

2. INTERPOLATION UND APPROXIMATION

Das Interpolationsproblem besteht darin, zu gegebenen Punkten

$$(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$$

ein Polynom vom Grad n mit $p(x_i) = y_i$, $i = 0, \dots, n$ zu bestimmen. Sind die Werte y_i die Funktionswerte einer Funktion f an den Knoten x_i , also $f(x_i) = y_i$, dann stellt sich die Frage, wie groß der Interpolationsfehler, d. h.

$$f(x) - p(x) = ?$$

ist.

In einem weiteren Approximationsproblem, welches wir untersuchen werden, lautet die Aufgabe, zu einer gegebenen Funktion $f : [a, b] \rightarrow \mathbb{R}$ eine einfach auszuwertende Funktion (zum Beispiel ein Polynom $p : [a, b] \rightarrow \mathbb{R}$) zu finden, so dass der Fehler $(f - p)$ "klein" ist, etwa in folgenden Sinne:

$$\begin{aligned} f(x) &= p(x) && \text{für endlich viele Punkte} \\ \int_a^b (f(x) - p(x))^2 dx &= \min && L^2\text{-Norm} \\ \max_{x \in [a, b]} |f(x) - p(x)| &= \min && L^\infty\text{-Norm} \end{aligned}$$

2.1 Newton'sche Interpolationsformel

Wir betrachten zunächst einfache Beispiele:

$n = 1$: Zu (x_0, y_0) und (x_1, y_1) suche ein Polynom vom Grad 1, welches beide Punkte verbindet:

$$p(x) = y_0 + (x - x_0) \frac{y_1 - y_0}{x_1 - x_0}.$$

$n = 2$: Zu (x_0, y_0) , (x_1, y_1) und (x_2, y_2) bestimme ein Polynom vom Grad 2 durch diese Punkte. Wir verwenden die obige Gerade und ergänzen einen weiteren Summanden, der an den bisherigen Knoten x_0 und x_1 verschwindet und daher die Interpolationseigenschaft an diesen Knoten nicht stört:

$$p(x) = y_0 + (x - x_0) \frac{y_1 - y_0}{x_1 - x_0} + a(x - x_0)(x - x_1).$$

Jetzt muss man nur noch a so bestimmen, dass $p(x_2) = y_2$ gilt:

$$\begin{aligned} p(x_1) &= y_1 = y_0 + (x_1 - x_0) \frac{y_1 - y_0}{x_1 - x_0} \\ p(x_2) &= y_2 = y_0 + (x_2 - x_0) \frac{y_1 - y_0}{x_1 - x_0} + a(x_2 - x_0)(x_2 - x_1) \end{aligned}$$

Subtraktion dieser beiden Gleichungen ergibt

$$y_2 - y_1 = (x_2 - x_1) \frac{y_1 - y_0}{x_1 - x_0} + a(x_2 - x_0)(x_2 - x_1)$$

und darauf folgt mit Division durch $(x_2 - x_0)(x_2 - x_1)$

$$a = \frac{1}{x_2 - x_0} \left[\frac{y_2 - y_1}{x_2 - x_1} - \frac{y_1 - y_0}{x_1 - x_0} \right].$$

Bevor wir diese Konstruktion rekursiv anwenden, führen wir eine vereinfachende Schreibweise ein:

Definition 2.1. (*Dividierte Differenzen*)

Für gegebene Punkte (x_i, y_i) , $i = 0, \dots, n$ mit paarweise verschiedenen x_i sei

$$\begin{aligned} y[x_i] &= y_i \\ \delta y[x_i, x_j] &= \frac{y[x_j] - y[x_i]}{x_j - x_i} \\ \delta^2 y[x_i, x_j, x_k] &= \frac{\delta y[x_j, x_k] - \delta y[x_i, x_j]}{x_k - x_i} \\ \delta^n y[x_{i_0}, x_{i_1}, \dots, x_{i_n}] &= \frac{\delta^{n-1} y[x_{i_1}, \dots, x_{i_n}] - \delta^{n-1} y[x_{i_0}, \dots, x_{i_{n-1}}]}{x_{i_n} - x_{i_0}}, \quad n \geq 2. \end{aligned}$$

Zur rekursiven Berechnung der dividierten Differenzen ordnet man diese am besten in einem Schema an:

$$\begin{array}{c|cccc} x_0 & y_0 = y[x_0] & & & \\ & & \delta y[x_0, x_1] & & \\ x_1 & y_1 = y[x_1] & & \delta^2 y[x_0, x_1, x_2] & \\ & & \delta y[x_1, x_2] & & \delta^3 y[x_0, x_1, x_2, x_3] \\ x_2 & y_2 = y[x_2] & & \delta^2 y[x_1, x_2, x_3] & \\ & & \delta y[x_2, x_3] & & \\ x_3 & y_3 = y[x_3] & & & \end{array}$$

Bemerkung. Falls die Punkte x_i äquidistant sind, also $x_i = x_0 + ih$, $i = 0, \dots, n$ gilt, so ist mit der Notation aus Abschnitt 1.10

$$\begin{aligned} \delta y[x_0, x_1] &= \frac{y_1 - y_0}{h} = \frac{1}{h} \Delta y_0 \\ \delta^2 y[x_0, x_1, x_2] &= \frac{\frac{1}{h} \Delta y_1 - \frac{1}{h} \Delta y_0}{2h} = \frac{1}{2!h^2} \Delta^2 y_0 \\ &\vdots \\ \delta^n y[x_0, x_1, \dots, x_n] &= \frac{1}{n!h^n} \Delta^n y_0 \end{aligned}$$

(Übung).

Satz 2.2. (*Newton'sche Interpolationsformel, 1669*)

Es existiert ein eindeutig bestimmtes Polynom vom Grad n , das durch die Punkte (x_i, y_i) , $i = 0, \dots, n$ mit paarweise verschiedenen x_i geht. Es lässt sich berechnen durch

$$\begin{aligned} p(x) &= y[x_0] + (x - x_0)\delta y[x_0, x_1] + (x - x_0)(x - x_1)\delta^2 y[x_0, x_1, x_2] \\ &\quad + \dots + (x - x_0)(x - x_1) \cdots (x - x_{n-1})\delta^n y[x_0, x_1, \dots, x_n]. \end{aligned}$$

Bemerkung. (Geschichte) Die Interpolation wurde bereits im 17. Jahrhundert wichtig, als man Logarithmentafeln für die Seefahrt benötigte. Harriot 1611 und Briggs 1617 verwendeten bereits obige Formel für äquidistante Stützstellen.

Beweis. Der Beweis erfolgt mit Induktion. Für $n = 1, 2$ haben wir die Behauptung bereits gezeigt. Nehmen wir an, sie ist für $n - 1$ richtig. Dann ist

$$p_1(x) = y[x_0] + (x - x_0)\delta y[x_0, x_1] + \dots + (x - x_0) \cdots (x - x_{n-2})\delta^{n-1}y[x_0, \dots, x_{n-1}]$$

das eindeutig bestimmte Polynom vom Grad $n - 1$ durch die Punkte (x_i, y_i) , $i = 0, \dots, n - 1$. Das gesuchte Polynom p muss dann die Form

$$p(x) = p_1(x) + a(x - x_0) \cdots (x - x_{n-1})$$

haben, denn es muss $p(x_i) = p_1(x_i) = y_i$ für $i = 0, \dots, n - 1$ gelten. Der Koeffizient a wird so bestimmt, dass $p(x_n) = y_n$ gilt. Damit ist die Eindeutigkeit von p gezeigt. Wir zeigen jetzt noch, dass $a = \delta^n y[x_0, x_1, \dots, x_n]$. Dazu sei p_2 das eindeutig bestimmte Polynom vom Grad $n - 1$ durch die Punkte (x_i, y_i) , $i = 1, \dots, n$, also

$$\begin{aligned} p_2(x) &= y[x_1] + (x - x_1)\delta y[x_1, x_2] + \dots + (x - x_1) \cdots (x - x_{n-1})\delta^{n-1}y[x_1, \dots, x_n] \\ &= x^{n-1}\delta^{n-1}y[x_1, x_2, \dots, x_n] + (\text{Polynom vom Grad } \leq n - 2). \end{aligned}$$

