

Kapitel 3

Interpolation

3.1 Einführung

Bemerkung 3.1 *Motivation, Aufgabenstellung.* Gegeben seien eine Funktion $f \in C([a, b])$ und $x_i \in [a, b]$, $i = 0, \dots, n$, mit

$$a \leq x_0 < x_1 < \dots < x_n \leq b. \quad (3.1)$$

Die Interpolationsaufgabe besteht darin, eine (einfache) Funktion $u_n \in U_n$ zu finden, so dass

$$u_n(x_i) = f(x_i), \quad i = 0, \dots, n, \quad (3.2)$$

ist. Dabei ist $U_n \subset C([a, b])$ eine Folge von Räumen.

Natürlich gibt es unendlich viele Funktionen, welche die Interpolationsaufgabe erfüllen. Deshalb muss man die Klasse der Funktionen festlegen, in der man die Interpolierende $u_n(x)$ sucht. Man unterscheidet zum Beispiel:

- $u_n(x)$ ist ein Polynom – Polynominterpolation,
- $u_n(x) = a_0 + a_1 e^{ix} + \dots + a_n e^{inx}$ – trigonometrische Interpolation ($i = \sqrt{-1}$),
- $u_n(x)$ ist stückweise ein Polynom – Spline-Interpolation,
- $u_n(x)$ ist eine rationale Funktion – rationale Interpolation.

In diesem Kapitel wird auf die Interpolation mit Polynomen und mit Splines eingegangen. \square

Definition 3.2 Stützstellen, Stützwerte. Man bezeichnet $\{x_i\}$ als die Menge der Stützstellen und die Menge $\{f(x_i)\}$ als die Menge der Stützwerte. Man sagt, dass die Funktion $u_n(x)$ die Stützwerte $\{f(x_i)\}$ an den Stützstellen $\{x_i\}$ interpoliert. \square

Bemerkung 3.3 Wichtige Fragestellungen. Natürlich sind die Untersuchung der Existenz und Eindeutigkeit einer Lösung von (3.2) von Bedeutung.

Die Interpolationsaufgabe (3.2) ist auch eine Art Bestapproximations-Aufgabe, nämlich gerade für die Stützstellen $\{x_i\}_{i=0}^n$. Es ist aber auch wichtig zu wissen, wie sich die Interpolierende $u_n(x)$ und die interpolierte Funktion $f(x)$ in den Intervallen zwischen den Stützstellen unterscheiden, zum Beispiel ob

$$\lim_{n \rightarrow \infty} \|f - u_n\|_{\infty} = 0. \quad (3.3)$$

Falls dies der Fall ist, interessiert man sich auch für die Konvergenzgeschwindigkeit. Dazu versucht man Fehlerabschätzungen der Gestalt

$$\|f - u_n\|_{\infty} \leq Cn^{-q} \quad (3.4)$$

mit einer von n unabhängigen Konstanten $C > 0$ und einer möglichst großen positiven Konstanten q zu beweisen. Hat man eine Abschätzung der Gestalt (3.4) bewiesen, so erhält man damit auch eine Abschätzung für die L^2 -Norm

$$\begin{aligned} \|f - u_n\|_{L^2} &= \left(\int_a^b (f - u_n)^2(x) dx \right)^{1/2} \\ &\leq \|f - u_n\|_{\infty} \left(\int_a^b dx \right)^{1/2} \\ &= \sqrt{b-a} \|f - u_n\|_{\infty} \\ &\leq Cn^{-q}, \end{aligned}$$

wobei C jetzt eine andere Konstante bezeichnet. Eine Umkehrung dieser Vorgehensweise gilt nicht. Aus dem Fehler in der L^2 -Norm kann man nicht auf den Fehler in der Maximumsnorm schließen.

Für die Praxis sind schließlich auch effiziente und robuste Verfahren zur Berechnung der Interpolierenden wichtig. \square

3.2 Polynominterpolation

3.2.1 Wiederholung

Bemerkung 3.4 Aufgabenstellung. Sei $U_n = P_n$. Die allgemeine Interpolationsaufgabe (3.2) hat in diesem Fall die Gestalt: Finde $u_n = p_n \in P_n$, so dass

$$p_n(x_i) = f(x_i), \quad i = 0, \dots, n, \quad (3.5)$$

unter der Voraussetzung (3.1).

Diese Aufgabe wurde bereits in CoMa 2 untersucht. Die gewonnenen Ergebnisse werden hier zusammengefasst. \square

Definition 3.5 Dividierte Differenz. Seien $(x_i, f(x_i)) \in \mathbb{R} \times \mathbb{R}$, $i = 0, \dots, n$, paarweise verschiedene Stützstellen. Die k -te dividierte Differenz

$$f[x_i, x_{i+1}, \dots, x_{i+k}]$$

wird rekursiv definiert durch

$$\begin{aligned} f[x_i] &= f(x_i) \quad \text{für } i = 0, \dots, n, \\ f[x_i, x_{i+1}, \dots, x_{i+k}] &= \frac{f[x_{i+1}, x_{i+2}, \dots, x_{i+k}] - f[x_i, x_{i+1}, \dots, x_{i+k-1}]}{x_{i+k} - x_i}. \end{aligned}$$

