Vorlesung: Numerik 1 für Ingenieure

Version 19.11.17

Michael Karow

8. Vorlesung

Themen: Hornerschema, Polynominterpolation, Splines

Das Horner-Schema I

Gegeben sei das Polynom $p(x) = a_4\,x^4 + a_3\,x^3 + a_2\,x^2 + a_1\,x + a_0, \qquad a_k, x \in \mathbb{C}.$

Aufgabe: Berechne $p(x_0)$ für eine vorgegebene Stelle x_0.

Naive Lösung: Man berechnet erst die Potenzen von x_0 und multiplizert diese anschließend mit den Koeffizienten a_k. Die Ergebnisse werden addiert. Dabei hat man 3+4=7 Multiplikationen auzuführen.

Bessere Lösung: Man schreibt das Polynom in der Form

$$p(x_0) = (((a_4\,x_0 + a_3)\,x_0 + a_2)\,x_0 + a_1)\,x_0 + a_0$$

und rechnet von innen nach außen: erst bestimmt man $a_4\,x_0 + a_3$, dies multipliziert man x_0 und addiert a_2 usw.. Bei dieser Rechenmethode braucht man nur 4 Multiplikationen. Algorithmus: Setze

$$\begin{aligned} q_3 &= a_4 \\ q_2 &= q_3\,x_0 + a_3 \\ q_1 &= q_2\,x_0 + a_2 \\ q_0 &= q_1\,x_0 + a_1 \\ p_0 &= q_0\,x_0 + a_0 \end{aligned}$$

Dann ist $p_0 = p(x_0)$. Bei einer schriftlichen Rechnung schreibt man die Rechenergebnisse in eine Tabelle **(Horner-Schema)**:

a_4	a_3	a_2	a_1	a_0
0	$+q_3\,x_0$	$+q_2\,x_0$	$+q_1\,x_0$	$+q_0\,x_0$
$q_3 = a_4$	q_2	q_1	q_0	$p_0 = p(x_0)$

Das Horner-Schema II

Die Zwischenergebnisse $q_3, \ldots, q_0$ im Horner-Schema haben eine eigenständige Bedeutung. Sie sind nämlich die Koeffizienten des Polynoms, das man bei Polynomdivision von $p(x)$ mit dem Linearfaktor $x - x_0$ erhält. Genauer gilt:

$$p(x) = (q_3\,x^3 + q_2\,x^2 + q_1\,x + q_0)\,(x - x_0) + p(x_0). \qquad (*)$$

Beweis:

$$\begin{aligned}
p(x) &= a_4\,x^4 + a_3\,x^3 + a_2\,x^2 + a_1\,x + a_0 \\
&= a_4\,x^3\,(x - x_0) + (a_4x_0 + a_3)\,x^3 + a_2\,x^2 + a_1\,x + a_0 \\
&= a_4\,x^3\,(x - x_0) + (a_4x_0 + a_3)\,x^2(x - x_0) + ((a_4x_0 + a_3)\,x_0 + a_2)x^2 + a_1\,x + a_0 \\
&= a_4\,x^3\,(x - x_0) + (a_4x_0 + a_3)\,x^2(x - x_0) + ((a_4x_0 + a_3)\,x_0 + a_2)x\,(x - x_0) \\
&\quad (((a_4x_0 + a_3)\,x_0 + a_2)x_0 + a_1)\,x + a_0 \\
&= a_4\,x^3\,(x - x_0) + (a_4x_0 + a_3)\,x^2(x - x_0) + ((a_4x_0 + a_3)\,x_0 + a_2)x\,(x - x_0) \\
&\quad (((a_4x_0 + a_3)\,x_0 + a_2)x_0 + a_1)\,(x - x_0) + (((a_4x_0 + a_3)\,x_0 + a_2)x_0 + a_1)x_0 + a_0 \\
&= (q_3\,x^3 + q_2\,x^2 + q_1\,x + q_0)\,(x - x_0) + p(x_0).
\end{aligned}$$

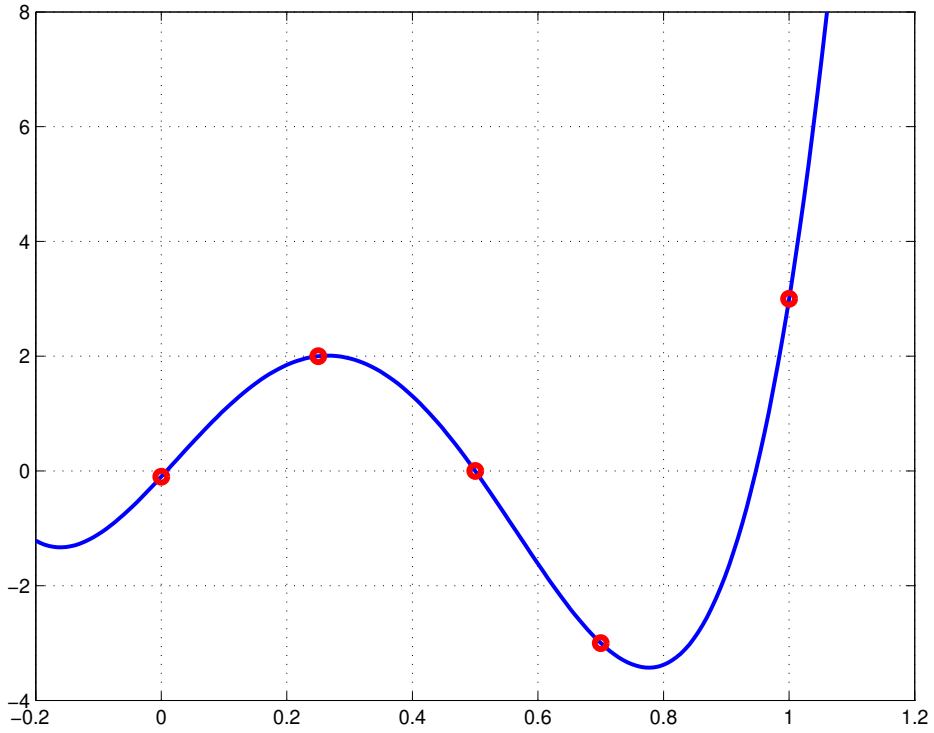
Alle hier getroffenen Aussagen gelten analog für Polynome p beliebigen Grades ≥ 1.

Das Lagrange-Interpolationsproblem.

Gegeben seien die $n+1$ Wertepaare (Stützpunkte) $(x_j, f_j) \in \mathbb{C}, \quad j = 1, \ldots, n+1$, wobei die Stützstellen x_j alle von einander verschieden sind.

Gesucht ist ein Polynom vom Grad $\leq n$, so dass

$$p(x_j) = f_j \qquad j = 1, \ldots, n+1.$$



Das Lagrange-Interpolationsproblem ist stets eindeutig lösbar.

Siehe die folgenden Seiten.

Zur Lösung des Polynominterpolationsproblems gibt es (mindestens) 3 Methoden:

1. Vandermonde-Methode (das ist keine offizielle Bezeichnung):
   Lösen eines linearen Gleichungssystems mit einer Vandermonde-Matrix.

2. Lagrange-Methode:
   Darstellung des Interpolationspolynoms mit Lagrange-Basispolynomen.
   Folgerung: baryzentrische Formel.

3. Newton-Methode:
   Darstellung des Interpolationspolynoms mit dividierten Differenzen.

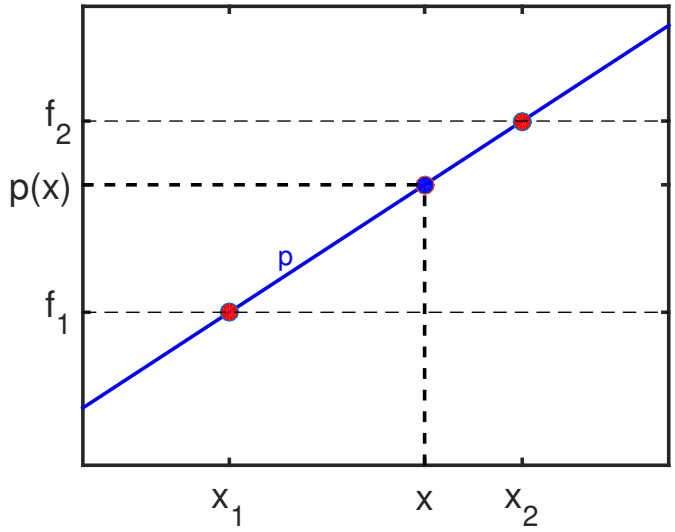
Polynominterpolation ist u.a. die Grundlage für

1. numerische Differentiation (finite Differenzenformeln),

2. numerische Integration,

3. numerische Lösung von Differentialgleichungen.

Einführendes Beispiel: Interpolation durch eine Gerade



Die Gerade p durch (x_1, f_1) und (x_2, f_2) in verschiedenen Darstellungen:

$$p(x) = f_1 + \frac{f_2 - f_1}{x_2 - x_1}(x - x_1) \qquad \textbf{(Newton-Methode)}$$

$$= f_1 \frac{x - x_2}{x_1 - x_2} + f_2 \frac{x - x_1}{x_2 - x_1} \qquad \textbf{(Lagrange-Methode)}$$

$$= a_1 x + a_0, \qquad \textbf{(Vandermonde-Methode)}$$

wobei $a_1 x_1 + a_0 = f_1, \quad a_1 x_2 + a_0 = f_2$

(lineares Gleichungssystem für a_0, a_1)

Die Vandermonde-Methode

Sei $p(x) = a_n\, x^n + a_{n-1}\, x^{n-1} + \ldots + a_2\, x^2 + a_1\, x + a_0$.

Zu bestimmen sind die Koeffizienten a_k, so dass $p(x_j) = f_j$, $j = 1, \ldots, n+1$.

Dies ist ein lineares Gleichungssystem

$$\begin{aligned}
a_0 + a_1\, x_1 + a_2\, x_1^2 + \ldots + a_{n-1}\, x_1^{n-1} + a_n\, x_1^n &= f_1 \\
a_0 + a_1\, x_2 + a_2\, x_2^2 + \ldots + a_{n-1}\, x_2^{n-1} + a_n\, x_2^n &= f_2 \\
&\vdots \quad \vdots \quad \vdots \\
a_0 + a_1\, x_{n+1} + a_2\, x_{n+1}^2 + \ldots + a_{n-1}\, x_{n+1}^{n-1} + a_n\, x_{n+1}^n &= f_{n+1}.
\end{aligned}$$

In Matrix-Vektor-Schreibweise:

$$\underbrace{\begin{bmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^n \\ 1 & x_2 & x_2^2 & \ldots & x_2^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n+1} & x_{n+1}^2 & \ldots & x_{n+1}^n \end{bmatrix}}_{V} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_{n+1} \end{bmatrix}$$

Die Matrix V heißt **Vandermonde-Matrix**.

Für größere n ist diese Matrix schlecht konditioniert.

Für $n = 20$ und $x_j = j/n$ hat man laut MATLAB $cond_2(V) \approx 10^{16}$.

Die Vandermonde-Methode II

MATLAB berechnet die Koeffizienten des Interpolationspolynoms mit der Vandermonde-Methode. Die Anweisung dazu ist

```
a=polyfit(x,f,n)
```

Dabei ist `n` der Grad des Polynoms. Wenn `n` kleiner als die Anzahl der Stützpunkte-1 ist, dann hat das Interpolationsproblem im allgemeinen keine Lösung. MATLAB findet dann ein Polynom, dass die Stützpunkte im Sinne der kleinsten Fehlerquadrate am besten approximiert (Ausgleichspolynom).