Dann ist

$$\tilde{p}(x) = \frac{1}{x_n - x_0} [(x_n - x)p_1(x) + (x - x_0)p_2(x)]$$

ein Polynom vom Grad n , das durch (x_i, y_i) , $i = 0, \dots, n$ geht, wegen der Eindeutigkeit muss es mit p übereinstimmen. Ein Vergleich der Koeffizienten von x^n ergibt

$$a = \frac{1}{x_n - x_0} (\delta^{n-1}y[x_1, \dots, x_n] - \delta^{n-1}y[x_0, x_1, \dots, x_{n-1}]) = \delta^n y[x_0, x_1, \dots, x_n]. \quad \square$$

Beispiel. Berechnung eines Interpolationspolynoms mit dem Newton-Schema.

x_i	y_i	δy	$\delta^2 y$	$\delta^3 y$	$\delta^4 y$	$\delta^5 y$
0	-1					
		1				
2	1		3/8			
		5/2		-77/120		
4	6		-17/6		167/960	
		-6		3/4		-287/9600
5	0		5/3		-1/8	
		2/3		-1/4		
8	2		1/6			
		3/2				
10	5					

Das Interpolationspolynom ist durch

$$\begin{aligned} p(x) &= -1 + x + x(x - 2)\frac{3}{8} - x(x - 2)(x - 4)\frac{77}{120} + x(x - 2)(x - 4)(x - 5)\frac{167}{960} \\ &\quad - x(x - 2)(x - 4)(x - 5)(x - 8)\frac{287}{9600} \end{aligned}$$

gegeben. \diamond

Die Auswertung des Newton'schen Interpolationspolynoms sollte mit dem **Hornerschema** erfolgen. Für Polynome der Form

$$\begin{aligned} p(x) &= a_0 + a_1x + \dots + a_{n-1}x^{n-1} + a_nx^n \\ &= a_0 + x(a_1 + x(a_2 + \dots x(a_{n-1} + xa_n) \dots)) \end{aligned} \quad (2.1)$$

erfolgt die Implementierung mit Algorithmus 2.1, für Polynome in der Newton-Form

$$\begin{aligned} p(x) &= y[x_0] + (x - x_0) \left(\delta y[x_0, x_1] + (x - x_1) \left(\delta^2 y[x_0, x_1, x_2] + \dots \right. \right. \\ &\quad \left. \left. + (x - x_{n-2}) (\delta^{n-1} y[x_0, x_1, \dots, x_{n-1}] + (x - x_{n-1}) \delta^n y[x_0, x_1, \dots, x_n]) \dots \right) \right) \end{aligned} \quad (2.2)$$

mit Algorithmus 2.2.

Algorithmus 2.1 Hornerschema

```
{Berechnet  $p(x)$  in der Form (2.1) bei gegebenen Koeffizienten  $a_i$ }
 $p := a_n$ 
for  $i = 1, \dots, n$  do
   $p := a_{n-i} + xp$ 
end for
```

Algorithmus 2.2 Hornerschema für Newton'sche Interpolationspolynome

```
{Berechnet  $p(x)$  in der Form (2.2) bei gegebenen dividierten Differenzen}
 $p := \delta^n y[x_0, x_1, \dots, x_n]$ 
for  $i = 1, \dots, n$  do
   $p := \delta^{n-i} y[x_0, \dots, x_{n-i}] + (x - x_{n-i})p$ 
end for
```

Benötigt man den Wert des Interpolationspolynoms nur an einer einzigen Stelle, so kann man diesen mit Hilfe des folgenden Satzes effizient berechnen.

Satz 2.3. (*Aitken–Neville–Interpolation*)

Zu gegebenen Interpolationsdaten (x_i, y_i) , $i = 0, 1, \dots$ bezeichne p_{i_0, i_1, \dots, i_k} das Polynom, das $p_{i_0, i_1, \dots, i_k}(x_{i_j}) = y_{i_j}$ für $j = 0, 1, \dots, k$ erfüllt. Die Polynome können wie folgt rekursiv berechnet werden:

$$\begin{aligned} p_i(x) &\equiv y_i, \\ p_{i_0, i_1, \dots, i_k}(x) &= \frac{1}{x_{i_k} - x_{i_0}} \left((x - x_{i_0}) p_{i_1, \dots, i_k}(x) - (x - x_{i_k}) p_{i_0, i_1, \dots, i_{k-1}}(x) \right). \end{aligned}$$

Beweis. Vollständige Induktion über k (Übung). □

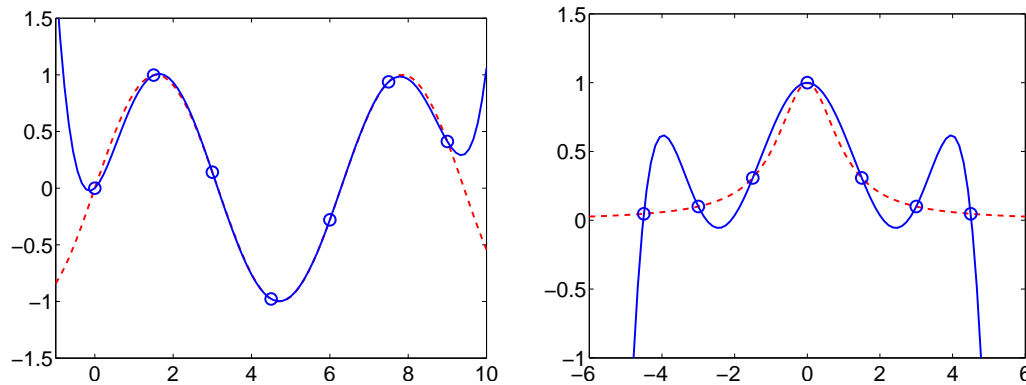


Abb. 2.1: Interpolation der Funktionen $f(x) = \sin x$ (links) und $f(x) = 1/(1+x^2)$ (rechts).

2.2 Fehler bei der Polynominterpolation

Wir nehmen jetzt an, dass die Punkte (x_i, y_i) aus dem Graphen einer Funktion $f : [a, b] \rightarrow \mathbb{R}$ stammen, also

$$y_i = f(x_i), \quad i = 0, \dots, n$$

gilt. Das Ziel dieses Abschnitts ist es, den Fehler $f(x) - p(x)$ des Interpolationspolynoms p zu untersuchen und abzuschätzen. Betrachten wir auch hier zunächst als Beispiel die Interpolation der Funktionen $f(x) = \sin x$ und $f(x) = 1/(1+x^2)$ in Abbildung 2.1. Offensichtlich wird die erste Funktion sehr gut durch das Interpolationspolynom approximiert, die zweite jedoch schlecht. Woran liegt das?

Bevor wir die Fehlerabschätzung zeigen, beweisen wir einen Zusammenhang zwischen den dividierten Differenzen und den Ableitungen der Funktion f .