\square

Satz 3.6 Zur Polynominterpolation. Die Interpolationsaufgabe (3.5) besitzt eine eindeutige Lösung. Diese hat die Lagrange¹-Darstellung

$$p_n(x) = \sum_{k=0}^n L_k(x) f(x_k), \quad (3.6)$$

wobei $L_k(x)$ das Lagrange-Polynom

$$L_k(x) = \prod_{j=0, j \neq k}^n \frac{x - x_j}{x_k - x_j} \quad (3.7)$$

¹Joseph Louis Lagrange (1736 – 1813)

ist. Die Newton²sche Darstellung des Interpolationspolynoms lautet

$$p_n(x) = \sum_{k=0}^n \omega_k(x) f[x_0, x_1, \dots, x_k] \quad \text{mit} \quad \omega_k(x) = \prod_{j=0}^{k-1} (x - x_j). \quad (3.8)$$

Sei $f \in C^{n+1}([a, b])$. Dann gibt es zu jedem $x \in [a, b]$ ein $\xi(x) \in (a, b)$, so dass für den Fehler

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega_{n+1}(x). \quad (3.9)$$

gilt. Daraus folgt

$$\|f - p_n\|_\infty \leq \frac{1}{(n+1)!} \|f^{(n+1)}\|_\infty \|\omega_{n+1}\|_\infty. \quad (3.10)$$

Bemerkung 3.7 *Interpretation der Aussage des Satzes.* Die Aussage von Satz 3.6 impliziert nicht, dass (3.3) gilt. Man kann Funktionen und zugehörige Mengen von Stützstellen so finden, zum Beispiel mit gleichem Abstand, dass es Teilintervalle von $[a, b]$ gibt, in denen $p_n(x) \not\rightarrow f(x)$ für alle Argumente x aus diesen Teilintervallen. \square

Beispiel 3.8 *Beispiel von Runge³.* Betrachte $f : [-5, 5] \rightarrow \mathbb{R}$ mit $f(x) = 1/(1+x^2)$. Es gilt $f \in C^\infty([-5, 5])$. Bei der Nutzung äquidistanter Stützstellen erhält man die in Abbildung 3.1 zu sehenden Interpolationspolynome für $n \in \{2, 5, 10\}$. Man sieht, dass insbesondere die Interpolationspolynome höheren Grades zum Rand hin stark von der zu interpolierenden Funktion abweichen. Die Interpolationsfehler werden immer größer. \square

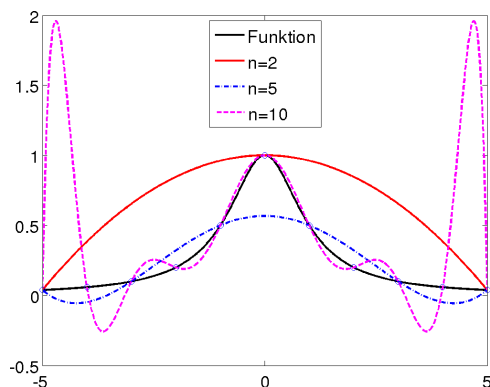


Abbildung 3.1: Beispiel 3.8, Interpolationspolynome für $n \in \{2, 5, 10\}$, äquidistante Stützstellen.

3.2.2 Hermite-Interpolation

Bemerkung 3.9 *Aufgabenstellung.* Die Interpolationsaufgabe (3.5) kann man dahingehend verallgemeinern, dass man nicht nur Funktionswerte vorgibt, sondern auch Werte von Ableitungen. Diese Aufgabe nennt man Hermite⁴-Interpolation. Seien Paare $(x_i, f^{(k)}(x_i))$ mit $i = 0, \dots, n$, $k = 0, \dots, m_i$, $m_i \in \mathbb{N}$, gegeben, wobei

²Isaac Newton (1642 – 1727)

³Carl David Tolmé Runge (1856 – 1927)

⁴Charles Hermite (1822 – 1901)

die Bedingung (3.1) an das Gitter erfüllt sei. Setze $N = \sum_{i=0}^n (m_i + 1)$. Die Aufgabe der Hermite-Interpolation lautet: Finde $p_N \in P_N$, so dass

$$p_N^{(k)}(x_i) = f^{(k)}(x_i), \quad i = 0, \dots, n, \quad k = 0, \dots, m_i. \quad (3.11)$$

In diesem Abschnitt wird im Wesentlichen ein Spezialfall der Hermite-Interpolation betrachtet. Für den allgemeinen Fall, sei auf die Literatur verwiesen. \square

Bemerkung 3.10 *Zusammenhang mit Taylor'scher Formel.* Seien $f \in C^{m+1}([a, b])$ und $x_0 \in (a, b)$. Dann besitzt die Taylorsche Formel mit dem Restglied von Lagrange die Gestalt

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

mit $\xi = x_0 + \theta(x - x_0)$, $\theta \in (0, 1)$. Das Taylor-Polynom

$$p_n = \sum_{k=0}^m \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k$$

löst offenbar die Hermite-Interpolationsaufgabe für $n = 0$ und $m_0 = m$. Die Gestalt des Fehlers

$$\frac{f^{(m+1)}(\xi)}{(m+1)!} (x - x_0)^{m+1}$$

erinnert an den Fehler (3.9) der Polynominterpolation ohne Ableitungen. \square

Bemerkung 3.11 *Spezielle Hermite-Interpolationsaufgabe.* In der Vorlesung wird sich auf folgenden Spezialfall beschränkt. Seien $f \in C^1([a, b])$ und seien die Bedingungen

$$f(x_i) = f_i \quad f'(x_i) = f'_i, \quad i = 0, \dots, n, \quad (3.12)$$

gegeben. Das sind $(2n + 2)$ Bedingungen. Gesucht ist also ein Polynom $p \in P_{2n+1}$, welches die Bedingungen (3.12) erfüllt. \square