Warnung. Die Koeffizienten eines Polynoms sind bei MATLAB andersherum nummeriert als auf diesen Folien: Der Koeffizient vor x^n is bei MATLAB $a(1)$ usw.. Dementsprechend erzeugt MATLAB mit dem Kommando `vander(x)` auch nicht die Vandermonde-Matrix von der vorigen Folie, sondern

$$\begin{bmatrix} x_1^n & \dots & x_1^2 & x_1 & 1 \\ x_2^n & \dots & x_2^2 & x_2 & 1 \\ \vdots & & \vdots & \vdots & \vdots \\ x_{n+1}^n & \dots & x_{n+1}^2 & x_{n+1} & 1 \end{bmatrix}$$

Zur Auswertung eines Polynoms an den Stellen x_j kann man bei MATLAB die Funktion `polyval` benutzen.

Die Lagrange-Methode

Man kann das Interpolationspolynom auch direkt hinschreiben, ohne etwas zu rechnen.

Beispiel: Das Polynom p, welches die Bedingungen

$$p(x_1) = f_1, \quad p(x_2) = f_2, \quad p(x_3) = f_3$$

erfüllt, ist

$$p(x) = f_1 \underbrace{\frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)}}_{L_1(x)} + f_2 \underbrace{\frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)}}_{L_2(x)} + f_3 \underbrace{\frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}}_{L_3(x)}.$$

Die Polynome L_k heißen **Lagrange'sche Basispolynome**. Ihre allgemeine Form für $n+1$ Stützstellen x_k ist

$$L_k(x) = \frac{(x-x_1)\dots(x-x_{k-1})(x-x_{k+1})\dots(x-x_{n+1})}{(x_k-x_1)\dots(x_k-x_{k-1})(x_k-x_{k+1})\dots(x_k-x_{n+1})}.$$

Es gilt dann

$$L_k(x_j) = \begin{cases} 1 & \text{für } j = k \\ 0 & \text{sonst.} \end{cases}$$

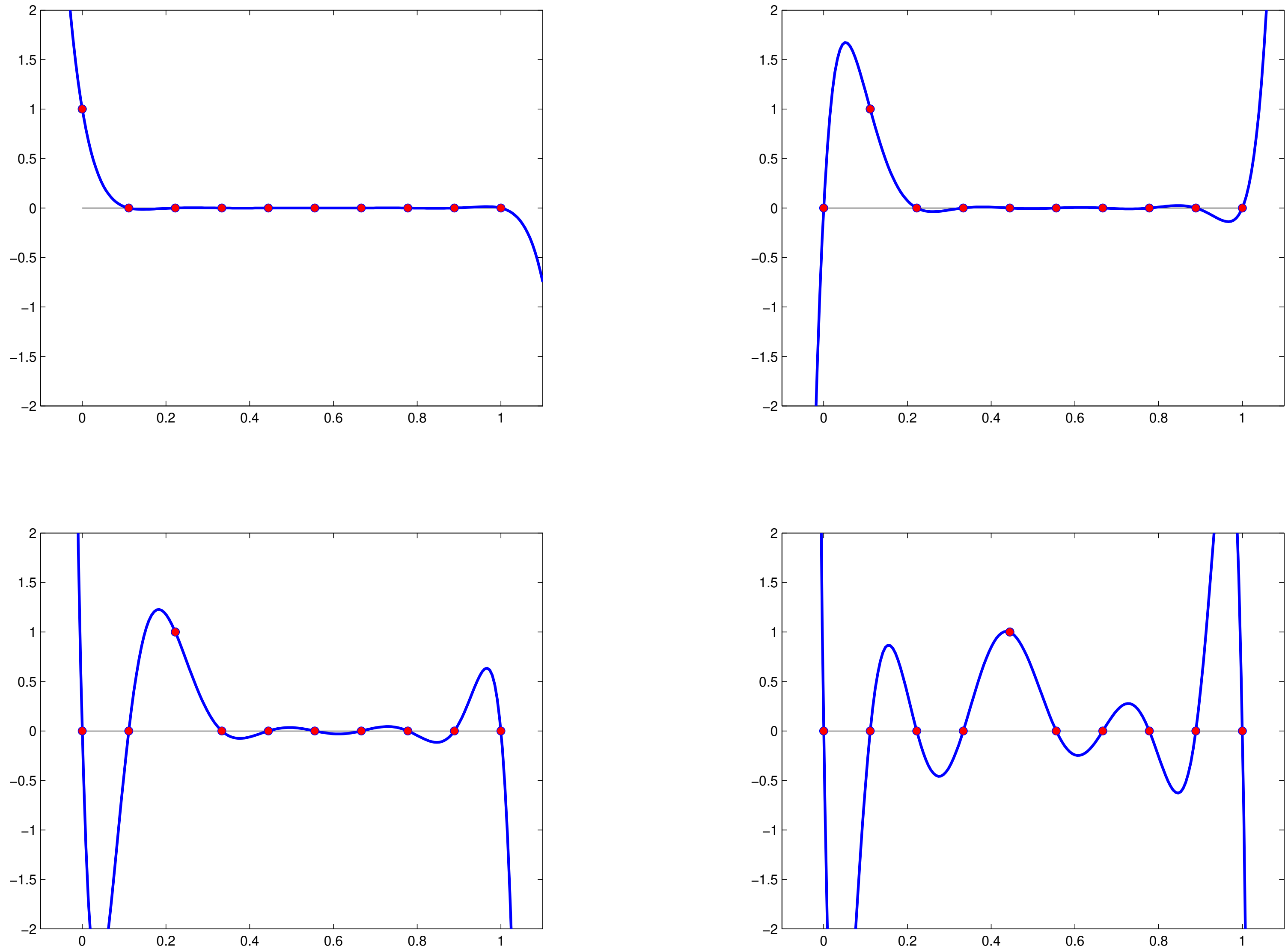
Das Interpolationspolynom zu den Daten (x_j, f_j) ist $\quad p(x) = \sum_{k=1}^{n+1} f_k \, L_k(x).$

Nachteil der Lagrange-Basis:

Zu großer Rechenaufwand und zu großer numerischer Fehler beim Auswerten.
Nimmt man einen Interpolationspunkt hinzu, so muss man alle Polynome neu berechnen.

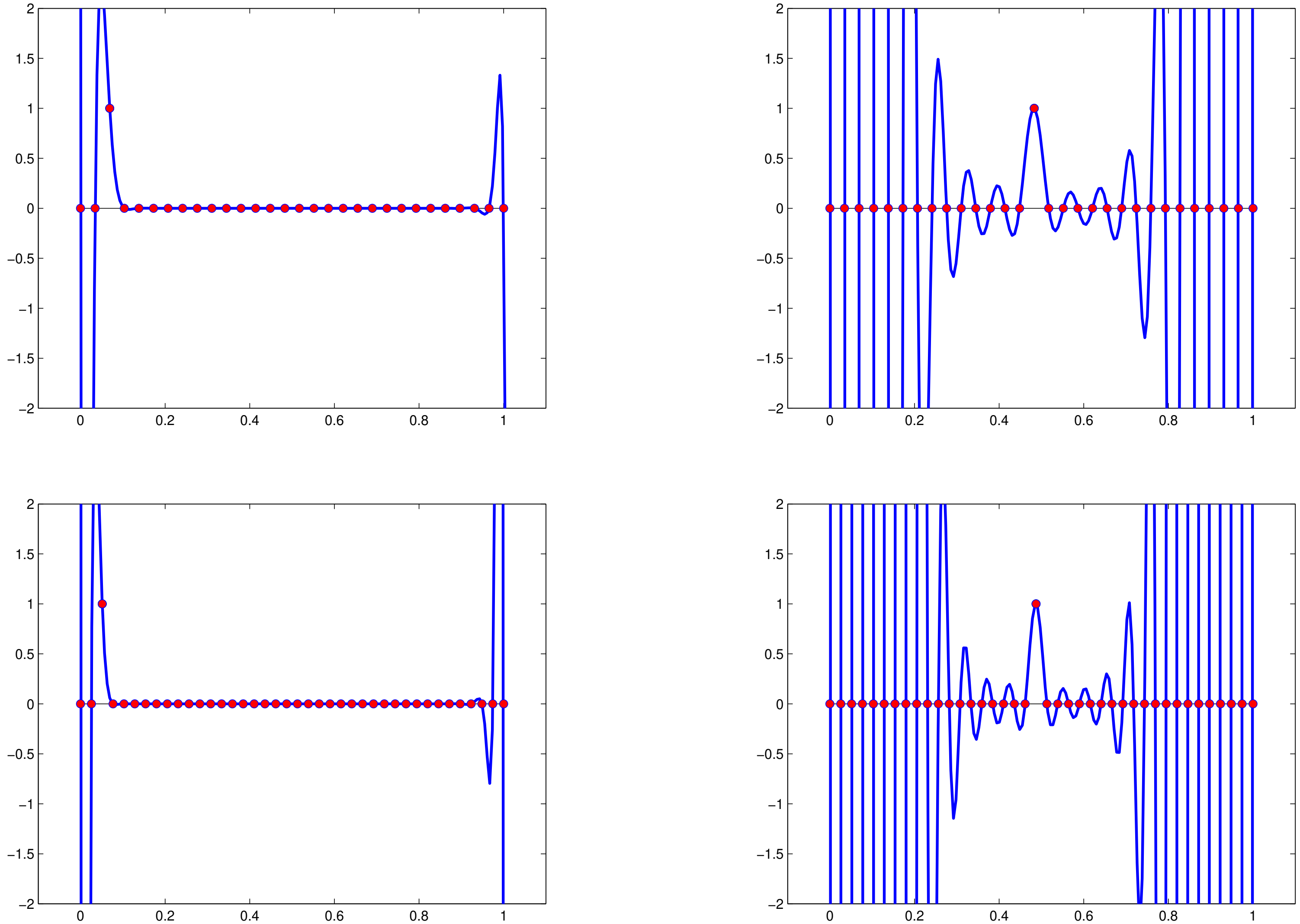
Anschauung: einige Langrange-Basispolynome

Die Bilder zeigen einige Lagrange-Basispolynome (blaue Kurven) zu 10 äquidistanten Stützstellen (rot markiert).



Anschauung: einige Lagrange-Basispolynome

Die Bilder zeigen Lagrange-Basispolynome (blaue Kurven) zu 30 (oben) bzw. 40 (unten) äquidistanten Stützstellen (rot markiert).