Lemma 2.4. *Sei f n -mal stetig differenzierbar und $y_i = f(x_i)$ für $i = 0, \dots, n$ mit paarweise verschiedenen x_i . Dann gibt es ein $\xi \in (\min_i x_i, \max_i x_i)$ mit*

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

Beweis. Es sei p das Interpolationspolynom vom Grad n durch die Punkte (x_i, y_i) und $d(x) := f(x) - p(x)$. Aus der Interpolationseigenschaft folgt

$$d(x_i) = 0, \quad i = 0, \dots, n.$$

d ist nach Voraussetzung n -mal stetig differenzierbar. Daher können wir den Satz von Rolle anwenden und erhalten

$$d'(x) = 0 \quad \text{an } n \text{ Nullstellen in } (\min_i x_i, \max_i x_i).$$

Dasselbe Argument angewendet auf d' liefert

$$d''(x) = 0 \quad \text{an } n-1 \text{ Nullstellen in } (\min_i x_i, \max_i x_i)$$

und nach Induktion

$$d^{(n)}(x) = 0 \quad \text{an einer Nullstelle in } (\min_i x_i, \max_i x_i).$$

Es sei ξ diese Nullstelle, $d^{(n)}(\xi) = 0$. Dann gilt

$$f^{(n)}(\xi) = p^{(n)}(\xi) = n! \delta^n y[x_0, x_1, \dots, x_n],$$

da $\delta^n y[x_0, x_1, \dots, x_n]$ der Höchstkoeffizient von $p(x)$ ist ($p(x) = x^n \delta^n y[x_0, x_1, \dots, x_n] + \text{Polynom vom Grad } \leq n-1$). \square

Satz 2.5. Sei $f : [a, b] \rightarrow \mathbb{R}$ $(n+1)$ -mal differenzierbar und sei p das Interpolationspolynom vom Grad n durch $(x_i, f(x_i))$, $i = 0, \dots, n$. Dann gibt es zu jedem $x \in [a, b]$ ein $\xi = \xi(x) \in (\min\{x_i, x\}, \max\{x_i, x\})$ mit

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

Beweis. Es sei $\bar{x} \in [a, b]$ beliebig vorgegeben. Falls $\bar{x} = x_i$ ist nichts zu zeigen. Wir nehmen daher $\bar{x} \neq x_i$ für alle i an. \bar{p} sei das eindeutig bestimmte Polynom vom Grad $n+1$, das durch die Punkte

$$(x_i, f(x_i)), \quad i = 0, 1, \dots, n \quad \text{und} \quad (\bar{x}, f(\bar{x}))$$

geht. Nach der Newton'schen Interpolationsformel ist

$$\bar{p}(x) = p(x) + (x - x_0)(x - x_1) \cdots (x - x_n) \delta^{n+1} y[x_0, x_1, \dots, x_n, \bar{x}].$$

Wegen $f(\bar{x}) = \bar{p}(\bar{x})$ folgt die Behauptung für $x = \bar{x}$ aus Lemma 2.4. Da \bar{x} beliebig war, gilt sie für alle x . \square

Beispiel. Im Beispiel aus Abbildung 2.1 ist $n+1 = 7$. Für die Sinusfunktion ist die 7-te Ableitung durch 1 beschränkt, für die Funktion $f(x) = 1/(1+x^2)$ jedoch durch 4392. (Übung) \diamond

Satz 2.5 zeigt, dass der Interpolationsfehler das Produkt der $(n+1)$ -ten Ableitung von f ausgewertet an einer unbekannten Zwischenstelle ξ und dem Wert des sogenannten Knotenpolynoms

$$\omega_{n+1}(x) = (x - x_0)(x - x_1) \cdots (x - x_n)$$

ist. Da das Knotenpolynom nur von der Wahl der Knoten x_i abhängt, stellt sich die Frage, ob man die x_i so wählen kann, dass

$$\max_{x \in [a, b]} |\omega_{n+1}(x)| = \min? \quad (2.3)$$

Dies ist tatsächlich möglich, wenn man Tschebyscheff-Polynome zu Hilfe nimmt, wie wir später in Abschnitt 2.3 sehen werden. Zunächst diskutieren wir jedoch, wie sich Fehler in den Funktionswerten (Rundfehler oder Messfehler) auf die Interpolation auswirken. Dazu ist die Lagrange-Darstellung des Interpolationspolynoms besser geeignet als die Newton'sche.

Satz 2.6. Das Interpolationspolynom p durch (x_i, y_i) , $i = 0, \dots, n$ ist gegeben durch

$$p(x) = \sum_{i=0}^n y_i \ell_i(x), \quad \ell_j(x) = \frac{\prod_{k \neq j} (x - x_k)}{\prod_{k \neq j} (x_j - x_k)}.$$

Beweis. Die Lagrange-Polynome ℓ_j haben Grad n und erfüllen

$$\ell_j(x_m) = \begin{cases} 1, & j = m, \\ 0, & j \neq m. \end{cases}$$

Daraus folgt sofort die Interpolationseigenschaft $p(x_j) = y_j$. Wegen der Eindeutigkeit ist p das Interpolationspolynom. \square

Für den Fehler des Interpolationspolynoms zu gestörten Daten \hat{y}_i gilt

Satz 2.7. *Ist p das Interpolationspolynom durch die Punkte (x_i, y_i) , $i = 0, \dots, n$ und \hat{p} das durch (x_i, \hat{y}_i) , $i = 0, \dots, n$, dann gilt*

$$\max_{x \in [a, b]} |p(x) - \hat{p}(x)| \leq \Lambda_n \max_{i=0, \dots, n} |y_i - \hat{y}_i|.$$

mit der **Lebesgue-Konstante**

$$\Lambda_n := \max_{x \in [a, b]} \sum_{i=0}^n |\ell_i(x)|$$

Beweis. Die Interpolationspolynome sind nach Satz 2.6 von der Form

$$p(x) = \sum_{i=0}^n y_i \ell_i(x), \quad \hat{p}(x) = \sum_{i=0}^n \hat{y}_i \ell_i(x).$$

Für den Fehler gilt also

$$|p(x) - \hat{p}(x)| \leq \max_{i=0, \dots, n} |y_i - \hat{y}_i| \sum_{i=0}^n |\ell_i(x)|. \quad \square$$

Beispiel. Für äquidistante Stützstellen $x_i = a + \frac{i}{n}(b-a)$ im Intervall $[a, b] = [-1, 1]$ gilt

$$\Lambda_{10} \approx 40, \quad \Lambda_{20} \approx 3 \cdot 10^4, \quad \Lambda_{40} \approx 10^{10}, \quad \Lambda_n \approx \frac{2^{n+1}}{en \log n}.$$

Die Lebesgue-Konstante wächst also exponentiell mit der Anzahl der Knoten. Daher sind Interpolationspolynome hohen Grades bei äquidistanten Stützstellen unbrauchbar. In Abschnitt 2.3 werden wir eine Stützstellenverteilung mit $\Lambda_n \leq 4$ für $n \leq 100$ kennenlernen. \diamond

Ist die Interpolation ein geeignetes Mittel, um eine Funktion durch Polynome zu approximieren?

Satz 2.8. *Sei $f : [a, b] \rightarrow \mathbb{R}$ eine beliebige Funktion. p sei das Interpolationspolynom zu f in den Stützstellen $x_0, \dots, x_n \in [a, b]$. Dann gilt*

$$\max_{x \in [a, b]} |f(x) - p(x)| \leq (1 + \Lambda_n) \max_{x \in [a, b]} |f(x) - q(x)|$$

für alle Polynome $q \in \mathcal{P}_n$.

Beweis. Sei q ein beliebiges Polynom vom Grad $\leq n$. Wir schreiben

$$f - p = (f - q) + (q - p).$$

Nach Satz 2.7 mit $y_i = f(x_i)$ und $\hat{y}_i = q(x_i)$ (q ist das Interpolationspolynom zu sich selbst) gilt

$$\begin{aligned} \max_{x \in [a,b]} |p(x) - q(x)| &\leq \Lambda_n \max_{i=0,\dots,n} |f(x_i) - q(x_i)| \\ &\leq \Lambda_n \max_{x \in [a,b]} |f(x) - q(x)|. \end{aligned}$$

Die Behauptung folgt somit aus der Dreiecksungleichung. \square

2.3 Tschebyscheff-Interpolation

Im letzten Abschnitt haben wir gesehen, dass äquidistante Stützstellen sehr ungünstig sein können, insbesondere für Interpolation mit vielen Knoten. Wir möchten jetzt eine günstigere Wahl vorstellen, die auf Tschebyscheff-Polynomen basiert.