Beispiel 3.12 *Zur speziellen Hermite-Interpolationsaufgabe.* Gesucht ist das Polynom $p_3(x)$ dritten Grades, welches durch die Bedingungen

$$p_3(-1) = 1, \quad p_3'(-1) = 2, \quad p_3(1) = 3, \quad p_3'(1) = 4,$$

bestimmt ist. Mit dem Ansatz

$$p_3(x) = a_0 + a_1x + a_2x^2 + a_3x^3 \quad \implies \quad p_3'(x) = a_1 + 2a_2x + 3a_3x^2$$

erhält man das Gleichungssystem

$$\begin{pmatrix} 1 & -1 & 1 & -1 \\ 0 & 1 & -2 & 3 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}.$$

Dieses System kann man durch geschicktes Addieren und Subtrahieren von Zeilen lösen. Man erhält

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} \frac{3}{2} \\ 0 \\ \frac{1}{2} \\ 1 \end{pmatrix}.$$

Das gesuchte Polynom besitzt also die Gestalt $p_3(x) = \frac{3}{2} + \frac{1}{2}x^2 + x^3$. \square

⁵Brook Taylor (1685 – 1731)

Satz 3.13 Existenz und Eindeutigkeit des Hermite–Interpolationspolynoms zur Aufgabe (3.12). Das Hermite–Interpolationspolynom $p \in P_{2n+1}$ zur Aufgabe (3.12) existiert und es ist eindeutig bestimmt.

Beweis: *Existenz.* Ein Hermite–Interpolationspolynom wird konstruiert. Dazu verwendet man den Ansatz

$$p(x) = \sum_{i=0}^n f_i \varphi_i(x) + \sum_{i=0}^n f'_i \psi_i(x) \quad (3.13)$$

mit $\varphi_i, \psi_i \in P_{2n+1}$, $i = 0, \dots, n$. Diese Polynome sollen folgende Bedingungen erfüllen

$$\begin{aligned} \varphi_i(x_j) &= \delta_{ij}, & \varphi'_i(x_j) &= 0, \\ \psi_i(x_j) &= 0, & \psi'_i(x_j) &= \delta_{ij}, \end{aligned} \quad i, j = 0, \dots, n. \quad (3.14)$$

Findet man solche Polynome, dann löst das Polynom (3.13) die Hermite–Interpolationsaufgabe (3.12).

Setze

$$\varphi_i(x) = (1 - 2L'_i(x_i)(x - x_i)) L_i^2(x), \quad (3.15)$$

$$\psi_i(x) = (x - x_i) L_i^2(x), \quad (3.16)$$

wobei $L_i(x)$ das in (3.7) definierte Lagrange–Polynom ist. Es gelten, wegen $L_i(x_j) = \delta_{ij}$,

$$\begin{aligned} \varphi_i(x_i) &= L_i^2(x_i) = 1, \\ \varphi_i(x_j) &= (1 - 2L'_i(x_j)(x_j - x_i)) L_i^2(x_j) = 0, \\ \psi_i(x_i) &= (x_i - x_i) L_i^2(x_i) = 0, \\ \psi_i(x_j) &= (x_j - x_i) L_i^2(x_j) = 0, \\ \varphi'_i(x_i) &= -2L'_i(x_i) L_i^2(x_i) + 2L'_i(x_i) L_i(x_i) = -2L'_i(x_i) + 2L'_i(x_i) = 0, \\ \varphi'_i(x_j) &= -2L'_i(x_i) L_i^2(x_j) + 2(1 - 2L'_i(x_i)(x_j - x_i)) L'_i(x_j) L_i(x_j) = 0, \\ \psi'_i(x_i) &= L_i^2(x_i) + 2(x_i - x_i) L_i(x_i) L'_i(x_i) = L_i^2(x_i) = 1, \\ \psi'_i(x_j) &= L_i^2(x_j) + 2(x_j - x_i) L_i(x_j) L'_i(x_j) = 0. \end{aligned}$$

Damit erfüllt p aus (3.13) mit (3.15), (3.16) die Hermite–Interpolationsaufgabe (3.12).

Eindeutigkeit. Angenommen, es gibt ein weiteres Polynom \tilde{p} , welches die Hermite–Interpolationsaufgabe (3.12) löst. Dann gilt $p - \tilde{p} \in P_{2n+1}$ und $p - \tilde{p}$ besitzt die $(n + 1)$ doppelten Nullstellen x_0, \dots, x_n . Nach dem Fundamentalsatz der Algebra muss dann aber $p - \tilde{p} \equiv 0$ gelten. ■

Satz 3.14 Interpolationsfehler in Aufgabe (3.12). Sei $f \in C^{2n+2}([a, b])$, sei (3.1) erfüllt und sei p das Hermite–Interpolationspolynom (3.13) mit (3.15), (3.16). Dann gibt es zu jedem $x \in [a, b]$ ein $\xi \in (a, b)$ mit

$$f(x) - p(x) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \omega_{n+1}^2(x) \quad (3.17)$$

mit $\omega_{n+1}(x)$ aus (3.8).

Beweis: Für $x = x_i$ sind der Fehler und die rechte Seite von (3.17) beide Null.