Folgerung aus dem Lagrange-Ansatz: die baryzentrische Formel

Es gibt einen genialen Trick, um ein Interpolationspolynom numerisch stabil auszuwerten: Man stellt es als Quotient von Partialbruchzerlegungen dar. Diese Darstellung heißt baryzentrische Formel. Wir betrachten hier beispielhaft 3 Stützpunkte. Sei p das Polynom welches die Bedingungen $p(x_1) = f_1, p(x_2) = f_2, p(x_3) = f_3$ erfüllt. Dann ist

$$\begin{aligned} p(x) &= f_1 \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} + f_2 \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} + f_3 \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)} \\ &= (x-x_1)(x-x_2)(x-x_3) * \\ &\quad \left(f_1 \frac{\frac{1}{x-x_1}}{(x_1-x_2)(x_1-x_3)} + f_2 \frac{\frac{1}{x-x_2}}{(x_2-x_1)(x_2-x_3)} + f_3 \frac{\frac{1}{x-x_3}}{(x_3-x_1)(x_3-x_2)} \right) \\ &= (x-x_1)(x-x_2)(x-x_3) \left(f_1 \frac{w_1}{x-x_1} + f_2 \frac{w_2}{x-x_2} + f_3 \frac{w_3}{x-x_3} \right), \qquad (*) \end{aligned}$$

wobei

$$w_1 = \frac{1}{(x_1-x_2)(x_1-x_3)}, \quad w_2 = \frac{1}{(x_2-x_1)(x_2-x_3)}, \quad w_3 = \frac{1}{(x_3-x_1)(x_3-x_2)}.$$

Insbesondere hat das konstante Polynom mit Wert 1 die Darstellung

$$1 = (x-x_1)(x-x_2)(x-x_3) \left(\frac{w_1}{x-x_1} + \frac{w_2}{x-x_2} + \frac{w_3}{x-x_3} \right). \qquad (**)$$

Teilen von $(*)$ durch $(**)$ ergibt

$$p(x) = \frac{f_1 \frac{w_1}{x-x_1} + f_2 \frac{w_2}{x-x_2} + f_3 \frac{w_3}{x-x_3}}{\frac{w_1}{x-x_1} + \frac{w_2}{x-x_2} + \frac{w_3}{x-x_3}}$$

Das ist die **baryzentrische Formel** (Schwerpunktsformel) für p.

Die Newton-Methode I

Ziel: Schnelle Berechnung der Koeffizienten c_k des Interpolationspolynoms p für die Stützpunkte (x_j, f_j) bezüglich der problemangepassten Newton-Basis.

Darstellung des Interpolationspolynoms bzgl. der **Monom-Basis**:

$$p(x) = a_0 + a_1\, x + a_2\, x^2 + \ldots + a_n\, x^n.$$

Darstellung des Interpolationspolynoms bzgl. der angepassten **Newton-Basis**:

$$p(x) = c_0 + c_1\,(x - x_1) + c_2\,(x - x_1)(x - x_2) + \ldots + c_n\,(x - x_1)(x - x_2)\ldots(x - x_n).$$

Wir führen für die Koeffizienten c_j eine Bezeichnung ein.

Definition:

$$f[x_1,\ x_2\ \ldots, x_j] := c_j$$

Der Koeffizient c_j heißt **j-te dividierte Differenz**.

Grund für die Bezeichnung siehe die nächsten Seiten.

Bemerkung: Da es auf die Nummerierung der Stützpunkte nicht ankommt, gilt z.B:

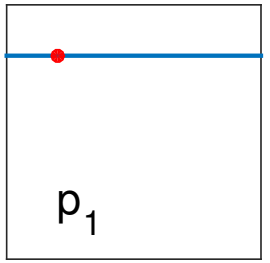
$$f[x_1,\ x_2,\ x_3] = f[x_1,\ x_2,\ x_3] = f[x_3,\ x_1,\ x_2].$$

Die Newton-Methode II

Wir betrachten die Interpolationspolynome für $1-3$ Stützpunkte:

Interpolationspolynom für einen Stützpunkt (x_1, f_1):

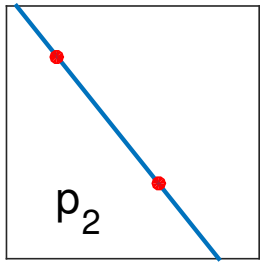
$$p_1(x) = f_1$$



Der Höchstkoeffizient ist f_1, also ist $f[x_1] = f_1$.

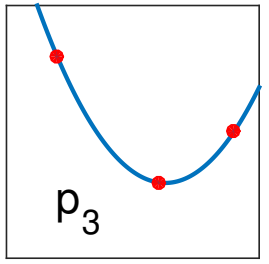
Interpolationspolynom für zwei Stützpunkte (x_1, f_1), (x_2, f_2):

$$p_2(x) = f_1 + \underbrace{\frac{f_2 - f_1}{x_2 - x_1}}_{f[x_1,\, x_2]}(x - x_1).$$



Interpolationspolynom für drei Stützpunkte (x_1, f_1), (x_2, f_2), (x_3, f_3):

$$p_3(x) = f_1 + \underbrace{\frac{f_2 - f_1}{x_2 - x_1}}_{f[x_1,\, x_2]}(x - x_1) + \underbrace{\frac{\frac{f_3 - f_1}{x_3 - x_1} - \frac{f_2 - f_1}{x_2 - x_1}}{x_3 - x_2}}_{f[x_1,\, x_2,\, x_3]}(x - x_1)(x - x_2)$$



Dies motiviert die Bezeichnung 'dividierte Differenz' für die Koeffizienten

$$f[x_1,\, x_2 \ldots, x_j].$$

Die Newton-Methode III

Interpolationspolynom für drei Stützpunkte (x_1, f_1), (x_2, f_2), (x_3, f_3):

$$p_3(x) = f_1 + \underbrace{\frac{f_2 - f_1}{x_2 - x_1}}_{f[x_1,\, x_2]} (x - x_1) + \underbrace{\frac{\frac{f_3-f_1}{x_3-x_1} - \frac{f_2-f_1}{x_2-x_1}}{x_3 - x_2}}_{f[x_1,\, x_2,\, x_3]} (x - x_1)(x - x_2)$$

Herleitung: Der Ansatz

$$p_3(x) = f_1 + \frac{f_2 - f_1}{x_2 - x_1}(x - x_1) + c\,(x - x_1)(x - x_2), \qquad c \in \mathbb{R}$$

ergibt ein Polynom mit

$$p_3(x_1) = f_1, \quad p_3(x_2) = f_2.$$

Der Koeffizient c ist so zu bestimmen, dass $p(x_3) = f_3$. Man hat allgemein

$$c = \frac{p_3(x) - f_1 - \frac{f_2-f_1}{x_2-x_1}(x - x_1)}{(x - x_1)(x - x_2)}.$$

Mit $p(x_3) = f_3$ folgt

$$c = \frac{f_3 - f_1 - \frac{f_2-f_1}{x_2-x_1}(x_3 - x_1)}{(x_3 - x_1)(x_3 - x_2)} = \frac{\frac{f_3-f_1}{x_3-x_1} - \frac{f_2-f_1}{x_2-x_1}}{x_3 - x_2}.$$

Die Newton-Methode IV

Rekursionsformel: $f[x_1] = f_1$ und

$$f[x_1,\ x_2,\ \ \ldots,\ x_k,\ x_{k+1}] = \frac{f[x_2,\ x_3,\ \ldots,\ x_k,\ \ x_{k+1}] - f[x_1,\ x_2,\ \ldots,\ x_k]}{x_{k+1} - x_1}.$$

Bemerkung: Auf die Reihenfolge der Werte in den eckigen Klammern und auf ihre Nummerierung kommt es bei der Berechnung nicht an.

Beispielrechnung:

Angenommen, es ist bekannt, dass

$$f[-1,\ 0.5,\ 7,\ 2] = 5, \qquad f[2,\ 9,\ 0.5,\ -1] = 3.$$

Die Listen in den eckigen Klammern unterscheiden sich abgesehen von der Reihenfolge der Einträge nur in den Zahlen 7 und 9. Setze (in Gedanken) $x_1 = 9$, $x_{n+1} = 7$. Dann ist die dividierte Differenz, welche zu allen vorkommenden Daten gehört, nach obiger Formel gegeben durch

$$f[-1,\ 0.5,\ 7,\ 2,\ 9] = \frac{f[2,\ 9,\ 0.5,\ -1] - f[-1,\ 0.5,\ 7,\ 2]}{9-7} = \frac{3-5}{9-7} = -1.$$

Newton-Methode V: Der Algorithmus zur Berechnung der dividierten Differenzen

Mit der Rekursionsformel kann man die dividierten Differenzen nacheinander ausrechnen. Dies führt auf das folgende

Schema zur Berechnung der dividierten Differenzen:

x_1	$f_1 = f[x_1]$	$f[x_1,\ x_2]$	$f[x_1,\ x_2,\ x_3]$	$f[x_1,\ x_2,\ x_3,\ x_4]$
x_2	$f_2 = f[x_2]$	$f[x_2,\ x_3]$	$f[x_2,\ x_3,\ x_4]$	
x_3	$f_3 = f[x_3]$	$f[x_3,\ x_4]$		
x_4	$f_4 = f[x_4]$			

Erläuterung: In den ersten beiden Spalten stehen die Interpolationsdaten. Die 3. Spalte rechnet man aus, indem man von zwei untereinander stehenden Einträgen der 2. Spalte die dividierte Differenz bildet. Die 4. Spalte rechnet man aus, indem man von zwei untereinander stehenden Einträgen der 3. Spalte die dividierte Differenz bildet, usw. In der oberen Zeile stehen dann die Koeffizienten des Interpolationspolynoms p bezüglich der zugehörigen Newton-Basis, d.h. es ist

$$\begin{aligned} p(x) \;=\; & f[x_1] + f[x_1,\ x_2]\,(x - x_1) + f[x_1,\ x_2,\ x_3]\,(x - x_1)(x - x_2) \\ & + f[x_1,\ x_2,\ x_3,\ x_4]\,(x - x_1)(x - x_2)(x - x_3). \end{aligned}$$

Wenn ein neuer Stützpunkt (x_5, f_5) hinzukommt, dann kann man im obigen Schema einfach eine neue untere Schrägzeile anfügen (siehe nächste Seite).

Schema zur Berechnung der dividierten Differenzen:

x_1	$f_1 = f[x_1]$	$f[x_1,\ x_2]$	$f[x_1,\ x_2,\ x_3]$	$f[x_1,\ x_2,\ x_3,\ x_4]$	$f[x_1,\ x_2,\ x_3,\ x_4,\ x_5]$
x_2	$f_2 = f[x_2]$	$f[x_2,\ x_3]$	$f[x_2,\ x_3,\ x_4]$	$f[x_2,\ x_3,\ x_4,\ x_5]$	
x_3	$f_3 = f[x_3]$	$f[x_3,\ x_4]$	$f[x_3,\ x_4,\ x_5]$		
x_4	$f_4 = f[x_4]$	$f[x_4,\ x_5]$			
x_5	$f_5 = f[x_5]$				

Erläuterung: Wenn man die blau gefärbten Werte schon berechnet hat, dann kann man anschließend die roten mit der Rekursionsformel berechnen.