Definition 2.9. (*Tschebyscheff-Polynome*)

Für $n = 0, 1, \dots$ und $x \in [-1, 1]$ heißt

$$T_n(x) = \cos(n \arccos x)$$

n -tes **Tschebyscheff-Polynom** (1. Art).

Lemma 2.10. (*Eigenschaften von Tschebyscheff-Polynomen*)

(a) Es gilt die Rekursion

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad n \geq 1.$$

Für $n \geq 1$ ist T_n also ein Polynom vom Grad n mit Höchstkoeffizient 2^{n-1} .

(b) $|T_n(x)| \leq 1$ für $x \in [-1, 1]$.

(c) $T_n\left(\cos \frac{k\pi}{n}\right) = (-1)^k$, $k = 0, 1, \dots, n$.

(d) $T_n\left(\cos \frac{(2k+1)\pi}{2n}\right) = 0$, $k = 0, 1, \dots, n-1$.

Beweis. (a) $T_0(x) = \cos(0) = 1$ und $T_1(x) = x$ folgen direkt. Die Rekursionsformel folgt mit $x = \cos \varphi$ aus dem Additionstheorem

$$\cos((n+1)\varphi) + \cos((n-1)\varphi) = 2\cos(\varphi)\cos(n\varphi).$$

(b),(c) und (d) folgen unmittelbar aus den Eigenschaften von $\cos x$. \square

Lemma 2.11. Es sei $q \neq T_n$ ein Polynom vom Grad n mit Höchstkoeffizient 2^{n-1} , $q(x) = 2^{n-1}x^n + \dots$. Dann gilt

$$\max_{x \in [-1,1]} |q(x)| > \max_{x \in [-1,1]} |T_n(x)| = 1.$$

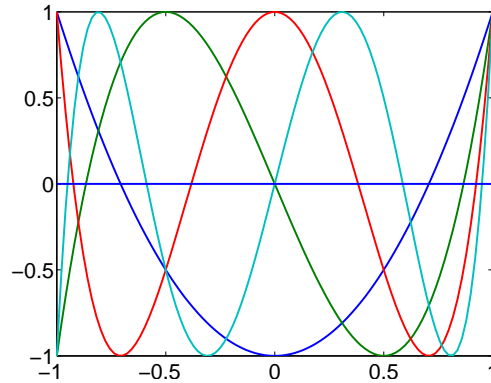


Abb. 2.2: Tschebyscheff-Polynome vom Grad 2 bis 5

Beweis. Der Beweis erfolgt durch Widerspruch. Angenommen, es wäre

$$|q(x)| \leq 1 \quad \forall x \in [-1, 1].$$

Wir betrachten die Differenz $d(x) = q(x) - T_n(x)$. d hat nach dem Zwischenwertsatz mindestens je eine Nullstelle in den n Intervallen

$$\left[\cos \frac{k\pi}{n}, \cos \frac{(k+1)\pi}{n} \right], \quad k = 0, 1, \dots, n-1,$$

denn T_n nimmt am Rand dieser Intervalle die Werte ± 1 an. Ist $\alpha \in (-1, 1)$ eine solche Nullstelle auf dem Rand eines der Intervalle, so muss wegen $|T_n(\alpha)| = 1$ und $T'_n(\alpha) = 0$ auch $q'(\alpha) = 0$ gelten, da sonst die Bedingung $|q(x)| \leq 1$ verletzt wäre. Damit ist α doppelte Nullstelle von d . Gezählt mit Vielfachheiten hat d insgesamt n Nullstellen in $[-1, 1]$. Da d ein Polynom vom Grad $n-1$ ist (die Höchstkoeffizienten von T_n und q stimmen überein), muss $d \equiv 0$ sein, was im Widerspruch zu $q \neq T_n$ steht. \square

Das Lemma zeigt, dass das Minmax-Problem (2.3) für $[a, b] = [-1, 1]$ durch $\omega_{n+1}(x) = 2^{-n}T_{n+1}(x)$ gelöst wird, also für die Wahl

$$x_k = \cos \frac{(2k+1)\pi}{2n+2}, \quad k = 0, \dots, n,$$

bei der als Interpolationspunkte die Nullstellen von T_{n+1} gewählt werden. Die Lösung für das Minmax-Problem für beliebige Intervalle $[a, b]$ folgt nun sofort mit Hilfe einer linearen Transformation:

Satz 2.12. Der Ausdruck (2.3) wird minimal genau dann, wenn die Knoten

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

gewählt werden.

Beweis. Folgt aus dem vorigen Lemma mit der Transformation

$$x \mapsto \frac{a+b}{2} + \frac{b-a}{2}x$$

des Intervalls $[-1, 1]$ auf $[a, b]$. \square

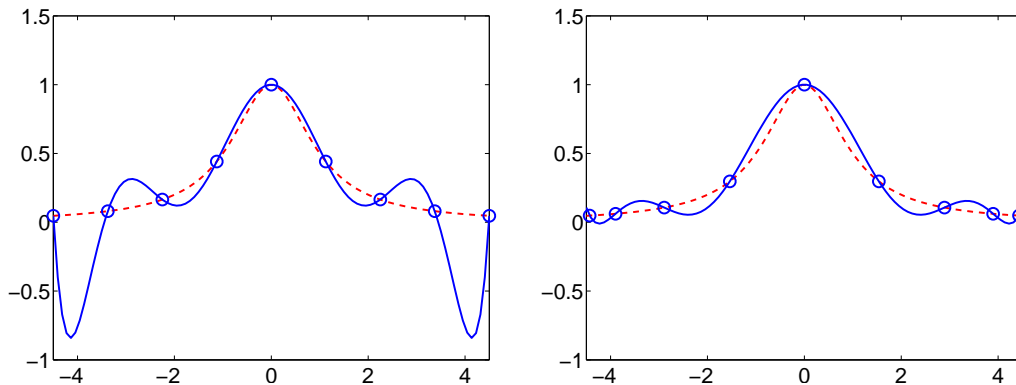


Abb. 2.3: Interpolation der Funktionen $f(x) = 1/(1+x^2)$ in äquidistanten Knoten (links) und in Tschebyscheff-Knoten (rechts).

Satz 2.13. Für die Lebesgue-Konstanten zu den Tschebyscheff-Knoten gilt

$$\Lambda_n \leq 3, \quad n \leq 20, \quad \Lambda_n \leq 4, \quad n \leq 100, \quad \Lambda_n \sim \frac{2}{\pi} \log n, \quad n \rightarrow \infty.$$

Beweis. Für $n \leq 100$ kann man die Abschätzungen explizit nachrechnen. Auf einen Beweis der Asymptotik verzichten wir hier. \square

Beispiel. Als Beispiel betrachten wir noch einmal die Funktion $f(x) = 1/(1+x^2)$ im Intervall $[-4.5, 4.5]$. In Abbildung 2.3 vergleichen wir für $n = 8$ die Interpolationspolynome zu äquidistanten Knoten und zu Tschebyscheff-Knoten. Man erkennt eine wesentliche Verbesserung der Approximation durch die Wahl von Tschebyscheff-Knoten. \diamond

Weitere Eigenschaften von Tschebyscheff-Polynomen erlauben eine effiziente Berechnung und Auswertung des Interpolationspolynoms.

Lemma 2.14. Die Tschebyscheff-Polynome T_0, T_1, \dots sind orthogonal bezüglich des Skalarprodukts

$$\langle f, g \rangle = \int_{-1}^1 \frac{1}{\sqrt{1-x^2}} f(x)g(x)dx.$$

Beweis. Übung. \square

Noch wichtiger ist die Orthogonalität bezüglich eines diskreten Skalarprodukts:

Lemma 2.15. Die Tschebyscheff-Polynome T_0, T_1, \dots, T_n sind orthogonal bezüglich des Skalarprodukts

$$\langle f, g \rangle = \sum_{j=0}^n f(x_j)g(x_j)$$

definiert auf \mathcal{P}_n , wobei x_j die Nullstellen von T_{n+1} sind.