Sei $x \in (a, b)$, $x \neq x_i$ für alle i , beliebig aber fest. Betrachte die Funktion

$$h(z) = (f(x) - p(x)) \omega_{n+1}^2(z) - (f(z) - p(z)) \omega_{n+1}^2(x).$$

Dann besitzt $h(z)$ in $z = x_i$ eine doppelte Nullstelle, weil $\omega_{n+1}(x_i) = 0$ und dieser Faktor quadratisch vorkommt. Für $z = x$ besitzt $h(z)$ eine einfache Nullstelle. Betrachte $h'(z)$. Dann besitzt $h'(z)$ Nullstellen in x_i , $i = 0, \dots, n$. Nach dem Satz von Rolle⁶ besitzt

⁶Michel Rolle (1652 – 1719)

$h'(z)$ ($n + 1$) weitere Nullstellen, die nicht x_i oder x sind. Insgesamt besitzt $h'(z)$ also $2n + 2$ Nullstellen. Fortgesetzte Anwendung des Satzes von Rolle liefert, dass $h^{(2n+2)}$ eine Nullstelle $\xi \in (a, b)$ besitzt. Es ist p ein Polynom vom Grad $(2n + 1)$ und ω_{n+1} ein Polynom vom Grad n mit dem Faktor Eins vor der höchsten Potenz. Deshalb gilt

$$h^{(2n+2)}(z) = (2n + 2)!(f(x) - p(x)) - f^{(2n+2)}(z)\omega_{n+1}^2(x).$$

Nun folgt für $z = \xi$ die Behauptung (3.17). ■

3.2.3 Zusammenhang zwischen Interpolations- und Bestapproximationsfehler

Bemerkung 3.15 Ziel. Die Bestapproximation und die Interpolation durch gegebene Stützstellen sind zwei Möglichkeiten, eine Funktion durch eine einfachere Funktion zu approximieren. In diesem Abschnitt werden diese beiden Möglichkeiten miteinander verglichen. □

Satz 3.16 *Kondition der Polynominterpolation. Der Interpolationsoperator*

$$\phi : C([a, b]) \rightarrow P_n, \quad f \mapsto \phi(f) = p_n,$$

ist eine Projektion. Das heißt, ϕ ist linear und es gilt $\phi^2 = \phi$. Für die Kondition der Polynominterpolation gilt

$$\|\phi\|_\infty = \sup_{f \in C([a, b])} \frac{\|\phi(f)\|_\infty}{\|f\|_\infty} = \Lambda_n$$

mit der Lebesgue-Konstanten

$$\Lambda_n = \left\| \sum_{k=0}^n |L_k| \right\|_\infty = \max_{x \in [a, b]} \sum_{k=0}^n |L_k| \quad (3.18)$$

mit $L_k(x)$ definiert in (3.7).

Beweis: Dieses Resultat findet man im CoMa-Skript. ■

Satz 3.17 Zusammenhang zwischen Interpolations- und Bestapproximationsfehler. Sei $f \in C([a, b])$ and sei $\hat{p}_n \in P_n$ die Lösung der Tschebyscheffschen Bestapproximations-Aufgabe (1.14) mit dem Approximationsfehler

$$E_n(f) = \|f - \hat{p}_n\|_\infty.$$

Sei $p_n \in P_n$ die Lösung der Interpolationsaufgabe (3.5). Dann gilt

$$\|f - p_n\|_\infty \leq (1 + \Lambda_n)E_n(f) \quad \forall f \in C([a, b]) \quad (3.19)$$

mit Λ_n definiert in (3.18).

Beweis: Der Beweis erfolgt durch Anwendung der Dreiecksungleichung, Nutzung der Eigenschaften von ϕ und Nutzung der Definition der Lebesgue-Konstanten

$$\begin{aligned} \|f - p_n\|_\infty &\leq \|f - \hat{p}_n\|_\infty + \|\hat{p}_n - p_n\|_\infty = \|f - \hat{p}_n\|_\infty + \|\hat{p}_n - \phi(f)\|_\infty \\ &= \|f - \hat{p}_n\|_\infty + \|\phi(\hat{p}_n - f)\|_\infty \\ &\leq \|f - \hat{p}_n\|_\infty + \|\phi\|_\infty \|\hat{p}_n - f\|_\infty \\ &= (1 + \Lambda_n) \|f - \hat{p}_n\|_\infty. \end{aligned}$$

■

Bemerkung 3.18 *Interpretation von Satz 3.17.* Abschätzung (3.19) besagt, dass der Interpolationsfehler um den Faktor $(1 + \Lambda_n)$ schlechter ist, als der Bestapproximationsfehler. Um genaue Interpolanten zu erhalten, ist ein kleines Λ_n nötig. \square

Bemerkung 3.19 *Zum Bestapproximationsfehler.* Aus der Analysis-Grundvorlesung ist bekannt, dass sich jede stetige Funktion beliebig genau durch Polynome approximieren lässt, Satz von Weierstraß. Das bedeutet, es gilt

$$\lim_{n \rightarrow \infty} E_n(f) = 0. \quad (3.20)$$

\square

Bemerkung 3.20 *Zur Lebesgue-Konstanten.* Die Lebesgue-Konstante verhält sich leider nicht besonders erfreulich. So bewies Faber⁷ im Jahre 1914:

- Zu jeder Folge von Stützstellen gibt es eine Funktion $f \in C([a, b])$, so dass gilt

$$\liminf_{n \rightarrow \infty} \|f - p_n\|_\infty > 0.$$

Der Interpolationsfehler geht also nicht gegen Null. Gemeinsam mit (3.20) bedeutet dies, dass $\Lambda_n \rightarrow \infty$ für $n \rightarrow \infty$.

