Wissenschaftlich praktische Arbeit

Interpolation

vorgelegt von
Steffen Weber und Nils Nichelmann

betreut durch
Frau Dr. Picht und Herrn Dr. Bruder

Georg-Cantor-Gymnasium Halle
31. Mai 2000

Überarbeitet von
Steffen Weber

Inhaltsverzeichnis

Kapitel 1

Polynom-Interpolation

Häufig tritt in der Nummerischen Mathematik die Situation auf, dass statt einer Funktion $f : \mathbb{R} \to \mathbb{R}$ nur einige diskrete Funktionswerte $f(x_i)$ und eventuell noch Ableitungen $f^{(j)}(x_i)$ an endlich vielen Punkten x_i gegeben sind. Dies trifft zum Beispiel zu, wenn die Funktion f in der Form experimenteller Daten vorliegt. Auch bei den meisten Verfahren zur Lösung von Differentialgleichungen wird die gesuchte Lösung $f(x)$ (einschließlich ihrer Ableitung) nur an endlich vielen Stellen berechnet. Historisch trat das Problem bei der Berechnung von zusätzlichen Funktionswerten zwischen tabellierten Werten auf. Heute ist eines der bedeutendsten Anwendungsfelder die Computergrafik, bekannt unter den Kürzeln CAD und CAGD.
Ist der gesamte Verlauf der Funktion interessant, so soll aus den gegebenen Daten

$$f^{(j)}(x_i), \qquad i = 0, ..., n, j = 0, ..., \ell_i$$

eine Funktion $F(x)$ konstruiert werden, die sich möglichst wenig von der ursprünglichen Funktion f unterscheidet. Zu dem sollte $F(x)$ leicht auswertbar sein, wie zum Beispiel Polynome, trigonometrische Funktionen, Expotentialfunktionen oder rationale Funktionen.

1.1 Existenz und Eindeutigkeit der Polynominterpolation

Gesucht ist eine Funktion $F(x)$, die die sogenannte *Interpolationbedingung* erfüllt: F soll an den *Knoten* oder *Stützpunkten* x_i mit der Funktion f übereinstimmen.

$$F^{(j)}(x_i) = f^{(j)}(x_i) \quad \forall i, j$$

Die Werte $f^{(j)}(x_i)$ heißen daher auch Stützwerte.

Bemerkung 1.1: Bei gegebenen $(n + 1)$ Stützstellen $(x_i, f(x_i))$, i = 0, 1 ,.., n lautet die *Interpolationsforderung*:

$$F(x_i) = a_0 + a_1 x_i^1 + a_2 x_i^2 + ... + a_n x_i^n = f(x_i) = f_i, \quad i = 0, 1, ..., n$$

Bei der Polynominterpolation ist daher ein Polynom n-ten Grades gesucht, das die Interpolationsforderung erfüllt. Es werden Polynome betrachtet, da diese einfach zu handhaben

und zu berechnen sind.
Im folgenden gilt daher

$$F(x) = p_n(x) = a_0 + a_1x + a_2x^2 + ... + a_{n-1}x^{n-1} + a_nx^n \tag{1.1}$$

Für gegebene $(n+1)$ Stützstellen $(x_i, f(x_i))$, i = 0, 1, ..., n, ergibt sich aus der Interpolationsforderung folgendes Gleichungssystem:

$$\underbrace{\begin{pmatrix} 1 & x_0 & x_0^2 & ... & x_0^n \\ 1 & x_1 & x_1^2 & ... & x_1^n \\ 1 & x_2 & x_2^2 & ... & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & ... & x_n^n \end{pmatrix}}_{\substack{\text{Koeffizientenmatrix/} \\ \text{Vandermonde'scheMatrix}}} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix} \tag{1.2}$$

Existiert eine eindeutige Lösung des Gleichungssystems, so gibt es genau ein Polynom n-ten Grades das die Aufgabe erfüllt.

Bemerkung 1.2: Eindeutigkeit
Ein lineares quadratisches Gleichungssystem ist genau dann eindeutig lösbar, wenn die Determinate der Koeffizientenmatrix ungleich null ist.

$$det \begin{pmatrix} 1 & x_0 & x_0^2 & ... & x_0^n \\ 1 & x_1 & x_1^2 & ... & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & ... & x_n^n \end{pmatrix} = \prod_{i=0}^{n-1} \prod_{j=i+1}^{n} (x_j - x_i)$$

Falls alle x_i paarweise verschieden sind, ist die Determinante der Koeffizientenmatrix ungleich null. $\Rightarrow$ Das Gleichungssystem (1.2) ist damit eindeutig lösbar.