Beweis. Nach Lemma 2.10 ist

$$x_j = \cos \frac{(2j+1)\pi}{2(n+1)} = \cos \left[\left(j + \frac{1}{2}\right)h \right], \quad h = \frac{\pi}{n+1}.$$

Daher gilt

$$\begin{aligned}
 \langle T_k, T_m \rangle &= \sum_{j=0}^n T_k(x_j) T_m(x_j) \\
 &= \sum_{j=0}^n \cos \left[k \left(j + \frac{1}{2} \right) h \right] \cos \left[m \left(j + \frac{1}{2} \right) h \right] \\
 &= \frac{1}{2} \sum_{j=0}^n \left\{ \cos \left((k-m) \left(j + \frac{1}{2} \right) h \right) + \cos \left((k+m) \left(j + \frac{1}{2} \right) h \right) \right\} \\
 &= \frac{1}{2} \operatorname{Re} \left\{ \sum_{j=0}^n \left(e^{i(k-m)(j+\frac{1}{2})h} + e^{i(k+m)(j+\frac{1}{2})h} \right) \right\} \\
 &= \begin{cases} 0, & k \neq m, \\ \frac{1}{2}(n+1), & k = m \neq 0, \\ n+1, & k = m = 0, \end{cases} \quad 0 \leq k, m \leq n,
 \end{aligned}$$

mit Hilfe der Formel für die geometrische Reihe, denn $e^{ih(n+1)} = -1$ und $k+m$ ist genau dann gerade, wenn $k-m$ gerade ist. \square

Als Folgerung aus dieser Orthogonalitätsrelation ergibt sich

Satz 2.16. *Das Interpolationspolynom p zu f in den Tschebyscheff-Knoten x_0, \dots, x_n (den Nullstellen von T_{n+1}) ist gegeben durch*

$$p(x) = \frac{1}{2}c_0 + c_1T_1(x) + \dots + c_nT_n(x) \quad (2.4)$$

mit

$$c_k = \frac{2}{n+1} \sum_{j=0}^n f(x_j) \cos \left[k \left(j + \frac{1}{2} \right) h \right], \quad k = 0, 1, \dots, n.$$

Beweis. Wegen der Orthogonalität gilt

$$\langle p, T_k \rangle = \frac{1}{2}c_0 \langle T_0, T_k \rangle + c_1 \langle T_1, T_k \rangle + \dots + c_n \langle T_n, T_k \rangle = c_k \frac{n+1}{2}, \quad k = 0, \dots, n.$$

Somit ist

$$c_k = \frac{2}{n+1} \sum_{j=0}^n p(x_j) T_k(x_j) = \frac{2}{n+1} \sum_{j=0}^n f(x_j) \cos \left[k \left(j + \frac{1}{2} \right) h \right],$$

wobei wir die Interpolationseigenschaft und die Definition von T_k verwendet haben. \square

Bemerkung. Für hinreichend große n ($n \geq 15$) ist es zweckmäßig, die Koeffizienten c_k mit schneller Fouriertransformation (FFT) zu berechnen. Diese ist besonders günstig, falls $n+1$ eine Zweierpotenz ist. Die direkte Berechnung von c_0, \dots, c_n kostet $(n+1)^2$ Multiplikationen und Additionen, mit FFT jedoch nur $O(n \log n)$. Die Berechnung der dividierten Differenzen im Newton-Schema kostet $n(n+1)/2$ Divisionen.

Bei bekannten Entwicklungskoeffizienten c_j lässt sich $p(x)$ leicht auswerten:

Satz 2.17. (Clenshaw-Algorithmus)

Es sei $p \in \mathcal{P}_n$ ein beliebiges Polynom gegeben in der Darstellung (2.4) und $x \in \mathbb{R}$ fest gewählt. Setzt man

$$d_k = c_k + 2xd_{k+1} - d_{k+2}, \quad k = n, n-1, \dots, 0, \quad d_{n+2} = d_{n+1} = 0,$$

dann ist $p(x) = \frac{1}{2}(d_0 - d_2)$.

Beweis. Der Beweis basiert auf der Rekursionsformel für Tschebyscheff-Polynome aus Lemma 2.10(a). Mit $c_n = d_n$ gilt

$$\begin{aligned} p(x) &= \frac{1}{2}c_0 + c_1T_1(x) + \dots + c_{n-1}T_{n-1}(x) + d_nT_n(x) \\ &= \frac{1}{2}c_0 + c_1T_1(x) + \dots + (c_{n-2} - d_n)T_{n-2}(x) + \underbrace{(c_{n-1} + 2xc_n)}_{d_{n-1}}T_{n-1}(x) \\ &= \frac{1}{2}c_0 + c_1T_1(x) + \dots + (c_{n-3} - d_{n-1})T_{n-3}(x) + \underbrace{(c_{n-2} - d_n + 2xd_{n-1})}_{d_{n-2}}T_{n-2}(x) \\ &= \dots \\ &= \frac{1}{2}c_0 - d_2 + \underbrace{(c_1 - d_3 + 2xd_2)}_{d_1} \underbrace{T_1(x)}_x \\ &= \frac{1}{2}c_0 - d_2 + xd_1 \\ &= \frac{1}{2} \left[\underbrace{(c_0 + 2xd_1 - d_2)}_{d_0} - d_2 \right] \\ &= \frac{1}{2}(d_0 - d_2), \end{aligned}$$

wobei wir die Rekursion im ersten Schritt für T_n , dann für T_{n-1} , T_{n-2} , usw. eingesetzt haben. \square

Bei Verwendung von Rekursionen ist wichtig, wie sich Fehler (zum Beispiel Rundungsfehler) fortpflanzen.

Beispiel. Startet man die Rekursion $x_{n+1} = 10x_n - 9$ mit $x_0 = 1$, so ist $x_n = 1$ für alle n . Startet man jedoch mit einem leicht gestörten $\hat{x}_0 = 1 + \epsilon$, so ist $\hat{x}_n = 1 + 10^n\epsilon$. Der Fehler wird also in jedem Schritt um den Faktor 10 verstärkt. \diamond

Der Clenshaw-Algorithmus ist stabil:

Lemma 2.18. Betrachtet man zum Clenshaw-Algorithmus die gestörte Rekursion

$$\hat{d}_k = c_k + 2x\hat{d}_{k+1} - \hat{d}_{k+2} + \epsilon_k, \quad k = n, n-1, \dots, 0, \quad \hat{d}_{n+2} = \hat{d}_{n+1} = 0$$

und setzt $\hat{p}(x) = \frac{1}{2}(\hat{d}_0 - \hat{d}_2)$, dann gilt

$$|\hat{p}(x) - p(x)| \leq \frac{1}{2}|\epsilon_0| + \sum_{j=1}^n |\epsilon_j| \quad \text{für} \quad |x| \leq 1.$$

Beweis. Es sei $e_k = \hat{d}_k - d_k$. Wegen der Linearität gilt

$$e_k = \epsilon_k + 2xe_{k+1} - e_{k+2}, \quad k = n, n-1, \dots, 0, \quad e_{n+2} = e_{n+1} = 0.$$