Im Jahre 1978 zeigten Brutman, de Boor und Pinkus eine untere Schranke für die Lebesgue-Konstante.

- Für jede Folge von Stützstellen gilt

$$\frac{2}{\pi} \ln(n+1) + \frac{2}{\pi} \left(\gamma + \ln \left(\frac{4}{\pi} \right) \right) \leq \Lambda_n.$$

Hierbei ist

$$\gamma = \lim_{n \rightarrow \infty} \left(\sum_{i=1}^n \frac{1}{i} - \ln(n) \right) \approx 0.57721$$

die Euler⁸-Mascheroni⁹-Konstante.

Das heißt, selbst für die bestmögliche Wahl von Stützstellen wächst Λ_n mindestens logarithmisch an.

Das Wachstumsverhalten kann aber noch viel schlimmer sein. So bewies Turetskii¹⁰ (1940)

- Für die Folge äquidistanter Stützstellen $x_j = a + jh$, $j = 0, \dots, n$, mit der Schrittweite $h = (b-a)/n$ gilt

$$\Lambda_n \approx e^{-1} \frac{2^{n+1}}{n \ln(n)}.$$

Die Lebesgue-Konstante wächst exponentiell! Das bedeutet, äquidistante Stützstellen sollten vermieden werden. \square

Bemerkung 3.21 *Geschickte Wahl der Stützstellen.* Für eine geschickte Wahl der Stützstellen kann man sich an der Fehlerdarstellung (3.10) orientieren. Vernachlässigt man den Einfluss von $f(x)$, dann erhält man einen kleinen Fehler, wenn

$$\|\omega_{n+1}\|_\infty = \left\| \prod_{j=0}^n (x - x_j) \right\|_\infty \quad (3.21)$$

⁷Georg Faber (1877 – 1966)

⁸Leonhard Euler (1707 – 1783)

⁹Lorenzo Mascheroni (1750 – 1800)

¹⁰A.H. Turetskii

möglichst klein ist. Das Polynom $\omega_{n+1}(x)$ ist vom Grad n und der Koeffizient vor dem Term höchsten Grades ist gleich Eins. Die Koeffizienten vor den anderen Termen kann man noch wählen. Damit lautet eine äquivalente Formulierung der die Aufgabe (3.21) zu minimieren, ein Polynom n -ten Grades zu finden, welches x^{n+1} in der Norm $\|\cdot\|_\infty$ am besten approximiert.

Die Lösung dieser Aufgabe im Fall $[a, b] = [-1, 1]$ ist aus Beispiel 1.37 bekannt. Man erhält die Tschebyscheff–Polynome 1. Art

$$\omega_{n+1}(x) = \frac{1}{2^n} T_{n+1}(x).$$

Die resultierenden Stützstellen sind gerade die Nullstellen von $T_{n+1}(x)$

$$x_j = \cos\left(\frac{2(n-j)+1}{2(n+1)}\pi\right).$$

Im Fall $[a, b] \neq [-1, 1]$ kann man die Tschebyscheff–Stützstellen durch Transformation bestimmen.

Im Jahre 1974 zeigte Rivlin¹¹

- Für die Tschebyscheff–Stützstellen gilt

$$\frac{2}{\pi} \ln(n+1) + \frac{2}{\pi} \left(\gamma + \ln\left(\frac{8}{\pi}\right) \right) \leq \Lambda_n \leq \frac{2}{\pi} \ln(n+1) + 1.$$

In diesem Fall hat man wirklich nur das logarithmische Wachstum der Lebesgue–Konstanten. Der konstante Term auf der linken Seite hat ungefähr den Wert 0.9625. \square

Beispiel 3.22 *Beispiel von Runge.* Es wird die gleiche Funktion wie in Beispiel 3.8 betrachtet. Nimmt man als Stützstellen die Tschebyscheff–Stützstellen, so erhält man die in Abbildung 3.2 dargestellten Interpolationen. Man erhält selbst für $n = 100$ eine gute Interpolation. \square

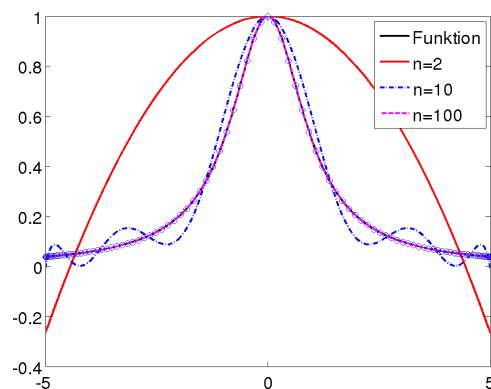


Abbildung 3.2: Beispiel 3.22, Interpolationspolynome für $n \in \{2, 10, 100\}$, Tschebyscheff–Stützstellen.

3.3 Spline–Interpolation

Bemerkung 3.23 *Motivation, Aufgabenstellung.* Die Polynominterpolation besitzt zwei sehr negative Eigenschaften:

¹¹Theodore J. Rivlin

- für große n , das heißt viele Stützstellen, kann sie instabil werden, insbesondere bei äquidistanten Stützstellen,
- für nichtglatte Funktionen kann der Interpolationsfehler zwischen den Stützstellen beliebig groß werden.