Eine geschlossene Formel für die dividierten Differenzen

Das Interpolationspolynom zu den Stützpunkten (x_j, f_j), $j = 1, \ldots, n+1$ ist

nach Newton:

$$p(x) \;=\; f[x_1] + f[x_1,\, x_2]\,(x - x_1) + \ldots + f[x_1,\, x_2,\, \ldots,\, x_{n+1}]\,(x - x_1)(x - x_2)\ldots(x - x_n),$$

nach Lagrange:

$$p(x) \;=\; \sum_{k=1}^{n+1} f_k\, L_k(x) \;=\; \sum_{k=1}^{n+1} f_k\, \frac{(x - x_1)\ldots(x - x_{k-1})(x - x_{k+1})\ldots(x - x_{n+1})}{(x_k - x_1)\ldots(x_k - x_{k-1})(x_k - x_{k+1})\ldots(x_k - x_{n+1})}.$$

Vergleich der Koeffizienten vor der höchsten Potenz x^n ergibt

$$f[x_1,\, x_2,\, \ldots,\, x_{n+1}] = \sum_{k=1}^{n+1} \frac{f_k}{(x_k - x_1)\ldots(x_k - x_{k-1})(x_k - x_{k+1})\ldots(x_k - x_{n+1})}.$$

Beispiel ($n = 2$):

$$f[x_1,\, x_2,\, x_3] \;=\; \frac{f_1}{(x_1 - x_2)(x_1 - x_3)} + \frac{f_2}{(x_2 - x_1)(x_2 - x_3)} + \frac{f_3}{(x_3 - x_1)(x_3 - x_2)}$$

Dividierte Differenzen und Ableitungen I

Dividierte Differenzen sind Differenzenquotienten (höherer Ordnung). Man erwartet daher, einen Zusammenhang mit Ableitungen. Für eine dividierte Differenz erster Ordnung hat man den

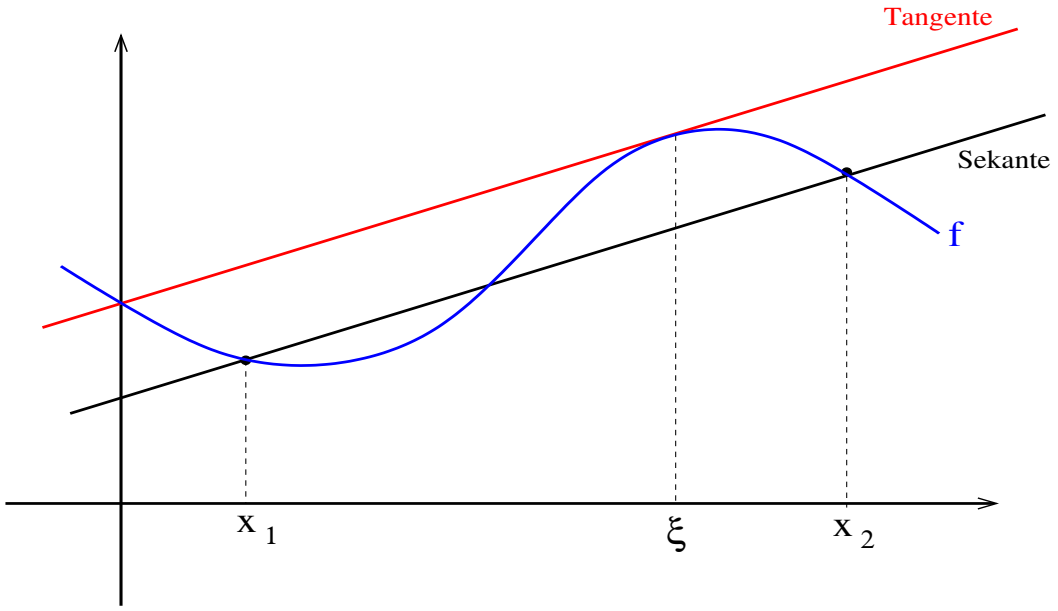
Mittelwertsatz der Differentialrechnung:

Sei $f : [a,\ b] \to \mathbb{R}$ stetig differenzierbar. Sei $a \leq x_1 < x_2 \leq b$.

Dann gibt es ein $\xi \in]x_1, x_2[$ so dass

$$\underbrace{\frac{f(x_2) - f(x_1)}{x_2 - x_1}}_{=f[x_1,\ x_2]} = f'(\xi).$$

(Kurz: die Sekantensteigung ist gleich der Tangentensteigung an einer Zwischenstelle ξ)



Dividierte Differenzen und Ableitungen II

Verallgemeinerter Mittelwertsatz:

Sei $f : [a, b] \to \mathbb{R}$ $(n+1)$-mal differenzierbar und seien $a \leq x_1 < x_2 < \ldots < x_{n+1} \leq b$.
Dann gibt es ein $\xi \in]x_1, x_{n+1}[$, so dass

$$f[x_1,\ x_2,\ \ldots,\ x_n,\ x_{n+1}] = \frac{f^{(n)}(\xi)}{n!}.$$

Beweis: Sei p das Interpolationspolynom zu den Daten $(x_j, f(x_j))$. Dann ist

$$p(x) = q(x) + f[x_1,\ x_2,\ \ldots,\ x_n,\ x_{n+1}]\ (x - x_1)(x - x_2)\ldots(x - x_n).$$

mit einem Polynom q mit $grad(q) \leq n - 1$. Leitet man p n-mal ab, dann bekommt man eine Konstante, nämlich

$$p^{(n)}(x) = f[x_1,\ x_2,\ \ldots,\ x_n,\ x_{n+1}]\ \ n!. \qquad (*)$$

Betrachte nun die Hilfsfunktion $\phi(x) = f(x) - p(x)$. ϕ hat die $n+1$ Nullstellen $x_1, \ldots, x_{n+1}$. Zwischen den Nullstellen hat ϕ lokale Extrema (Maxima oder Minima). Dies ergibt mindestens n Nullstellen der Ableitung ϕ'. Zwischen den Nullstellen hat ϕ' lokale Extrema. Dies ergibt mindestens $n - 1$ Nullstellen für die 2. Ableitung, usw.. Die n-te Ableitung hat mindestens eine Nullstelle ξ. Es ist also

$$0 = \phi^{(n)}(\xi) = f^{(n)}(\xi) - f[x_1,\ x_2,\ \ldots,\ x_n,\ x_{n+1}]\ \ n!.$$

Dividierte Differenzen und Ableitungen III

Aus dem **verallgemeinerten Mittelwertsatz**

$$f[x_1,\ x_2,\ \ldots,\ x_n,\ x_{n+1}] = \frac{f^{(n)}(\xi)}{n!}, \qquad \min_j x_j < \xi < \max_j x_j.$$

folgt durch Grenzübergang $x_j \to x_0$,

$$\lim_{x_1 \to x_0} \lim_{x_2 \to x_0} \ldots \lim_{x_{n+1} \to x_0} f[x_1,\ x_2,\ \ldots,\ x_n,\ x_{n+1}] = \frac{f^{(n)}(x_0)}{n!}.$$

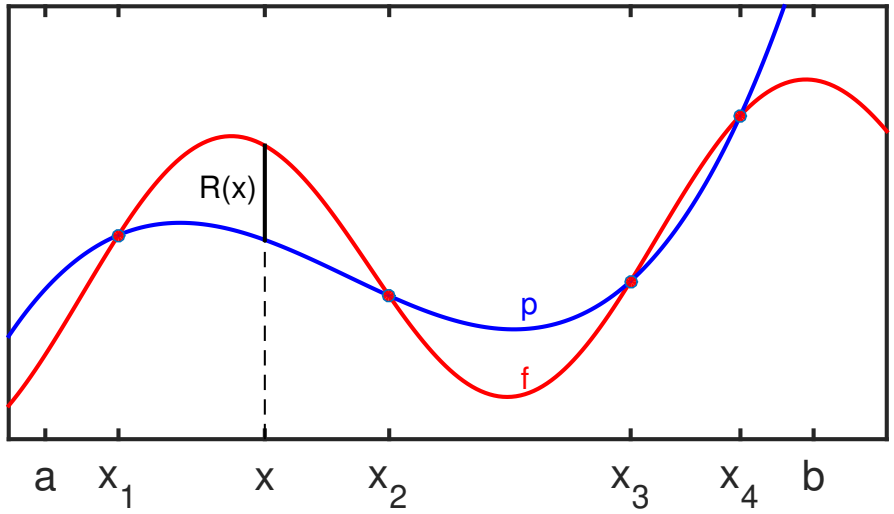
Aus diesem Grund definiert man:

$$f[\underbrace{x_0,\ x_0,\ \ldots,\ x_0,\ x_0}_{(n+1)\text{-mal}}] := \frac{f^{(n)}(x_0)}{n!}.$$

Dividierte Differenzen, bei denen einige, aber nicht alle Stützstellen gleich sind, definiert man über die Rekursionsformel. Beispiel:

$$f[x_0, x_0, x_1] := \frac{f[x_0, x_0] - f[x_0, x_1]}{x_0 - x_1} = \frac{\frac{f''(x_0)}{2} - \frac{f(x_0) - f(x_1)}{x_0 - x_1}}{x_0 - x_1}.$$

Das Restglied I



Satz: Gegeben sei eine Funktion $f : [a, b] \to \mathbb{R}$, die an den Stellen $a \leq x_1 < \ldots < x_{n+1} \leq b$ durch das Polynom p vom Grad $\leq n$ interpoliert wird, d.h. $p(x_j) = f(x_j) =: f_j$ und

$$p(x) = \sum_{j=0}^{n} f[x_1, \ldots x_{j+1}]\ (x - x_1) \ldots (x - x_j) \qquad \text{(Newton-Darstellung)}$$

Dann gilt für alle $x \in [a, b] \setminus \{x_1, \ldots, x_{n+1}\}$ die **Restgliedformel**

$$f(x) \quad = \quad p(x) + \mathbf{R}(x)$$

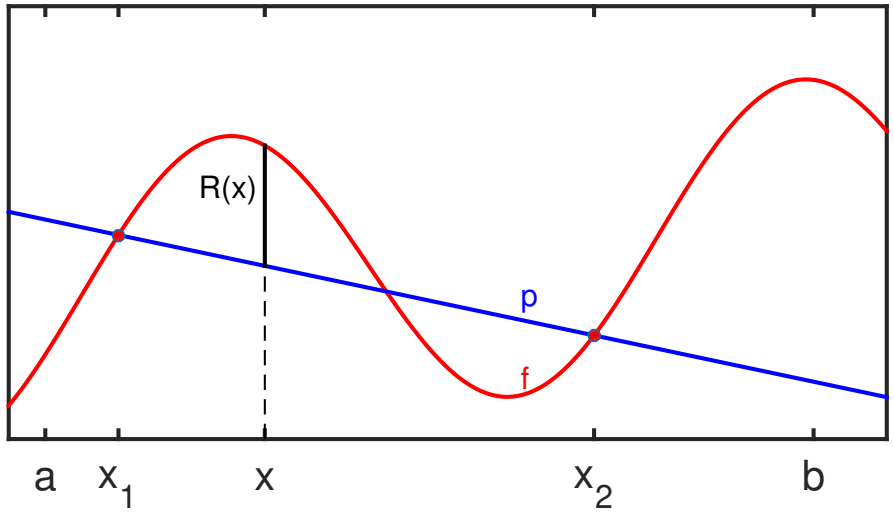
mit dem Restglied

$$\mathbf{R}(x) \quad = \quad f[x_1,\ x_2,\ \ldots,\ x_n,\ x_{n+1},\ x]\ (x - x_1)(x - x_2) \ldots (x - x_{n+1})$$

Wenn f $(n+1)$-mal stetig differenzierbar ist, dann ist für ein (von x abhängiges) $\xi \in [a, b]$,

$$f[x_1,\ x_2,\ \ldots,\ x_n,\ x_{n+1},\ x] \quad = \quad \frac{f^{(n+1)}(\xi)}{(n+1)!}$$

Das Restglied II



Beispiel: Interpolation durch eine Gerade (Sekante).

$$f(x) = \underbrace{f(x_1) + \frac{f(x_2)-f(x_1)}{x_2-x_1}(x-x_1)}_{\text{Sekante } p} + \underbrace{\frac{\frac{f(x)-f(x_2)}{x-x_2} - \frac{f(x_2)-f(x_1)}{x_2-x_1}}{x-x_1}}_{f[x_1,\,x_2,\,x]}(x-x_1)(x-x_2)$$

$$= f(x_1) + \frac{f(x_2)-f(x_1)}{x_2-x_1}(x-x_1) + \frac{f''(\xi)}{2}(x-x_1)(x-x_2)$$

$$= \underbrace{f(x_1)\frac{x-x_2}{x_1-x_2} + f(x_2)\frac{x-x_1}{x_2-x_1}}_{\text{Lagrange-Darstellung von } p} + \frac{f''(\xi)}{2}(x-x_1)(x-x_2)$$

Das Restglied III

Noch einmal die Restgliedformel:

Sei p das Interpolationspolynom für $f : \mathbb{R} \to \mathbb{R}$ an den Stützstellen $x_1, \ldots, x_{n+1} \in \mathbb{C}$.
Dann gilt für alle $x \in \mathbb{R}$:

$$f(x) = p(x) + \underbrace{f[x_1,\ x_2,\ \ldots,\ x_n,\ x_{n+1},\ x]\ (x - x_1)(x - x_2)\ldots(x - x_{n+1})}_{\text{Restglied}},$$

Beweis: Das Interpolationspolynom zu $x_1, \ldots, x_{n+1}, x$ ist

$$q(\tilde{x}) = p(\tilde{x}) + f[x_1,\ x_2,\ \ldots,\ x_n,\ x_{n+1},\ x]\ (\tilde{x} - x_1)(\tilde{x} - x_2)\ldots(\tilde{x} - x_{n+1})$$

Nach Definition ist $f(x) = q(x)$.