Somit ist nach Satz 2.17

$$\hat{p}(x) - p(x) = \frac{1}{2}(e_0 - e_2) = \frac{1}{2}\epsilon_0T_0(x) + \epsilon_1T_1(x) + \dots + \epsilon_nT_n(x)$$

und die Behauptung folgt aus $|T_j(x)| \leq 1$ für $|x| \leq 1$. \square

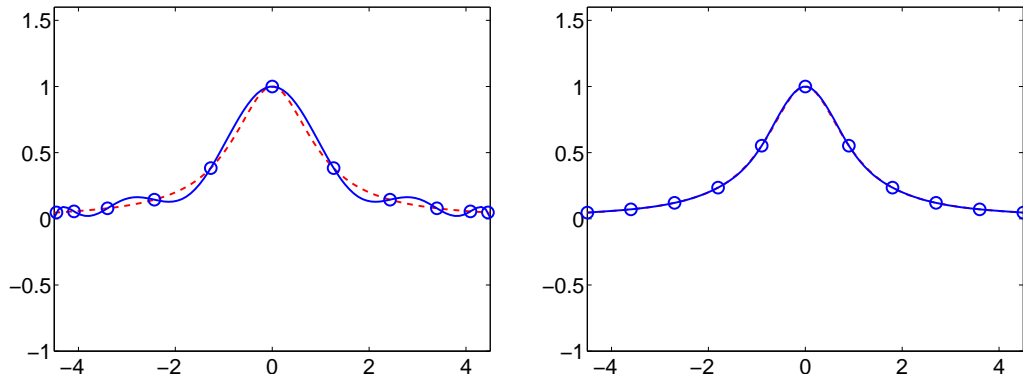


Abb. 2.4: Vergleich von Tschebyscheff-Interpolation und Spline-Interpolation am Beispiel der Funktion $f(x) = 1/(1+x^2)$.

Approximationen durch Summen von Tschebyscheff-Polynomen werden im Rechner zu Berechnung von Funktionen wie \log , \exp , \sin , \cos , ... verwendet.

2.4 Splineinterpolation

Wir haben in Abschnitt 2.2 gesehen, dass Polynominterpolation mit sehr vielen Knoten nicht geeignet ist, um eine Funktion gut zu approximieren. Als Alternative stellen wir jetzt die Splineinterpolation vor, bei der das Ziel ist, eine möglichst "glatte" Funktion s (einen "Spline") zu bestimmen, die durch beliebig vorgegebene Punkte (x_i, y_i) geht. Anschaulich kann man sich darunter eine elastische Feder vorstellen, die man an diesen Punkten festklemmt. Mathematisch formuliert suchen wir eine Funktion $s : [a, b] \rightarrow \mathbb{R}$ mit

$$(a) \quad s(x_i) = y_i, \quad i = 0, \dots, n,$$

$$(b) \quad s \in C^2([a, b]),$$

$$(c) \quad \int_a^b (s''(x))^2 dx \rightarrow \min.$$

Das Integral in (c) ist ein Maß für die Deformationsenergie der Feder und eine Approximation an die Krümmung, die durch

$$\int_a^b \frac{s''(x)}{\sqrt{1 + s'(x)^2}} dx$$

definiert ist. Durch diese Bedingung vermeidet man Oszillationen, wie sie bei der Polynominterpolation mit äquidistanten Stützstellen auftreten.

Als ein erstes Beispiel betrachten wir in Abbildung 2.4 die Interpolation der Funktion $f(x) = 1/(1+x^2)$, bei der wir schon im letzten Abschnitt gesehen haben, dass Interpolation in äquidistanten Knoten schlechte Ergebnisse liefert und Interpolation in Tschebyscheff-Knoten zu bevorzugen ist. Splines eignen sich auch sehr gut zum Glätten von Kurven, wie zum Beispiel der in Abbildung 2.5.

Setzen wir $f(x) = s(x) + \varepsilon h(x)$, $\varepsilon \in \mathbb{R}$ und

$$h \in C^2([a, b]), \quad h(x_i) = 0, \quad i = 0, \dots, n. \quad (2.5)$$



Abb. 2.5: Glättung durch Splines (©Daten von Gerhard Wanner)

Dann ist die Minimierungseigenschaft (c) wegen

$$\begin{aligned} \int_a^b (s''(x))^2 dx &\leq \int_a^b (s''(x) + \varepsilon h''(x))^2 dx \\ &= \int_a^b (s''(x))^2 dx + 2\varepsilon \int_a^b s''(x)h''(x) dx + \varepsilon^2 \int_a^b (h''(x))^2 dx \end{aligned}$$

genau dann erfüllt, wenn

$$\int_a^b s''(x)h''(x) dx = 0$$

für alle h , die (2.5) erfüllen. Ist s sogar 3-mal stetig differenzierbar auf jedem Teilintervall $[x_{j-1}, x_j]$, dann folgt mit partieller Integration

$$\int_a^b s''(x)h''(x) dx = s''(x)h'(x) \Big|_a^b - \int_a^b s'''(x)h'(x) dx = 0 \quad (2.6)$$

Satz 2.19. Seien $s, f : [a, b] \rightarrow \mathbb{R}$ zweimal stetig differenzierbare Funktionen, die in $a = x_0 < x_1 < \dots < x_n = b$ dieselben Werte annehmen

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

Ist $s|_{[x_{i-1}, x_i]}$ ein Polynom vom Grad drei, $i = 1, \dots, n$ und gilt

$$s''(a)[f'(a) - s'(a)] = s''(b)[f'(b) - s'(b)], \quad (2.7)$$

so ist

$$\int_a^b (s''(x))^2 dx \leq \int_a^b (f''(x))^2 dx.$$

Beweis. Setzen wir $h(x) = f(x) - s(x)$, so erfüllt h die Bedingung (2.5). Aus (2.7) folgt, dass der erste Term in (2.6) verschwindet. Da s''' nach Voraussetzung auf jedem Teilintervall $[x_{i-1}, x_i]$ konstant ist, sagen wir identisch α_i , wird der zweite Term in (2.6) zu

$$\int_a^b s'''(x)h'(x) dx = \sum_{i=1}^n \alpha_i \int_{x_{i-1}}^{x_i} h'(x) dx = \sum_{i=1}^n \alpha_i (h(x_i) - h(x_{i-1})) = 0$$

wegen der Interpolationseigenschaft (2.5) von h . Aus den Vorüberlegungen folgt daraus die Minimierungseigenschaft. \square

Definition 2.20. (*Kubischer Spline*)

Eine Funktion $s : [a, b] \rightarrow \mathbb{R}$ heißt **kubischer Spline**, falls

$$s \in C^2([a, b]), \quad s_i := s|_{[x_{i-1}, x_i]} \in \mathcal{P}_3, \quad i = 1, \dots, n,$$

für $a = x_0 < x_1 < \dots < x_n = b$.

Die Bedingung (2.7) kann auf verschiedene Arten erreicht werden:

- (a) Natürlicher Spline: $s''(a) = s''(b) = 0$
- (b) Eingespannter Spline: $s'(a) = v_0, s'(b) = v_n$ vorgegeben
- (c) Periodischer Spline: $s'(a) = s'(b)$ und $s''(a) = s''(b)$

Für den eingespannten Spline ist die Bedingung (2.7) natürlich nur erfüllt, wenn für f' ebenfalls $f'(a) = v_0$ und $f'(b) = v_n$ gilt. Daher gilt die Minimierungseigenschaft von s nur in der Menge aller Funktionen mit diesen Randbedingungen. Analog ist die Situation für die periodischen Splines.