Diese beiden Situationen (viele äquidistante Stützstellen oder nichtglatte Funktionen) treten in den Anwendung jedoch häufig auf. In diesen Fällen muss man eine andere Interpolation als die Polynom-Interpolation verwenden. Eine oft genutzte Herangehensweise ist die Spline-Interpolation:

spline (engl.) – längliches, dünnes Stück Holz oder Metall.

□

Definition 3.24 Spline. Seien $x_0, \dots, x_n \in [a, b]$ paarweise verschiedene Punkte mit $a = x_0 < x_1 < \dots < x_n = b$. Die Funktion $s_k(x) : [a, b] \rightarrow \mathbb{R}$ wird Spline vom Grad k bezüglich der Stützstellen $\{x_j\}$ genannt, falls

$$\begin{aligned} s_k(x)|_{[x_j, x_{j+1}]} &\in P_k, \quad j = 0, \dots, n-1, \\ s_k(x) &\in C^{k-1}([a, b]). \end{aligned} \quad (3.22)$$

□

Bemerkung 3.25 Anzahl der gegebenen Bedingungen. Definition 3.24 bedeutet: Ein Spline vom Grad k ist

- eine stückweise polynomiale Funktion, in jedem Teilintervall ein Polynom vom Grad k ,
- im Gesamtintervall noch $(k-1)$ -mal differenzierbar.

In jedem Teilintervall kann man den Spline als

$$s_{k,j}(x) = \sum_{i=0}^k s_{ij}(x-x_j)^i, \quad x \in [x_j, x_{j+1}], \quad j = 0, \dots, n-1,$$

darstellen. Man hat also $n(k+1)$ unbekannte Koeffizienten s_{ij} . Aus (3.22) folgt

$$s_{k,j-1}^{(m)}(x_j) = s_{k,j}^{(m)}(x_j), \quad j = 1, \dots, n-1, \quad m = 0, \dots, k-1.$$

Das sind $k(n-1)$ Bedingungen an die Koeffizienten. Im Allgemeinen soll der Spline eine Funktion $f(x)$ approximieren, deren Werte in den Stützstellen berechnet werden können. Damit hat man weitere $(n+1)$ Bedingungen: $s_k(x_j) = f(x_j)$, $j = 0, \dots, n$. Es fehlen noch $(k-1)$ Bedingungen. □

Beispiel 3.26 Linearer Spline. Ein linearer Spline, das heißt $k=1$, ist nichts weiter als ein Polygonzug. Ein solcher Polygonzug gehört zum Raum S_n , siehe Definition 1.8. Für lineare Splines fehlen keine Bedingungen.

Die Voraussetzungen der Interpolationsaufgabe seien wie in Bemerkung 3.1. Dann lautet die Interpolationsaufgabe für lineare Splines: Finde $u_n \in S_n$, so dass

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

Die Lösung dieser Aufgabe ist

$$u_n(x) = \sum_{i=0}^n f(x_i) \varphi_i(x),$$

wobei $\{\varphi_i(x)\}_{i=0}^n$ die in (1.16) definierte Knotenbasis ist. □

Satz 3.27 Fehlerabschätzung für lineare Spline-Interpolation. Seien $a = x_0$, $b = x_n$ und $f \in C^2([a, b])$. Dann gilt für die lineare Spline-Interpolierte $u_n(x)$ die Fehlerabschätzung

$$\|f - u_n\|_\infty \leq h^2 \frac{\|f''\|_\infty}{8}$$

mit

$$h = \max_{j=1, \dots, n} h_j, \quad h_j = x_j - x_{j-1}.$$

Beweis: Man wendet die Fehlerabschätzung der Polynominterpolation (3.9) auf jedes Teilintervall an, da man dort mit einem linearen Polynom interpoliert. Damit ergibt sich für $j = 1, \dots, n$, $\xi \in [x_{j-1}, x_j]$,

$$\begin{aligned} \max_{x \in [x_{j-1}, x_j]} |f(x) - u_n(x)| &= \max_{x \in [x_{j-1}, x_j]} \left| \frac{f''(\xi)}{2!} \omega_2(x) \right| \\ &= \max_{x \in [x_{j-1}, x_j]} \left| \frac{f''(x)}{2} (x - x_{j-1})(x - x_j) \right| \\ &\leq \max_{x \in [x_{j-1}, x_j]} \left| \frac{f''(x)}{2} \right| \max_{x \in [x_{j-1}, x_j]} |(x - x_{j-1})(x - x_j)| \\ &= \frac{h_j^2}{8} \max_{x \in [x_{j-1}, x_j]} |f''(x)|, \end{aligned}$$

da der größte Wert des Produktes für den Mittelpunkt $x = (x_{j-1} + x_j)/2$ angenommen wird. Damit folgt

$$\|f - u_n\|_\infty = \max_{j=1, \dots, n} \left(\max_{x \in [x_{j-1}, x_j]} |f(x) - u_n(x)| \right) \leq \frac{h^2}{8} \max_{x \in [a, b]} |f''(x)|. \quad \blacksquare$$

Bemerkung 3.28 Kubische Splines. Lineare Splines sind einfach zu handhaben, jedoch relativ ungenau und die Interpolation besitzt Ecken, siehe unten in Abbildung 3.3.