Einschub: Taylorformel als Grenzfall

Wir haben eben bewiesen, dass

$$f(x) \;=\; \underbrace{\sum_{j=0}^{n} f[x_1,\dots x_{j+1}]\,(x-x_1)\dots(x-x_j)}_{\text{Interpolationspolynom } p(x)} + \underbrace{\frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_1)(x-x_2)\dots(x-x_{n+1})}_{\text{Restglied}}$$

(Der erste Summand ($j=0$) ist $[x_1]f = f(x_1)$). Wendet man auf jeden der Koeffizienten $f[x_1,\dots x_{j+1}]$ den verallgemeinerten Mittelwertsatz an, so bekommt man

$$f(x) \;=\; \underbrace{\sum_{j=0}^{n} \frac{f^{(j)}(\xi_j)}{j!}\,(x-x_1)\dots(x-x_j)}_{\text{Interpolationspolynom } p(x)} + \underbrace{\frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_1)(x-x_2)\dots(x-x_{n+1})}_{\text{Restglied}}$$

mit ξ_j zwischen x_1 und x_j. Nun lässt man die Stellen x_j zu einem einzigen Punkt x_0 zusammenlaufen: $x_j \to x_0,\ \ j = 1,\dots,n+1$. Dann konvergieren die ξ_j gegen x_0. Die Produkte $(x-x_1)\dots(x-x_j)$ konvergieren gegen $(x-x_0)^j$. Man bekommt so die Taylorformel

$$f(x) \;=\; \underbrace{\sum_{j=0}^{n} \frac{f^{(j)}(x_0)}{j!}\,(x-x_0)^j}_{\text{Taylorpolynom}} + \underbrace{\frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)^{n+1}}_{\text{Restglied}}$$

mit ξ zwischen x_0 und x.

Restgliedminimierung bei variablen Stützstellen.

Gegeben seien die Stützstellen $a \leq x_1 < \ldots < x_{n+1} \leq b$. Aus der Restgliedformel folgt:

$$\max_{x\in[a,b]} |f(x) - p(x)| \leq \frac{1}{(n+1)!} \max_{\xi\in[a,b]} |f^{(n+1)}(\xi)| \max_{x\in[a,b]} |\omega_{n+1}(x)|.$$

wobei

$$\omega_{n+1}(x) = (x - x_1)(x - x_2)\ldots(x - x_{n+1})$$

Man kann zeigen, dass für jede Wahl von Stützstellen gilt:

$$2\left(\frac{b-a}{2}\right)^{n+1} \leq \max_{x\in[a,b]} |\omega_{n+1}(x)| \leq (b-a)^{n+1}$$

Das Maximum ist dann am kleinsten, wenn

$$2\left(\frac{b-a}{2}\right)^{n+1} = \max_{x\in[a,b]} |\omega_{n+1}(x)|$$

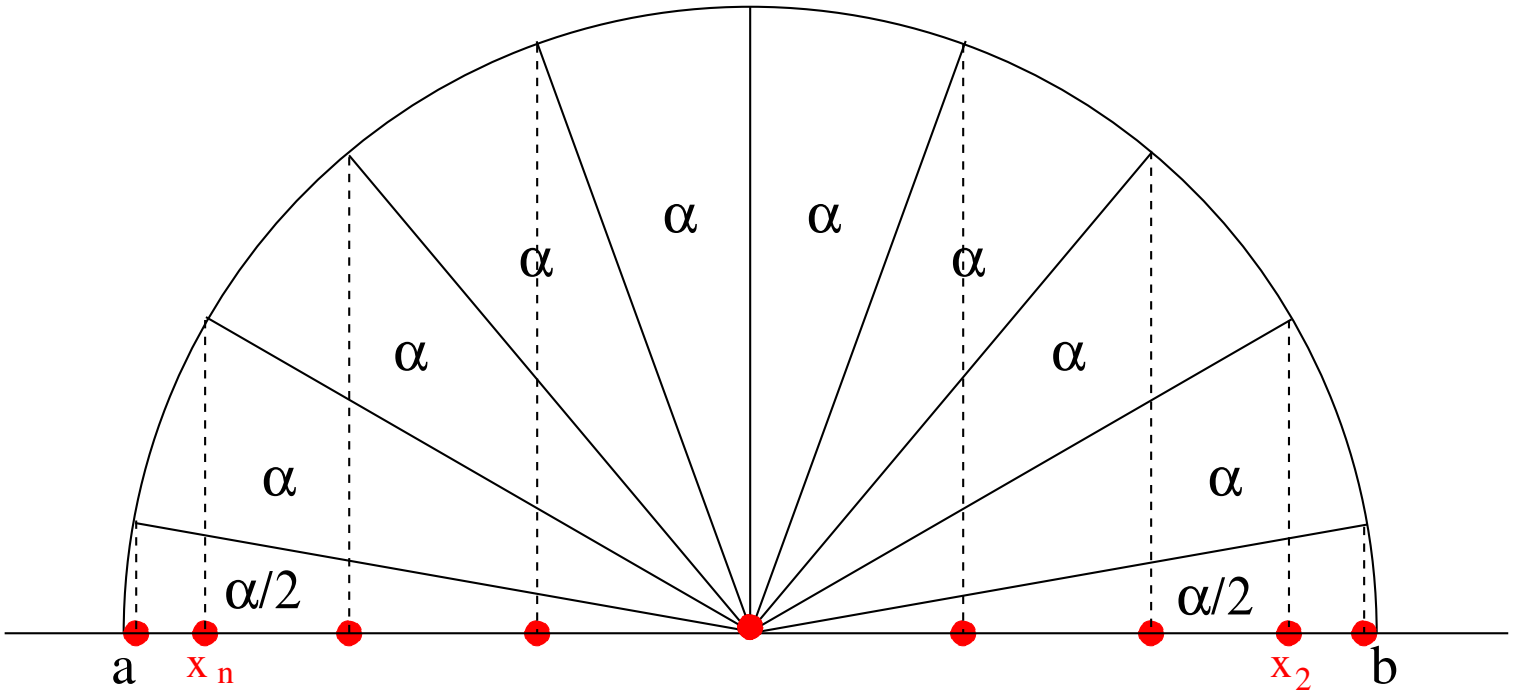
Dies ist der Fall, wenn die Stützstellen folgendermaßen gewählt sind:

$$x_j = \frac{a+b}{2} + \frac{b-a}{2}\,\tau_{n+1,j}, \qquad \tau_{n+1,j} = \cos\left(\frac{2j-1}{2(n+1)}\,\pi\right), \qquad j = 1,\ldots,n+1.$$

Die so gewählten x_j heißen **Tschebyscheff-Stützstellen**.
Die Zahlen $\tau_{n+1,j}$ sind die Nullstellen der Tschebyscheff-Polynome. (Siehe nächste Seiten)

Veranschaulichung: Lage der Tschebyscheff-Stützstellen

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



● = Tschebyscheff–Stuetzstellen $\qquad \alpha = \frac{\pi}{n+1}$

Die Tschebyscheff-Polynome

Mit Hilfe des Additionstheorems für Cosinus beweist man, dass es Polynome $T_n(x)$ mit ganzzahligen Koeffizienten und $grad(T_n) = n$ gibt, so dass

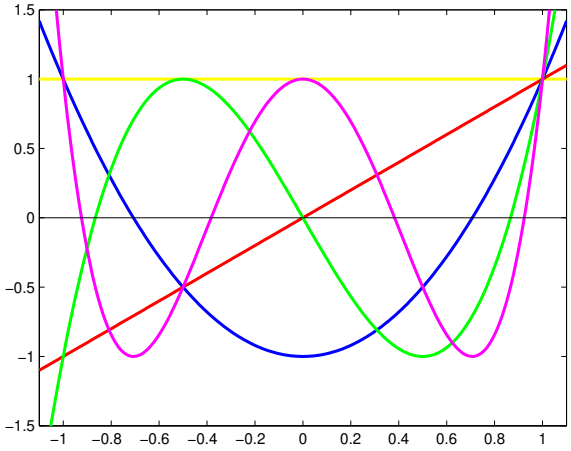
$$\cos(n\,\alpha) = T_n(\cos(\alpha)) \qquad (*)$$

Sei $x \in [-1, 1]$, so dass $\cos(\alpha) = x$. Dann ist $\alpha = \arccos(x)$ und Einsetzten in $(*)$ ergibt

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

Die Tschebyscheff-Polynome berechnet man mit der Rekursionsformel

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



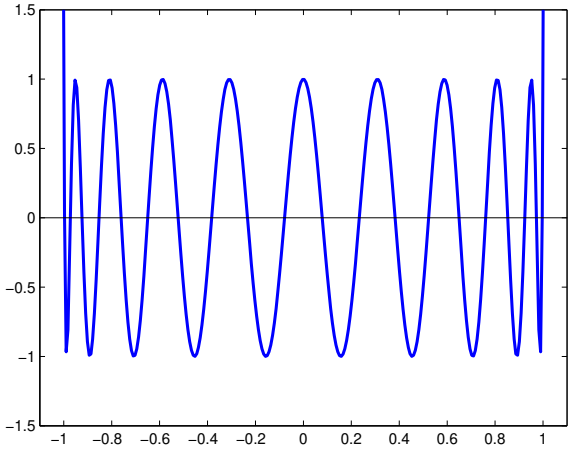


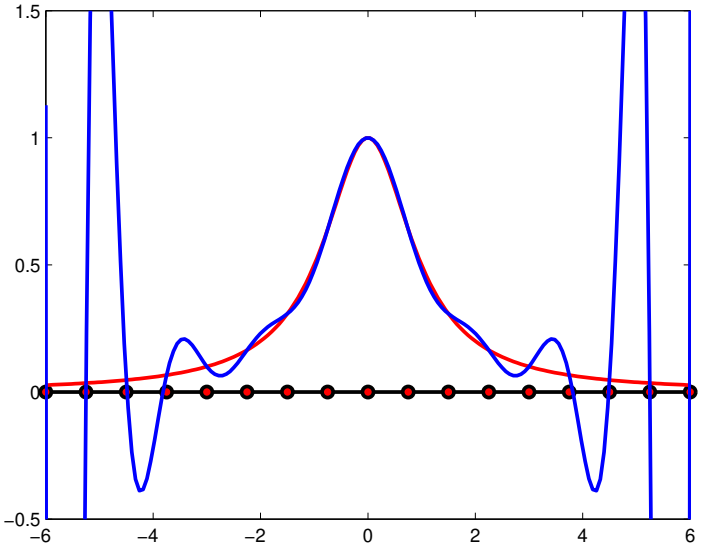
Bild links: Die Tschebyscheff-Polynome $T_0, \ldots, T_4$