Wie konstruiert man kubische Splines? Auf dem Teilintervall $[x_{i-1}, x_i]$ der Länge $h_i = x_i - x_{i-1}$ suchen wir ein Polynom $s_i \in \mathcal{P}_3$ mit

$$\begin{aligned} s_i(x_{i-1}) &= y_{i-1} & s'_i(x_{i-1}) &= v_{i-1} \\ s_i(x_i) &= y_i & s'_i(x_i) &= v_i \end{aligned}$$

mit zunächst unbekannten Werten v_{i-1}, v_i . Die Interpolationsaufgabe, bei der Funktionswerte und Ableitungen in gegebenen Stützstellen vorgegeben werden, heißt Hermite-Interpolation. Das Polynom s_i kann man mit der Newton'schen Interpolationsformel berechnen, wenn man die Bedingungen an die Ableitung durch

$$s_i(x_{i-1} + \epsilon) = y_{i-1} + \epsilon v_{i-1} + O(\epsilon^2), \quad s_i(x_i - \epsilon) = y_i - \epsilon v_i + O(\epsilon^2)$$

ersetzt und den Grenzübergang $\epsilon \rightarrow 0$ betrachtet. Das Tableau der dividierten Differenzen hat dann die Form

x_{i-1}	y_{i-1}			
x_{i-1}	y_{i-1}	v_{i-1}		
x_i	y_i	$\delta y[x_{i-1}, x_i]$	$(\delta y[x_{i-1}, x_i] - v_{i-1})/h_i$	
x_i	y_i	v_i	$(v_i - \delta y[x_{i-1}, x_i])/h_i$	$(v_i + v_{i-1} - 2\delta y[x_{i-1}, x_i])/h_i^2$

und für s_i ergibt sich daraus

$$\begin{aligned} s_i(x) &= y_{i-1} + (x - x_{i-1})v_{i-1} + (x - x_{i-1})^2 \frac{\delta y[x_{i-1}, x_i] - v_{i-1}}{h_i} \\ &\quad + (x - x_{i-1})^2 (x - x_i) \frac{v_i + v_{i-1} - 2\delta y[x_{i-1}, x_i]}{h_i^2}. \end{aligned} \tag{2.8}$$

Für beliebige v_0, \dots, v_n erhalten wir eine Funktion $s : [a, b] \rightarrow \mathbb{R}$ mit

$$s \in C^1([a, b]), \quad s(x_i) = y_i, \quad s|_{[x_{i-1}, x_i]} = s_i \in \mathcal{P}_3, \quad i = 0, \dots, n.$$

Wir wählen jetzt v_0, \dots, v_n so, dass s sogar in $C^2([a, b])$ ist, also

$$s_i''(x_i) = s_{i+1}''(x_i), \quad i = 1, \dots, n-1 \quad (2.9)$$

gilt. Im Falle des eingespannten Splines sind v_0 und v_n gegeben, so dass wir $n-1$ Bedingungen für die $n-1$ Unbekannten v_1, \dots, v_{n-1} haben. Aus (2.8) folgt

$$\begin{aligned} s_i''(x_i) &= \frac{2}{h_i} (2v_i + v_{i-1} - 3\delta y[x_{i-1}, x_i]), \\ s_{i+1}''(x_i) &= -\frac{2}{h_{i+1}} (2v_i + v_{i+1} - 3\delta y[x_i, x_{i+1}]). \end{aligned}$$

Die Bedingung (2.9) ist damit äquivalent zu

$$\frac{v_{i-1}}{h_i} + 2 \left(\frac{1}{h_i} + \frac{1}{h_{i+1}} \right) v_i + \frac{v_{i+1}}{h_{i+1}} = 3 \left(\frac{\delta y[x_{i-1}, x_i]}{h_i} + \frac{\delta y[x_i, x_{i+1}]}{h_{i+1}} \right), \quad i = 1, \dots, n-1.$$

Die unbekannten Werte v_i sind also Lösung des linearen Gleichungssystems

$$\begin{bmatrix} 2 \left(\frac{1}{h_1} + \frac{1}{h_2} \right) & \frac{1}{h_2} & & & \\ \frac{1}{h_2} & 2 \left(\frac{1}{h_2} + \frac{1}{h_3} \right) & \frac{1}{h_3} & & \\ & \frac{1}{h_3} & 2 \left(\frac{1}{h_3} + \frac{1}{h_4} \right) & \ddots & \\ & & \ddots & \ddots & \frac{1}{h_{n-1}} \\ & & & \frac{1}{h_{n-1}} & 2 \left(\frac{1}{h_{n-1}} + \frac{1}{h_n} \right) \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_{n-2} \\ v_{n-1} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{n-2} \\ b_{n-1} \end{bmatrix},$$

wobei

$$b_i = 3 \left(\frac{\delta y[x_{i-1}, x_i]}{h_i} + \frac{\delta y[x_i, x_{i+1}]}{h_{i+1}} \right) - \begin{cases} 0, & i = 2, \dots, n-2, \\ \frac{v_0}{h_1}, & i = 1, \\ \frac{v_n}{h_n}, & i = n-1. \end{cases}$$

Die Koeffizientenmatrix, die wir mit A bezeichnen, ist eine symmetrische Tridiagonalmatrix.

Satz 2.21. *Es sei $Av = b$. Dann gilt*

$$\max_i |v_i| \leq \frac{h}{2} \max_i |b_i|, \quad h = \max_i h_i.$$

Insbesondere ist A invertierbar.

Beweis. Wir wählen einen Index j so aus, dass $|v_j| = \max_i |v_i|$ gilt. Aus $Av = b$ folgt dann

$$2 \left(\frac{1}{h_j} + \frac{1}{h_{j+1}} \right) v_j = -\frac{v_{j-1}}{h_j} - \frac{v_{j+1}}{h_{j+1}} + b_j$$

und mit Hilfe der Dreiecksungleichung

$$2 \left(\frac{1}{h_j} + \frac{1}{h_{j+1}} \right) |v_j| \leq \left(\frac{1}{h_j} + \frac{1}{h_{j+1}} \right) |v_j| + |b_j|.$$

Dies liefert

$$\frac{2}{h} |v_j| \leq \left(\frac{1}{h_j} + \frac{1}{h_{j+1}} \right) |v_j| \leq |b_j| \leq \max_i |b_i|.$$

Setzt man $b = 0$, so folgt aus der Abschätzung $v = 0$. Das homogene lineare Gleichungssystem hat also nur die triviale Lösung. Daher ist A invertierbar. \square

Die Invertierbarkeit von A zeigt, dass der eingespannte kubische Spline existiert und eindeutig ist.

Beispiel. Für eine äquidistante Unterteilung, d. h. $h_i = h$ für alle i , vereinfacht sich nach Multiplikation mit h die Koeffizientenmatrix A zu

$$hA = \text{tridiag}(1, 4, 1) = \begin{bmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & 1 & 4 \end{bmatrix}$$

und der Vektor $b = (b_i)$ zu

$$hb_i = \frac{3}{h}(y_{i+1} - y_{i-1}) - \begin{cases} 0, & i = 2, \dots, n-2, \\ v_0, & i = 1, \\ v_n, & i = n-1. \end{cases}$$

Die Lösung dieses Gleichungssystems kann durch Elimination erfolgen. Dabei ergibt sich aus

$$\begin{bmatrix} 4 & 1 & & & \\ 0 & d_2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 0 & d_{n-2} & 1 \\ & & & 0 & d_{n-1} \end{bmatrix} \begin{bmatrix} v_1 \\ \vdots \\ v_{n-1} \end{bmatrix} = \begin{bmatrix} c_1 \\ \vdots \\ c_{n-1} \end{bmatrix}$$

mit

$$\begin{aligned} d_1 &= 4, & c_1 &= b_1 \\ d_{k+1} &= 4 - \frac{1}{d_k}, & c_{k+1} &= b_{k+1} - \frac{c_k}{d_k}, \quad k = 1, \dots, n-2 \end{aligned}$$

die Lösung

$$v_{n-1} = \frac{c_{n-1}}{d_{n-1}}, \quad v_k = \frac{c_k - v_{k+1}}{d_k}, \quad k = n-2, \dots, 1$$

in $O(n)$ arithmetischen Operationen. \diamond

2.5 Fehler bei der Splineinterpolation

Wir beschränken uns hier auf die Fehleranalyse für eingespannte Splines.

Die Konstruktion kubischer Splines verwendete als wesentliches Hilfsmittel die Hermite-Interpolation auf einem Teilintervall $[x_{i-1}, x_i]$. Daher liegt es nahe, zunächst den Fehler dieser Interpolation zu untersuchen, bevor man den des Splines analysiert.