Mit dem Gauß-Verfahren lässt sich das Gleichungssystem lösen und man erhält die Koeffizienten des gesuchten Polynoms $p_n(x)$. Das Horner-Schema ist eine effiziente Möglichkeit den Funktionswert eines Polynoms an einer Stelle x_i zu berechnen.

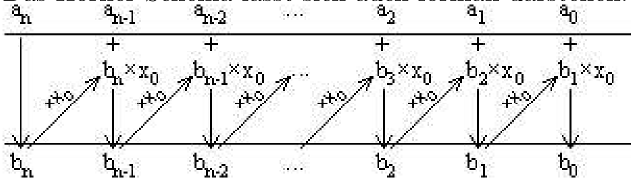
Bemerkung 1.3: Horner-Schema
Das Polynom wird durch fortlaufendes Ausklammern von x umgeformt und man erhält dadurch folgende Darstellung.

$$\begin{aligned} p_n(x) &= a_0 + a_1x + a_2x^2 + ... + a_{n-1}x^{n-1} + a_nx^n & (1.3) \\ p_n(x) &= a_0 + x(a_1 + a_2x^1 + ... + a_{n-1}x^{n-2} + a_nx^{n-1}) \\ p_n(x) &= a_0 + x(a_1 + x(a_2 + ... + a_{n-1}x^{n-3} + a_nx^{n-2})) \\ &\vdots \\ p_n(x) &= a_0 + x(a_1 + x(a_2 + x(... + x(a_{n-1} + xa_n)...))) & (1.4) \end{aligned}$$

Aus der Darstellung (1.4) ergibt sich folgender, leicht zu programmierender Algorithmus zur Berechnung des Funktionswertes eines Polynoms an einer Stelle x:
Algorithmus:
$x_0 := x;$

$b_n := a_n$;
for $i := n-1$ downto 0 do $b_i := b_{i+1}x_0 + a_i$;
$f_n(x_0) = b_0$

Das Horner-Schema lässt sich auch formall darstellen:



Das Horner-Schema hat den Vorteil, dass weniger Rechenschritte braucht werden als bei der Berechnung des Funktionswert eines Polynoms der Form (1.3). Um den Funktionswert eines Polynoms dritten Grades zu berechnen, muss man zum Beispiel neun Schritte (3 Additionen und 6 Multiplikationen) ausführen und nach dem Horner-Schema braucht man nur sechs Operationen (3 Multiplikationen und 3 Additionen). Weiterhin kann man das Horner-Schema auch für die Berechnung von Ableitungen benutzen. (Dazu benötigt man die im Beispiel 1.1. eingeführeten Variablen c, d, e.)

Beispiel 1.1.: Berechne den Funktionswert des Polynoms $p_3(x) = x^3 + 2x^2 + x + 1$ an der Stelle $x = x_0 = (-1)$.

$1 = a_3$	$2 = a_2$	$1 = a_1$	$1 = a_0$
$\Downarrow$	$-1 = b_3x_0$	$-1 = b_2x_0$	$0 = b_1x_0$
$1 = b_3$	$1 = b_2$	$0 = b_1$	$1 = b_0$
$\Downarrow$	$-1 = c_3x_0$	$0 = c_2x_0$	
$1 = c_3$	$0 = c_2$	$0 = c_1$	
$\Downarrow$	-1		
$1 = d_3$	$-1 = d_2$		
$\Downarrow$			
$1 = e3$			

Berechne den Funktionswert des Polynoms $p_5(x) = x^5 + x^3 + x + 1$ an der Stelle $x = x_0 = 1$.