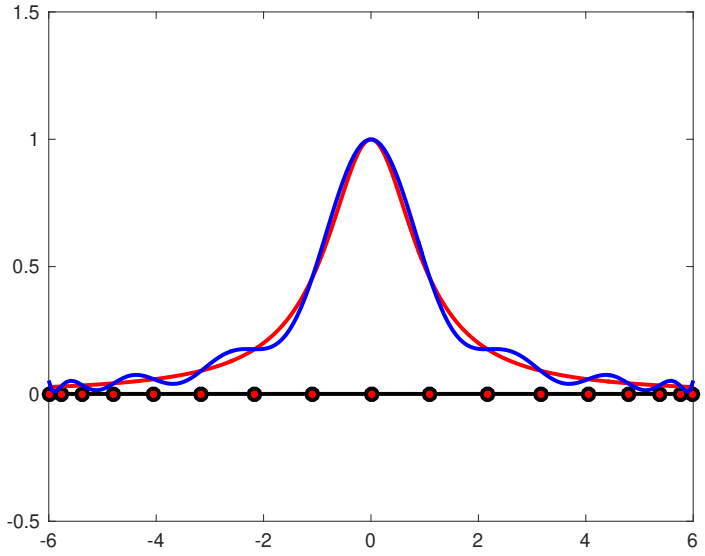
Bild rechts: Das Tschebyscheff-Polynom T_{20}

Interpolationsbeispiel (von Runge):

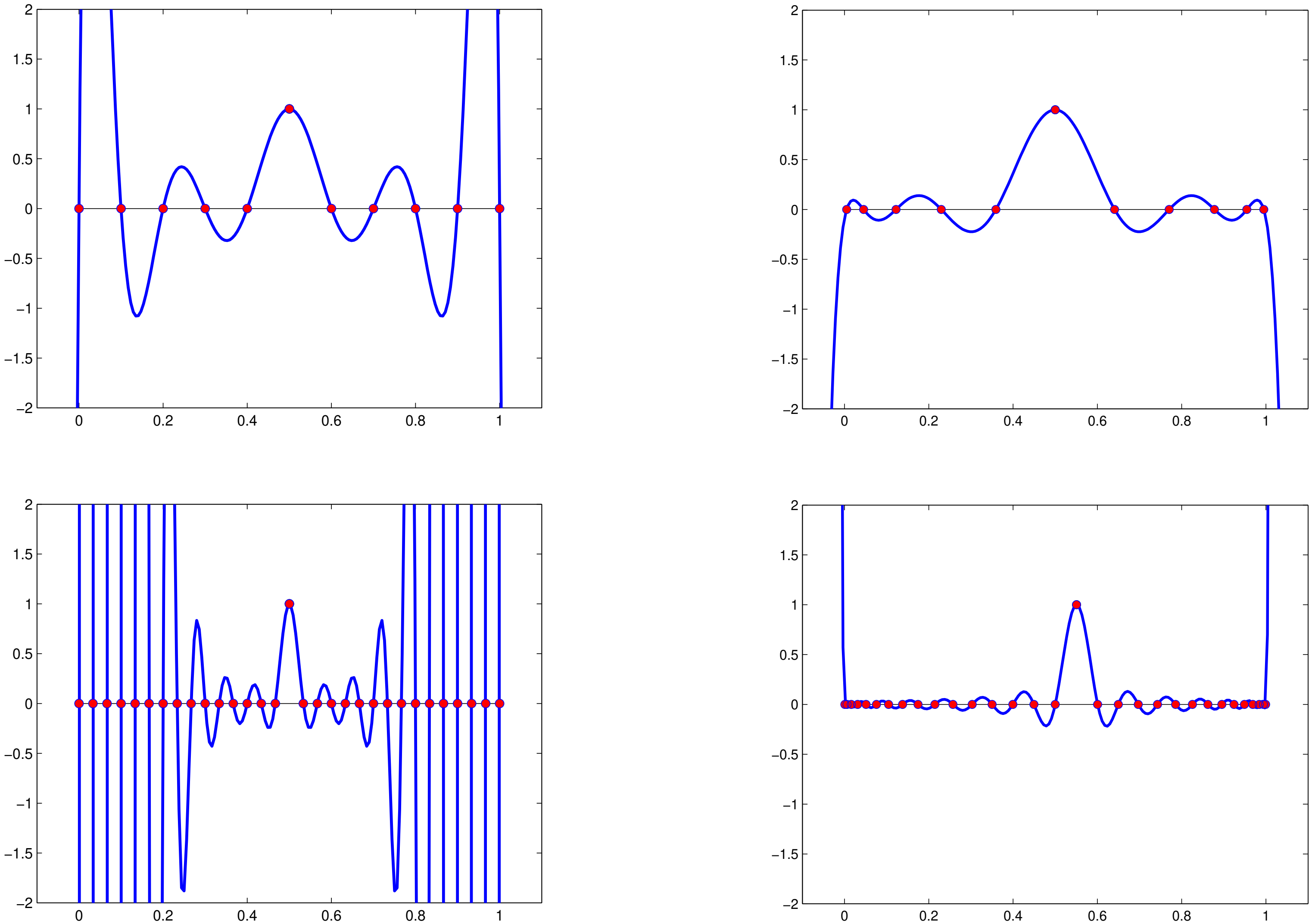
Interpolationpolynome (blau) für die Funktion $f(x) = \frac{1}{1+x^2}$ (rot) mit bei

17 äquidistanten Stützstellen (links) und 17 Tschebyscheff-Stützstellen (rechts)





Lagrange-Basispolynome bzgl. äquidistanter (links) und Tschebyscheff-Stützstellen (rechts)



Hermite-Interpolation

Bei der Hermite-Interpolation sind an den Stützstellen außer den Funktionswerten auch noch Werte für die Ableitung vorgegeben.

Das **Hermite-Interpolationsproblem** für Polynome lautet folgendermaßen:

Gegen seien die Stützstellen $x_1, \ldots, x_r \in \mathbb{C}$ und zu jeder Stützstelle x_j eine Liste von m_j Werten

$$f_j = f_j^{(0)}, \quad f_j' = f_j^{(1)}, \quad \ldots, f_j^{(m_j-1)} \in \mathbb{C}.$$

Bestimme ein Polynom p mit $grad(p) \leq m_1 + m_2 + \ldots m_r - 1 =: n$,
so dass für die Ableitungen von p gilt

$$p^{(k)}(x_j) = f_j^{(k)}, \qquad k = 0, \ldots, m_{j-1}. \qquad (*)$$

Vandermonde-Methode. Man kann zeigen, dass das Hermite-Interpolationproblem stets lösbar ist. Macht man einen Ansatz der Form

$$p(x) = c_1\, \beta_1(x) + c_2\, \beta_2(x) + \ldots + c_{n+1}\, \beta_{n+1}(x)$$

mit linear unabhängigen Polynomen $\beta_j(x)$ vom Grad $\leq n$ (z.B. $\beta_j(x) = x^j$) ,
dann lauten die Gleichungen $(*)$:

$$c_1\, \beta_1^{(k)}(x) + c_2\, \beta_2^{(k)}(x) + \ldots + c_{n+1}\, \beta_{n+1}^{(k)}(x) = f_j^{(k)} \qquad k = 0, \ldots, m_{j-1}$$

Dies ergibt ein lineares Gleichungssystem für die Koeffizienten c_k,
dessen Matrix nicht singulär ist.

Der einfachste Fall der Hermite-Interpolation: Das Taylor-Polynom

Aufgabe: Gegeben sei eine Funktion $f : U \to \mathbb{C}$, $U \subseteq \mathbb{R}$ oder $\mathbb{C}$, die an der Stelle $x_0 \in U$ n-mal differenzierbar ist. Bestimme ein Polynom p, mit $grad(p) \leq n$, so dass

$$p^{(k)}(x_0) = f^{(k)}(x_0), \qquad k = 0, \ldots, n \qquad (*)$$

Lösung: Die eindeutige Lösung ist das Taylor-Polynom

$$\begin{aligned} p(x) &= f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2}(x - x_0)^2 + \ldots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n \\ &= \sum_{k=0}^{n} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k. \end{aligned}$$

Begründung:

Für das Polynom $\beta_k(x) := (x - x_0)^k$ gilt

$$\beta_k'(x) = k\,(x - x_0)^{k-1}, \quad \beta_k''(x) = k(k-1)\,(x - x_0)^{k-2}, \quad \ldots \quad, \beta_k^{(k)}(x) = k!, \quad \beta_k^{(k+1)}(x) = 0.$$

Daraus folgt

$$\beta_k^{(\ell)}(x_0) = \begin{cases} k! & \text{falls } \ell = k \\ 0 & \text{sonst.} \end{cases} \qquad (**)$$

Der Ansatz $p(x) = c_0\,\beta_0(x) + c_1\,\beta_1(x) + \ldots + c_n\,\beta_n(x)$ ergibt:

$$f^{(k)}(x_0) \underset{(*)}{=} p^{(k)}(x_0) \underset{(**)}{=} c_k\,k! \qquad \Rightarrow c_k = f^{(k)}(x_0)/k!.$$

Der Kalkül der dividierten Differenzen für das Hermite-Problem

Der Kalkül der dividierten Differenzen lässt sich auf den Fall des Hermite-Interpolationsproblems erweitern, indem man im Ausdruck $f[x_1,\ x_2,\ \ldots,\ x_{n+1}]$ auch zulässt, dass einige oder alle x_j gleich sind. Im letzteren Fall definiert man

$$f\underbrace{[x_j,\ x_j,\ \ldots,\ x_j]}_{(n+1)-mal} := \frac{f^{(n)}}{n!}. \qquad (*)$$

Damit schreibt sich das Taylorpolynom n-ten Grades an der Stelle x_j so:

$$p(x) = \sum_{k=0}^{n} f\underbrace{[x_j,\ x_j,\ \ldots,\ x_j]}_{(k+1)-mal}\ (x - x_j)^k.$$

Dividierte Differenzen, bei denen nur einige x_j identisch sind, lassen sich durch die Rekursionsformel

$$f[x_1,\ x_2,\ \ldots,\ x_n,\ x_{n+1}] = \frac{f[x_2,\ x_3,\ \ldots,\ x_n,\ x_{n+1}] - f[x_1,\ x_2,\ \ldots,\ x_n]}{x_{n+1} - x_1}, \quad \text{wenn } x_{n+1} \neq x_n$$

auf den Fall $(*)$ zurückführen. Beispiel:

$$f[3,\ 3,\ 3,\ 7] = \frac{[3,\ 3,\ 7] - f[3,\ 3,\ 3]}{7-3} = \frac{f[3,\ 3,\ 7] - f''(3)/2}{7-3},$$

$$f[3,\ 3,\ 7] = \frac{f[3,\ 7] - f[3,\ 3]}{7-3} = \frac{\frac{f(7)-f(3)}{7-3} - f'(3)}{7-3}.$$

Der Kalkül der dividierten Differenzen für das Hermite-Problem II

Mit den dividierten Differenzen kann man die Lösung eines Hermite-Interpolationsproblems in Newtonscher Form darstellen. Hier kommt der für die Anwendung wichtigste Fall als Beispiel:

Problem: Bestimme eine Polynom p vom Grad ≤ 3 (kubisches Polynom), welches die folgenden Bedingungen erfüllt:

$$p(x_1) = f_1, \quad p'(x_1) = s_1, \qquad p(x_2) = f_2, \quad p'(x_2) = s_2,$$

wobei $x_1, x_2, f_1, f_2, s_1, s_2 \in \mathbb{C}$ vorgegebene Werte sind, und $x_1 \neq x_2$.