Satz 2.22. Sei $f \in C^4([x_{i-1}, x_i])$ gegeben und sei $q_i \in \mathcal{P}_3$ das Hermite-Interpolationspolynom zu den Daten

$$\begin{aligned} q_i(x_{i-1}) &= f(x_{i-1}), & q'_i(x_{i-1}) &= f'(x_{i-1}), \\ q_i(x_i) &= f(x_i), & q'_i(x_i) &= f'(x_i). \end{aligned}$$

Dann gilt für $x \in [x_{i-1}, x_i]$

$$|f(x) - q_i(x)| \leq \frac{h_i^4}{384} \max_{\xi \in [x_{i-1}, x_i]} |f^{(4)}(\xi)|.$$

Beweis. Wir verwenden den Fehler des Newton'schen Interpolationspolynoms vom Grad 3 durch die Punkte

$$(x_{i-1}, f(x_{i-1})), \quad (x_i, f(x_i)), \quad (x_{i-1} + \epsilon, f(x_{i-1} + \epsilon)), \quad (x_i - \epsilon, f(x_i - \epsilon)).$$

Nach Satz 2.5 gilt im Grenzfall $\epsilon \rightarrow 0$

$$|f(x) - q_i(x)| \leq (x - x_{i-1})^2 (x - x_i)^2 \frac{1}{4!} \max_{\xi \in [x_{i-1}, x_i]} |f^{(4)}(\xi)|.$$

Die Funktion $(x - x_{i-1})^2 (x - x_i)^2$ nimmt ihr Maximum in der Mitte des Intervalls an und hat dort den Wert $h_i^4/2^4$. Die behauptete Abschätzung folgt aus $4!2^4 = 384$. \square

Den Gesamtfehler des Splines $f - s$ betrachten wir auf dem i -ten Teilintervall:

$$f - s_i = (f - q_i) + (q_i - s_i).$$

Den ersten Term haben wir im letzten Satz abgeschätzt, den zweiten können wir mit Hilfe von (2.8) bzw. der analogen Darstellung für q_i als

$$q_i(x) - s_i(x) = \frac{(x - x_{i-1})(x - x_i)}{h_i^2} \left[(f'(x_i) - v_i)(x - x_{i-1}) + (f'(x_{i-1}) - v_{i-1})(x - x_i) \right] \quad (2.10)$$

schreiben (Übung). Wir benötigen also nur noch eine Abschätzung für $f'(x_i) - v_i$.

Lemma 2.23. *Es sei $f \in C^4([a, b])$ und $h = \max_i h_i$. v_1, \dots, v_{n-1} seien Lösung des linearen Gleichungssystems aus dem vorigen Abschnitt und es gelte $v_0 = f'(a)$, $v_n = f'(b)$. Dann gilt*

$$|f'(x_i) - v_i| \leq \frac{h^3}{24} \max_{x \in [a, b]} |f^{(4)}(x)|.$$

Falls sogar $h_i = h$ für alle i und $f \in C^5([a, b])$ gilt

$$|f'(x_i) - v_i| \leq \frac{h^4}{60} \max_{x \in [a, b]} |f^{(5)}(x)|.$$

Beweis. Wir beweisen nur den äquidistanten Fall. Der allgemeine Fall lässt sich völlig analog mit erheblich höherem technischem Aufwand beweisen.

Im Spezialfall $h_i = h$ sind die Werte v_i durch

$$\frac{1}{h}(v_{i-1} + 4v_i + v_{i+1}) - \frac{3}{h^2}(f(x_{i+1}) - f(x_{i-1})) = 0$$

definiert. Wir definieren daher den Defekt

$$d_i := \frac{1}{h}(f'(x_{i-1}) + 4f'(x_i) + f'(x_{i+1})) - \frac{3}{h^2}(f(x_{i+1}) - f(x_{i-1})).$$

Der Satz von Taylor mit Restglied in Integraldarstellung liefert

$$f(x_i \pm h) = \sum_{m=0}^4 f^{(m)}(x_i) \frac{(\pm h)^m}{m!} \pm h^5 \int_0^1 \frac{(1-t)^4}{4!} f^{(5)}(x_i \pm th) dt,$$

$$f'(x_i \pm h) = \sum_{m=0}^3 f^{(m+1)}(x_i) \frac{(\pm h)^m}{m!} + h^4 \int_0^1 \frac{(1-t)^3}{3!} f^{(5)}(x_i \pm th) dt.$$

Hieraus folgt mit

$$\begin{aligned} 3(f(x_{i+1}) - f(x_{i-1})) &= 6hf'(x_i) + h^3 f^{(3)}(x_i) \\ &\quad + h^5 \int_0^1 \frac{3(1-t)^4}{4!} [f^{(5)}(x_i + th) + f^{(5)}(x_i - th)] dt, \\ f'(x_{i-1}) + f'(x_{i+1}) &= 2f'(x_i) + h^2 f^{(3)}(x_i) \\ &\quad + h^4 \int_0^1 \frac{(1-t)^3}{3!} [f^{(5)}(x_i + th) + f^{(5)}(x_i - th)] dt \end{aligned}$$

für den Defekt

$$d_i = h^3 \int_0^1 \left(\frac{(1-t)^3}{3!} - 3 \frac{(1-t)^4}{4!} \right) [f^{(5)}(x_i + th) + f^{(5)}(x_i - th)] dt.$$

Da $\frac{(1-t)^3}{3!} - 3 \frac{(1-t)^4}{4!} \geq 0$ für alle $t \in [0, 1]$, erhält man die Abschätzung

$$|d_i| \leq 2h^3 \int_0^1 \left(\frac{(1-t)^3}{3!} - 3 \frac{(1-t)^4}{4!} \right) dt \max_{x \in [x_{i-1}, x_{i+1}]} |f^{(5)}(x)| = \frac{h^3}{30} \max_{x \in [x_{i-1}, x_{i+1}]} |f^{(5)}(x)|.$$

Nach Definition der Defekte $d = (d_i)_{i=1}^{n-1}$ und wegen $Av = b$ löst $w = (w_i) = (f'(x_i) - v_i)$ das lineare Gleichungssystem $Aw = d$. Satz 2.21 impliziert

$$|w_i| \leq \frac{h}{2} \max_i |d_i| \leq \frac{h^4}{60} \max_{x \in [a, b]} |f^{(5)}(x)|.$$

Damit ist die Fehlerabschätzung bewiesen. □

Satz 2.24. Sei $f : [a, b] \rightarrow \mathbb{R}$, $f \in C^4$ gegeben. s sei der eingespannte kubische Spline durch die Punkte $(x_i, f(x_i))$, $i = 0, \dots, n$ mit $s'(a) = f'(a)$ und $s'(b) = f'(b)$. Dann gilt

$$|f(x) - s(x)| \leq \frac{5}{384} h^4 \max_{\xi \in [a, b]} |f^{(4)}(\xi)|, \quad x \in [a, b].$$

Falls sogar $h_i = h$ für alle i und $f \in C^5([a, b])$ gilt

$$|f(x) - s(x)| \leq \frac{h^4}{384} \max_{\xi \in [a, b]} |f^{(4)}(\xi)| + \frac{h^5}{240} \max_{\xi \in [a, b]} |f^{(5)}(\xi)|, \quad x \in [a, b].$$

Beweis. Auch hier beweisen wir nur die Aussage für den äquidistanten Fall. Mit Hilfe von Lemma 2.23 erhält man aus (2.10) für $x \in [x_{i-1}, x_i]$

$$|q_i(x) - s_i(x)| \leq \frac{(x - x_{i-1})(x_i - x)}{h^2} \frac{h^5}{60} \max_{\xi \in [a, b]} |f^{(5)}(\xi)| \leq \frac{h^5}{4 \cdot 60} \max_{\xi \in [a, b]} |f^{(5)}(\xi)|.$$

Der erste Term der Abschätzung wurde in Satz 2.22 gezeigt. □

