}^{(k)} h_j^{2l}}_{\overset{(4.2)}{=} 0} \\ + \;& \frac{1}{(2k+2)!} \int_A^B \Big(\sum_{j=i}^{i+k} c_{ij}^{(k)} h_j^{2k+2} [\overline{B}_{2k+2}(\frac{x-A}{h_j}) - B_{2k+2}(0)] \Big) f^{(2k+2)}(x)dx \\ = \;& T_i^{(k)} + \frac{1}{(2k+2)!} \int_A^B K(x)\, f^{(2k+2)}(x)dx \end{aligned}$$

mit

$$K(x) := \sum_{j=i}^{i+k} c_{ij}^{(k)} h_j^{2k+2} \Big[\overline{B}_{2k+2}(\frac{x-A}{h_j}) - B_{2k+2}(0) \Big].$$

Man kann nun zeigen, dass $K(x)$ in $[A, B]$ sein Vorzeichen nicht ändert (siehe z.B. Burlisch: Numerische Mathematik **6** (1964), S. 6 - 16.) Dann ergibt sich aus dem Mittelwertsatz der Integralrechnung

$$\begin{aligned} \int_A^B f(x)dx - T_i^{(k)} &= \frac{1}{(2k+2)!} f^{(2k+2)}(\xi) \int_A^B K(x)dx \\ &= \frac{1}{(2k+2)!} f^{(2k+2)}(\xi)\, [-(B-A)\, B_{2k+2} \sum_{j=i}^{i+k} c_{ij}^{(k)} h_j^{2k+2}] \end{aligned}$$

da

$$\int_A^B \overline{B}_{2k+2}(\frac{x-A}{h_j}) - B_{2k+2}(0)dx = -(B-A)B_{2k+2}(0).$$

Schließlich gilt mit $f(x) = x^{k+1}$

$$x^{k+1} = \sum_{j=i}^{i+k} (h_j^2)^{k+1} \prod_{\substack{l=i \\ l\neq j}}^{i+k} \frac{x - h_l^2}{h_j^2 - h_l^2} + \prod_{j=i}^{i+k} (x - h_j^2),$$

denn auf beiden Seiten stehen Polynome von Grad $k+1$, die in den $k+1$ Stützstellen $h_i^2, \ldots, h_{i+k}^2$ übereinstimmen und den Höchstkoeffizienten 1 haben. Für $x = 0$ folgt

$$0 = \sum_{j=i}^{i+k} h_j^{2k+2} \underbrace{\prod_{\substack{l=i \\ l\neq j}}^{i+k} \frac{h_l^2}{h_l^2 - h_j^2}}_{c_{ij}^{(k)}} + \underbrace{\prod_{j=i}^{i+k} (-h_j^2)}_{\frac{(-1)^{k+1}(B-A)^{2k+2}}{2^{2i}, 2^{2(i+1)} \ldots 2^{2(i+k)}}},$$

und damit

$$\int_A^B f(x)dx - T_i^{(k)} = \frac{1}{(2k+2)!} f^{(2k+2)}(\xi)\, (-1)^{k+1} \frac{(B-A)^{(2k+3)}}{2^{(k+1)(k+2i)}} B_{2k+2}.$$

□

Folgerung 4.15. *Bei gegebenem $f \in C[A, B]$ sei $T_i^{(k)}$ eine durch das Romberg-Verfahren berechnete Näherung von $\int_A^B f(x)dx$. Dann gilt:*

(i) Ist $f \in \Pi_{2k+1}$, so ist $T_i^{(k)} = \int_A^B f(x)dx$.

(ii) Ist $f \in C^{2k+2}[A, B]$, so ist $\lim_{i\to\infty} T_i^{(k)} = \int_A^B f(x)dx$.

Beweis. Die Behauptung (i) folgt aus Satz 4.14 da $f^{(2k+2)}(\xi) = 0$. Aussage (ii) folgt ebenfalls aus Satz 4.14, da

$$\lim_{i\to\infty} (B-A)^{2k+3} \frac{(-1)^{k+1}}{2^{(k+1)(k+2i)}} \frac{B_{2k+2}}{(2k+2)!} f^{(2k+2)}(\xi) = 0.$$

□

Beispiel: Wir wollen das Integral $\int_0^1 t^5 dt = \frac{1}{6} = 0.16666..$ mit Hilfe des Romberg-Verfahrens berechnen. Mit $h_0 = 1, h_1 = \frac{1}{2}, h_2 = \frac{1}{4}$ und den Funktionswerten $f(0) = 0$, $f(1) = 1$, $f(\frac{1}{2}) = \frac{1}{32}$, $f(\frac{1}{4}) = \frac{1}{1024}$, $f(\frac{3}{4}) = \frac{243}{1024}$, erhalten wir das Schema

$$\begin{array}{l} T_0^{(0)} = T(1) = 0.500000 \\ T_1^{(0)} = T(1/2) = 0.265625 \quad \vdots, \quad T_0^{(1)} = 0.187500 \\ T_2^{(0)} = T(1/4) = 0.192383 \quad \vdots, \quad T_0^{(1)} = 0.167969 \quad \vdots, \quad T_0^{(2)} = 0.166667 \end{array}$$

mit

$$
\begin{aligned}
T_0^{(0)} &= T(1) = \frac{f(0)+f(1)}{2} = \frac{1}{2} = 0.5, \\
T_1^{(0)} &= T(\frac{1}{2}) = \frac{T(1)}{2} + \frac{f(\frac{1}{2})}{2} = \frac{1}{4} + \frac{1}{64} = \frac{17}{64} = 0.265625, \\
T_2^{(0)} &= T(\frac{1}{4}) = \frac{T(\frac{1}{2})}{2} + \frac{f(\frac{1}{4}) + f(\frac{3}{4})}{4} = \frac{17}{128} + \frac{244}{4096} = \frac{197}{1024} = 0.192383, \\
T_0^{(1)} &= T_1^{(0)} + \frac{T_1^{(0)} - T_0^{(0)}}{3} = \frac{17}{64} - \frac{5}{64} = \frac{3}{16} = 0.1875, \\
T_1^{(1)} &= T_2^{(0)} + \frac{T_2^{(0)} - T_1^{(0)}}{3} = \frac{197}{1024} - \frac{25}{1024} = \frac{43}{256} = 0.167969, \\
T_0^{(2)} &= T_1^{(1)} + \frac{T_1^{(1)} - T_0^{(1)}}{15} = \frac{1}{6} = 0.166667.
\end{aligned}
$$

Literatur

1. C. De Boor, A practical guide to splines, Springer, 2001.

2. G. Hämmerlin, K.-H. Hoffmann, Numerische Mathematik, Springer, 1989.

3. K. Jetter, Numerische Mathematik III, Vorlesung gehalten in Duisburg-Essen, Sommersemester 1995, ausgearbeitet von Andreas Hochhaus.

4. R.Q. Jia, Subdivision schemes in L_p spaces, Adv. Comput. Math. 3, 309-341 (1995).

5. W. Schaback, H. Wendland, Numerische Mathematik, Springer, 2005.

6. R. Schwarz, Numerische Mathematik, Teubner, Stuttgart, 1986.

7. J. Stoer, Numerische Mathematik 1, Springer, 1989.