Lösung: Man macht den Ansatz

$$p(x) = f[x_1] + f[x_1,\ x_1]\ (x-x_1) + f[x_1,\ x_1,\ x_2]\ (x-x_1)^2 + f[x_1,\ x_1,\ x_2,\ x_2]\ (x-x_1)^2(x-x_2)$$

und rechnet

$$\begin{aligned}
f[x_1,\ x_2] &= \frac{f_1 - f_2}{x_1 - x_2} \\
f[x_1,\ x_1,\ x_2] &= \frac{f[x_1,\ x_1] - f[x_1,\ x_2]}{x_1 - x_2} = \frac{s_1 - \frac{f_1 - f_2}{x_1 - x_2}}{x_1 - x_2} \\
f[x_1,\ x_1,\ x_2,\ x_2] &= \frac{f[x_1,\ x_1,\ x_2] - f[x_1,\ x_2,\ x_2]}{x_1 - x_2} \\
&= \frac{\frac{f[x_1,x_1] - f[x_1,x_2]}{x_1 - x_2} - \frac{f[x_1,\ x_2] - f[x_2,\ x_2]}{x_1 - x_2}}{x_1 - x_2} = \frac{s_1 + s_2 - 2\,\frac{f_1 - f_2}{x_1 - x_2}}{(x_1 - x_2)^2}.
\end{aligned}$$

Die Hermite-Basispolynome (für 2 Stützstelllen und 1. Ableitungen)

Wir betrachten noch einmal das Problem von der vorigen Seite:

Problem: Bestimme eine Polynom p vom Grad ≤ 3 (kubisches Polynom), welches die folgenden Bedingungen erfüllt:

$$p(x_1) = f_1, \quad p'(x_1) = s_1, \qquad p(x_2) = f_2, \quad p'(x_2) = s_2. \qquad (*)$$

Die auf der vorigen Seite angegebene Darstellung der Lösung p ist unsymmetrisch in den Stützstellen x_1, x_2. Mit einer anderen Herangehensweise bekommt man folgende symmetrische Darstellung der Lösung:

Definiere

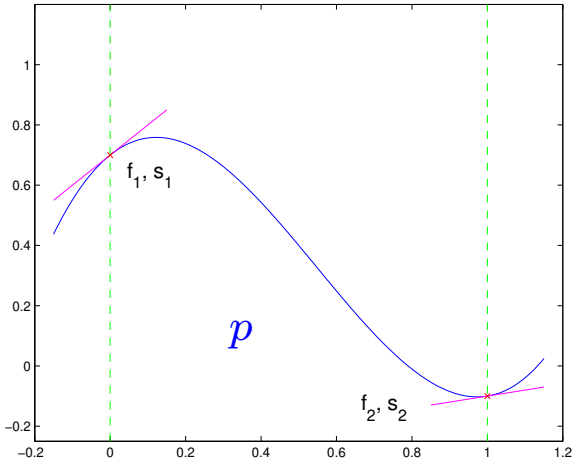
$$\begin{aligned} \beta_1(x) &:= \frac{(x-x_2)^2(x-x_1)}{(x_2-x_1)^2}, & \beta_3(x) &:= \frac{x-x_2-\beta_1(x)-\beta_2(x)}{x_1-x_2}, \\ \beta_2(x) &:= \frac{(x-x_1)^2(x-x_2)}{(x_2-x_1)^2}, & \beta_4(x) &:= \frac{x-x_1-\beta_1(x)-\beta_2(x)}{x_2-x_1}. \end{aligned}$$

Das Polynom p, welches $(*)$ erfüllt, hat die Darstellungen

$$\begin{aligned} p(x) &= s_1\,\beta_1(x) + s_2\,\beta_2(x) + f_1\,\beta_3(x) + f_2\,\beta_4(x) \qquad (**) \\ &= (s_1 - m)\,\beta_1(x) + (s_2 - m)\,\beta_2(x) + f_1\,\frac{x-x_2}{x_1-x_2} + f_2\,\frac{x-x_1}{x_2-x_1}, \qquad m := \frac{f_2-f_1}{x_2-x_1}. \end{aligned}$$

Beweis durch Nachrechnen (wie man drauf kommt, wird in der VL erklärt). Vorteil der Darstellung: Die Koeffizienten in $(**)$ sind dieselben wie in $(*)$. Hat man erst einmal die β_k bestimmt, muss man nichts mehr rechnen, um das Problem $(*)$ zu lösen. Die Polynome β_k heißen **Hermitesche Basispolynome**.

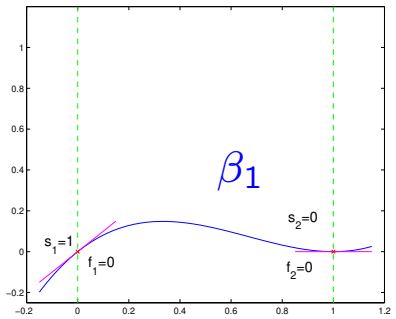
Anschauung: ein Polynom 3. Grades und die Hermite-Basispolynome

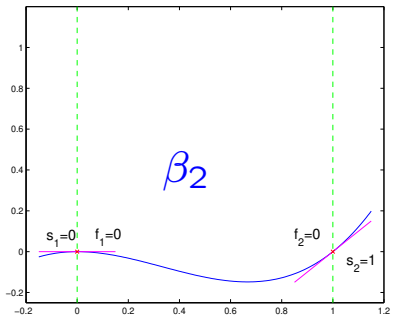


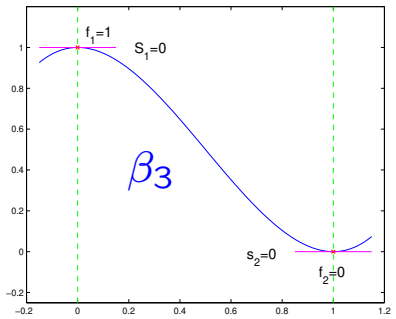
Jedes Polynom p mit $grad(p) \leq 3$ ist Linearkombination der 4 Hermite-Basispolynome:

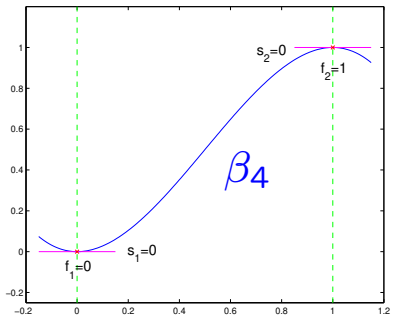
$$p(x) = s_1\ \beta_1(x) + s_2\ \beta_2(x) + f_1\ \beta_3(x) + f_2\ \beta_4(x).$$

Die Hermite-Basispolynome für das Intervall $[0, 1]$ sehen so aus:









Hermite-Basispolynome im allgemeinen Fall

Seien $x_1, \ldots, x_r \in \mathbb{C}$ paarweise verschieden. Seien außerdem die Werte

$$f_j^{(k)} \in \mathbb{C}, \qquad j = 0, \ldots, m_j - 1$$

vorgegeben.
Dann gibt es genau ein Polynom p mit $grad(p) \leq m_1 + m_2 + \ldots m_r - 1 =: n$,
so dass für die Ableitungen von p gilt

$$p^{(k)}(x_j) = f_j^{(k)}, \qquad k = 0, \ldots, m_{j-1}. \qquad (*)$$

Es ist

$$p(x) = \sum_{j=1}^{r} \sum_{k=0}^{m_j - 1} f_j^{(k)} \, L_{j,k}(x),$$

mit den **Hermite-Basispolynomen**

$$L_{j,k}(x) = \frac{(x - x_j)^k}{k!} \left(\sum_{i=0}^{m_j - k} \frac{g_j^{(i)}(x_j)}{i!} (x - x_j)^i \right) \ell_j(x),$$

wobei

$$\ell_j(x) = \prod_{\substack{1 \leq i \leq r \\ i \neq j}} (x - x_i)^{m_i}, \qquad g_j(x) = 1/\ell_j(x).$$

Spline-Interpolation

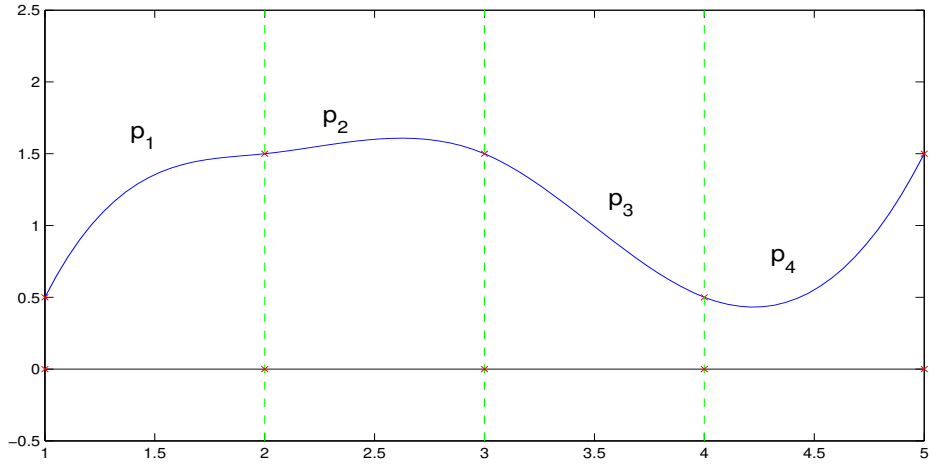
Wir haben gesehen, dass die Interpolation einer Funktion f mit Polynomen hohen Grades oft keine gute Näherung (Approximation) für die interpolierte Funktion f liefert. Der Grund dafür ist, dass Polynome hohen Grades dazu neigen, in der Umgebung der äußeren Interpolationspunkte stark zu schwanken. Bessere Approximationen bekommt man mit Splines. Dies sind Funktionen, die stückweise Polynome sind, welche an den Stützstellen glatt ineinander übergehen.

Definition: Ein Spline vom Grad k zu den Stützstellen

$$a = x_1 < x_2 < \ldots < x_{n+1} = b$$

ist eine Funktion $\sigma : [a, b] \to \mathbb{R}$ mit folgenden Eigenschaften:

- σ ist mindestens $(k-1)$-mal stetig differenzierbar.
- Auf jedem der Intervalle $[x_j, x_{j+1}]$ ist σ eine Polynomfunktion p_j vom Grad $\leq k$.



Kubische Splines

Die für die Anwendungen wichtigsten Splines sind die kubischen.

Definition: Ein kubischer Spline zu den Stützstellen (Knoten)

$$a = x_1 < x_2 < \ldots < x_{n+1} = b$$

ist eine Funktion $\sigma : [a, b] \to \mathbb{R}$ mit folgenden Eigenschaften:

- σ ist mindestens 2-mal stetig differenzierbar.
- Auf jedem der Intervalle $[x_j, x_{j+1}]$ ist σ eine Polynomfunktion p_j vom Grad ≤ 3.