In der Praxis möchte man glattere Übergänge und dafür nutzt man oft kubische Splines, das heißt $k = 3$. Ein kubischer Spline ist zweimal stetig bis zum Rand differenzierbar, insbesondere ist seine Krümmung wohldefiniert. Für die zwei fehlenden Bedingungen setzt man zum Beispiel die Randwerte

$$s_3''(a) = s_3''(b) = 0. \quad (3.23)$$

Diese Randbedingung nennt man natürliche Randbedingung. Die Randbedingung

$$s'(a) = f'(a), \quad s'(b) = f'(b) \quad (3.24)$$

heißt vollständige Randbedingung.

Betrachten wird nun eine Funktion $f(x)$, die durch einen kubischen Spline interpoliert werden soll. Wir verwenden die Bezeichnungen

$$f_i = s_3(x_i), \quad m_i = s_3'(x_i), \quad M_i = s_3''(x_i), \quad i = 0, \dots, n.$$

Die zweite Ableitung $s_3''(x)$ ist eine stetige, stückweise lineare Funktion (Polygonzug). Mit den obigen Bezeichnungen gilt

$$s_3''(x) = M_{i-1} \frac{x_i - x}{x_i - x_{i-1}} + M_i \frac{x - x_{i-1}}{x_i - x_{i-1}}, \quad x \in [x_{i-1}, x_i].$$

Die Stetigkeit von $s_3''(x)$ überprüft man durch Einsetzen der Intervallgrenzen. Zweimaliges integrieren liefert eine Darstellung des Splines in $[x_{i-1}, x_i]$

$$s_3(x) = \frac{M_{i-1}}{6} \frac{(x_i - x)^3}{x_i - x_{i-1}} + \frac{M_i}{6} \frac{(x - x_{i-1})^3}{x_i - x_{i-1}} + c_{i-1}(x - x_{i-1}) + d_{i-1}. \quad (3.25)$$

Einsetzen der Endpunkte des Teilintervalls ergibt die Konstanten:

$$\begin{aligned} f_{i-1} &= s_3(x_{i-1}) = \frac{M_{i-1}}{6}(x_i - x_{i-1})^2 + d_{i-1}, \quad \implies \\ d_{i-1} &= f_{i-1} - \frac{M_{i-1}}{6}(x_i - x_{i-1})^2, \end{aligned} \quad (3.26)$$

und

$$\begin{aligned} f_i &= s_3(x_i) = \frac{M_i}{6}(x_i - x_{i-1})^2 + c_{i-1}(x_i - x_{i-1}) + d_{i-1} \implies \\ c_{i-1} &= \frac{f_i - f_{i-1}}{x_i - x_{i-1}} - \frac{x_i - x_{i-1}}{6}(M_i - M_{i-1}). \end{aligned} \quad (3.27)$$

Man rechnet schnell nach, dass mit diesen Konstanten die Stetigkeit von $s_3(x)$ in den Stützstellen gewährleistet ist.

Aus der Stetigkeitsbedingung für die erste Ableitung erhält man weitere Gleichungen für M_0, \dots, M_n . Es gilt für $x \in [x_{i-1}, x_i]$

$$s'_3(x) = -\frac{M_{i-1}}{2} \frac{(x_i - x)^2}{x_i - x_{i-1}} + \frac{M_i}{2} \frac{(x - x_{i-1})^2}{x_i - x_{i-1}} + c_{i-1}.$$

Für den rechten Randpunkt erhält man

$$\begin{aligned} s'_3(x_{i+0}) &= \frac{M_i}{2}(x_i - x_{i-1}) + \frac{f_i - f_{i-1}}{x_i - x_{i-1}} - \frac{x_i - x_{i-1}}{6}(M_i - M_{i-1}) \\ &= \left(\frac{M_{i-1}}{6} + \frac{M_i}{3} \right) (x_i - x_{i-1}) + \frac{f_i - f_{i-1}}{x_i - x_{i-1}}. \end{aligned}$$

Betrachtet man nun den linken Endpunkt des Intervalls $[x_i, x_{i+1}]$, so erhält man auf die gleiche Weise

$$\begin{aligned} s'_3(x_{i+0}) &= -\frac{M_i}{2}(x_{i+1} - x_i) + \frac{f_{i+1} - f_i}{x_{i+1} - x_i} - \frac{x_{i+1} - x_i}{6}(M_{i+1} - M_i) \\ &= \left(-\frac{M_i}{3} - \frac{M_{i+1}}{6} \right) (x_{i+1} - x_i) + \frac{f_{i+1} - f_i}{x_{i+1} - x_i}. \end{aligned}$$

Setzt man diese Bedingungen gleich, so erhält man $(n-1)$ Gleichungen für M_0, \dots, M_n . Für die zwei fehlenden Gleichungen setzt man

$$2M_0 + \lambda_0 M_1 = g_0, \quad \mu_n M_{n-1} + 2M_n = g_n$$

mit vorgegebenen Koeffizienten $\lambda_0, \mu_n \in [0, 1]$ und g_0, g_n . Zur Erfüllung von (3.23) setzt man alle diese Koeffizienten gleich Null.