Das Interpolationsproblem für kubische Splines lautet:
Finde zu gegebenen Stützwerten f_j einen kubischen Spline, so dass

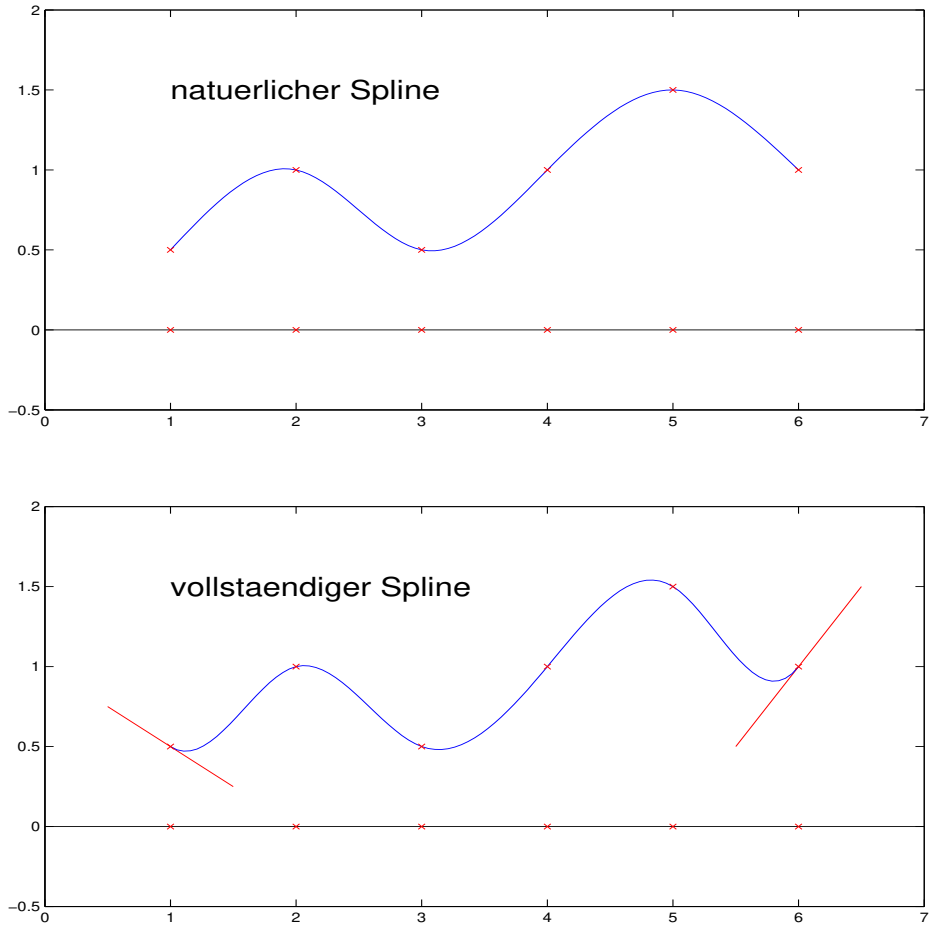
$$\sigma(x_j) = f_j, \qquad j = 1, \ldots, n+1.$$

Dieses Problem hat mehrere Lösungen. Um die Lösung eindeutig zu machen, muss man 2 Randbedingungen vorgeben. Die wichtigsten Fälle sind

1. $\sigma''(x_1) = \sigma''(x_{n+1}) = 0$ (natürlicher Spline)
2. $\sigma'(x_1) = s_1$, $\sigma'(x_{n+1}) = s_{n+1}$ (vollständiger Spline)
   Dabei sind $s_1, s_{n+1} \in \mathbb{R}$ vorgegebene Werte.
3. $\sigma'(x_1) = \sigma'(x_{n+1}), \quad \sigma''(x_1) = \sigma''(x_{n+1})$ (periodischer Spline)

Natürlicher und vollständiger kubischer Spline im Vergleich

Die Bilder zeigen Splines zu denselben Stützpunkten.



Natürlicher Spline: 2. Ableitungen am Rand sind 0.

Vollständiger Spline: 1. Ableitungen am Rand sind vorgegeben.

Zwei wichtige Eigenschaften von Splines

Splines haben minimale Gesamtkrümmung

Sei $\sigma : [a,b] \to \mathbb{R}$ ein naürlicher, vollständiger oder periodischer kubischer Spline zu den Stützstellen $a = x_1, \ldots, x_{n+1} = b$, und sei $f : [a,b] \to \mathbb{R}$ eine 2-mal stetig differenzierbare Funktion, die an allen Stützstellen dieselben Werte wie σ hat und ausserdem dieselben Randbedingungen erfüllt. Dann gilt

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

Eine Fehlerabschätzung

Sei $\sigma : [a,b] \to \mathbb{R}$ ein kubischer Spline zu den Stützstellen $a = x_1, \ldots, x_{n+1} = b$, und sei $f : [a,b] \to \mathbb{R}$ eine 4-mal stetig differenzierbare Funktion, die an allen Stützstellen dieselben Werte wie σ hat und ausserdem dieselben 1. Randableitungen hat. Dann gilt

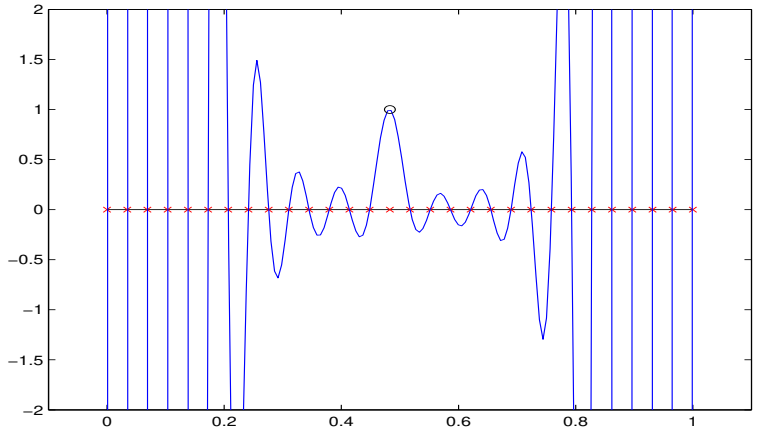
$$\max_{x\in[a,b]} |\sigma(x) - f(x)| \;\leq\; \frac{5}{384}\,h^4 \max_{x\in[a,b]} |f^{(4)}(x)|.$$

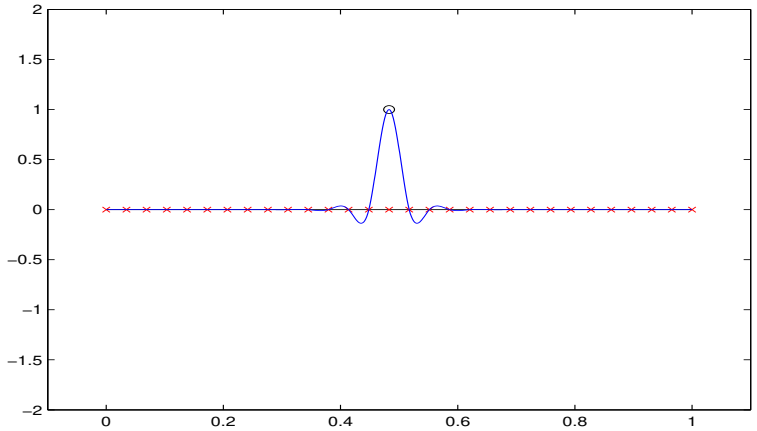
Dabei ist h der maximale Abstand zweier benachbarter Knoten x_j.

Bemerkung: Es gibt noch viele andere Fehlerabschätzungen ähnlicher Art.

Vergleich: Polynom- und Spline-Interpolation

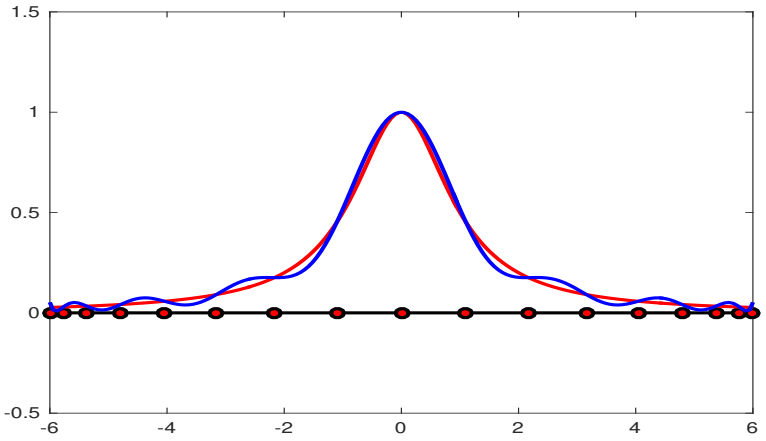
Links: Lagrange-Basispolynom. Rechts: Spline, welcher dieselben Punkte interpoliert.

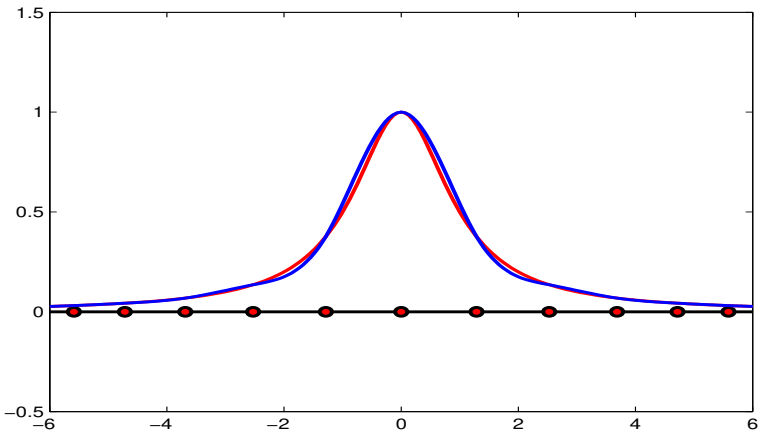




Links: Lagrange-Interpolation von $f(x) = 1/(1 + x^2)$ mit Tschebyscheff-Stützstellen.

Rechts: Spline-Interpolation von $f(x) = 1/(1 + x^2)$.





Wie man kubische Splines berechnet

Ein kubischer Spline σ mit

$$\sigma(x_j) = f_j, \qquad \sigma'(x_j) = s_j, \qquad j = 1, \ldots, n+1$$

ist aus Polynomen 3 Grades p_j zusammengesetzt. Diese haben die Darstellung

$$\sigma(x) = p_j(x) = (s_j - m_j)\,\beta_{j,1}(x) + (s_{j+1} - m_j)\,\beta_{j,2}(x) + f_j\,\frac{x - x_{j+1}}{x_{j+1} - x_j} + f_{j+1}\,\frac{x - x_j}{x_{j+1} - x_j}, \qquad (*)$$

wobei $x \in [x_j, x_{j+1}]$, $m_j := \frac{f_{j+1} - f_j}{x_{j+1} - x_j}$ und $\beta_{j,1}, \beta_{j,2}(x)$ die ersten beiden zum Intervall $[x_j, x_{j+1}]$ gehörenden Hermite-Basipolynome sind.

Die Forderung, dass σ am Knoten x_j 2mal differenzierbar ist, bedeutet

$$p''_{j-1}(x_j) = p''_j(x_j), \qquad j = 2, \ldots, n$$

Aus diesen Bedingungen folgen nach einigen Umformungen die folgenden linearen Gleichungen zwischen den Steigungen:

$$\lambda_j\, s_{j-1} + 2\, s_j + (1 - \lambda_j) s_{j+1} = 3\,(\lambda_j\, m_{j-1} + (1 - \lambda_j)\, m_j), \qquad \lambda_j = \frac{x_{j+1} - x_j}{x_{j+1} - x_{j-1}}.$$

Zusammen mit den Randbedingungen ergibt das ein lineares Gleichungssystem für alle Steigungen s_j. Hat man dieses gelöst, dann kann man mit Hilfe von $(*)$ den Spline berechnen.

Details stehen in jedem guten Numerik-Buch.