Zur Berechnung der Koeffizienten M_0, \dots, M_n hat man damit folgendes lineares Gleichungssystem zu lösen

$$\begin{pmatrix} 2 & \lambda_0 & & & & & \\ \mu_1 & 2 & \lambda_1 & & & & \\ & \mu_2 & 2 & \lambda_2 & & & \\ & & & \ddots & & & \\ & & & & \mu_{n-1} & 2 & \lambda_{n-1} \\ & & & & & \mu_n & 2 \end{pmatrix} \begin{pmatrix} M_0 \\ M_1 \\ M_2 \\ \vdots \\ M_{n-1} \\ M_n \end{pmatrix} = \begin{pmatrix} g_0 \\ g_1 \\ g_2 \\ \vdots \\ g_{n-1} \\ g_n \end{pmatrix}.$$

Die Koeffizienten sind

$$\begin{aligned} \mu_i &= \frac{x_i - x_{i-1}}{x_{i+1} - x_{i-1}}, \\ \lambda_i &= \frac{x_{i+1} - x_i}{x_{i+1} - x_{i-1}}, \\ g_i &= \frac{6}{x_{i+1} - x_{i-1}} \left(\frac{f_{i+1} - f_i}{x_{i+1} - x_i} - \frac{f_i - f_{i-1}}{x_i - x_{i-1}} \right), \quad i = 1, \dots, n-1. \end{aligned}$$

Nach der Lösung dieses Systems erhält man mit (3.26), (3.27) durch Einsetzen in (3.25) die Darstellung des kubischen Splines. \square

Beispiel 3.29 *Beispiel von Runge.* Dieses Beispiel wurde in Beispiel 3.8 zur Illustration der Nachteile der Polynominterpolation verwendet. Betrachte wieder $f : [-5, 5] \rightarrow \mathbb{R}$ mit $f(x) = 1/(1+x^2)$. Man erhält mit Spline-Interpolation und $n \in \{2, 5, 10\}$ die in Abbildung 3.3 dargestellten Ergebnisse. Für den kubischen Spline wurden als zusätzliche Randbedingungen $s''(-5) = s''(5) = 0$ verwendet. \square

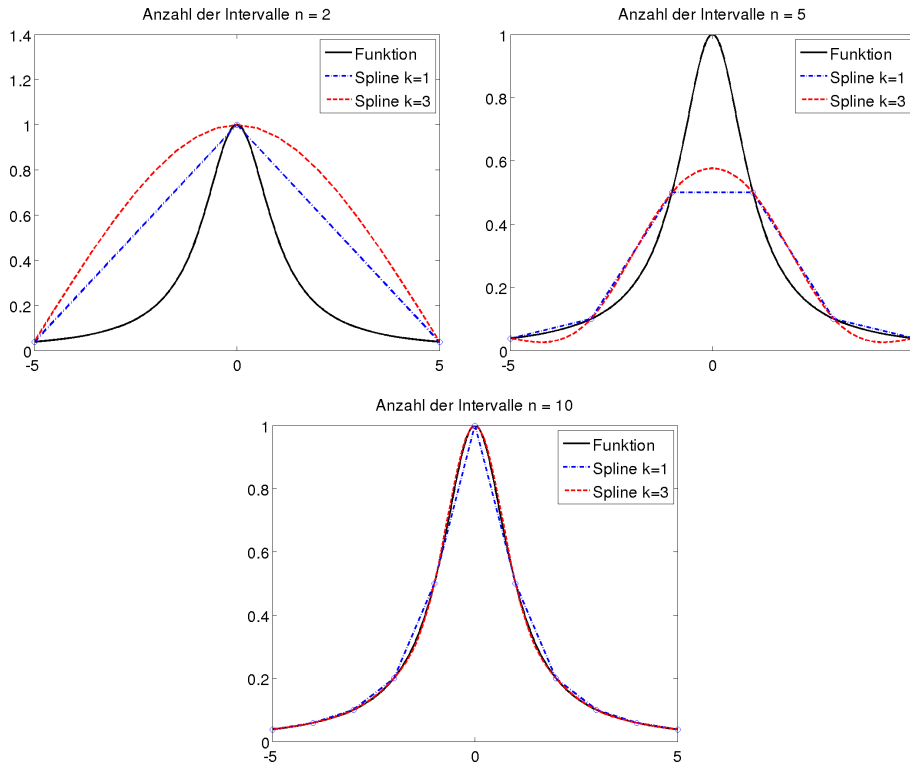


Abbildung 3.3: Beispiel 3.29. Interpolation mit Splines.

Satz 3.30 *Interpolationsfehlerabschätzung.* Seien $f \in C^4([a, b])$,

$$h_{\max} = \max_{i=0, \dots, n-1} (x_{i+1} - x_i), \quad h_{\min} = \min_{i=0, \dots, n-1} (x_{i+1} - x_i), \quad \beta = \frac{h_{\max}}{h_{\min}} \geq 1.$$

Dann gilt für die kubische Spline-Interpolation mit vollständiger Randbedingung (3.24)

$$\max_{x \in [a, b]} |f^{(r)}(x) - s_3^{(r)}(x)| \leq C_r h_{\max}^{4-r} \max_{x \in [a, b]} |f^{(4)}(x)|$$

mit $C_0 = 5/384$, $C_1 = 1/24$, $C_2 = 3/8$ und $C_3 = \frac{1}{2}(\beta + 1/\beta)$.

Beweis: Siehe Literatur. \blacksquare

Bemerkung 3.31 *Interpretation der Aussage des Satzes.* Die Aussage von Satz 3.30 bedeutet, dass für $h_{\max} \rightarrow 0$ der kubische Spline samt seinen ersten beiden Ableitungen punktweise gegen $f(x)$ beziehungsweise gegen die entsprechenden Ableitungen von $f(x)$ konvergiert. Innerhalb der Intervalle konvergiert auch noch die dritte Ableitung des kubischen Splines, an den Stützstellen konvergiert der Mittelwert der beiden einseitigen dritten Ableitungen. \square