1	0	1	0	1	1
$\Downarrow$	1	1	2	2	3
1	1	2	2	3	4
$\Downarrow$	1	2	4	6	
1	2	4	6	9	
$\Downarrow$	1	3	7		
1	3	7	13		
$\Downarrow$	1	4			
1	4	11			
$\Downarrow$	1				
1	5				
$\Downarrow$					
1					

1.2 Lagrange-Interpolation

Bei gegebenen (n + 1) paarweise verschiedenen Stützstellen $(x_i, f(x_i))$, i = 0, 1, ..., n, lässt sich mittels (1.2) immer eindeutig ein Polynom finden, das die Interpolationsforderung erfüllt. Aber selbst mit Hilfsmitteln wie dem Gauß-Algorithmus ist es schwierig, die Koeffi-

zienten eines Polynom höheren Grades zu bestimmen.
Für die Lagrange'sche Darstellung des Interpolationspolynoms wird folgender Ansatz gemacht:

$$p_n(x) = f_0 L_0(x) + f_1 L_1(x) + ... + f_n L_n(x) \tag{1.5}$$

Mit gegebenen $f(x_i) = f_i$ und paarweise verschiedenen x_i gilt:

$$L_i(x) = \prod_{k=0;k\neq i}^{n} \frac{x - x_k}{x_i - x_k}$$

Dabei heißen die $L_i(x)$ Lagrange'sche Multiplikatoren.

Satz 1.1: Die Lagrange'sche Darstellung erfüllt die Interpolationsforderung.

Beweis 1.1:
Voraussetzung: $x = x_j$
Behauptung: $p_n(x_j) = f(x_j)$, j = 0, 1, ..., n
Beweis:

$$\begin{aligned}
p_n(x_j) &= f_0 L_0(x_j) + f_1 L_1(x_j) + ... + f_{n-1} L_{n-1}(x_j) + f_n L_n(x_j) \\
& \sum_{i=0}^{n} f_i \prod_{k=0;k\neq i}^{n} \frac{x_j - x_k}{x_i - x_k} \\
& f_j + \sum_{i=0;i\neq j}^{n} f_i \prod_{k=0;k\neq i}^{n} \frac{x_j - x_k}{x_i - x_k} \\
& f_j + \sum_{i=0;i\neq j}^{n} f_i \frac{x_j - x_j}{x_i - x_j} \prod_{k=0;k\neq i,j}^{n} \frac{x_j - x_k}{x_i - x_k} \\
p_n(x_j) &= f_j
\end{aligned}$$

□

Satz 1.2: Sei $L_i(x) = \prod_{k=0;l\neq i}^{n} \frac{x-x_k}{x_i-x_k}$, dann gilt $\sum_{i=0}^{n} L_i(x) = 1$.
Beweis 1.2: Es gibt nur ein Polynom n-ten Grades $p_n(x)$, für das $p_n(x_j) = f_j$ gilt. Deshalb gibt es für die Stützwert $f_j = 1$, j = 0, 1, ..., n, auch nur das eindeutige Polynom $p_n(x) = 1$. Wegen $f_j = 1$, j = 0, 1, ..., n, ist $p_n(x) = \sum_{i=0}^{n} L_i(x) = 1$.

□

Bemerkung 1.4: Für die Lagrange'schen Multiplikatoren gilt:

$$L_i(x_j) = \begin{cases} 0, & \text{falls } i \neq j \\ 1, & \text{falls } i = j \end{cases}$$

Beispiel 1.2: Gegeben sind 4 Stützstellen $(x_i, f(x_i))$, i = 0, 1, 2. Stelle $p_3(x)$ als Lagrange'sches Interpolationspolynom dar.

x	-1	0	2	3
$f(x)$	-1	3	11	27

$$p_3(x) = \tfrac{-1}{-12}x(x-2)(x-3) + \tfrac{3}{6}(x+1)(x-2)(x-3) + \tfrac{11}{-6}(x+1)x(x-3) + \tfrac{27}{12}(x+1)x(x-2)$$

Bemerkung 1.5: Ein Nachteil der Lagrange-Interpolation ist, dass bei Hinzunahme einer weiteren Stützstelle die Lagrange'schen Multiplikatoren neu berechnet werden müssen.

1.3 Newton-Interpolation

Für die Newton'sche Darstellung des Interpolationspolynom wird folgender Ansatz gemacht:

$$p_n(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + ... + c_n(x - x_0)(x - x_1) \cdot ... \cdot (x - x_{n-1}) \quad (1.6)$$

Aus der Interpolationsforderung folgt:

$$\begin{pmatrix} 1 & 0 & 0 & ... & 0 \\ 1 & x_1 - x_0 & 0 & ... & 0 \\ 1 & x_2 - x_0 & (x_2 - x_0)(x_2 - x_1) & ... & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n - x_0 & (x_n - x_0)(x_n - x_1) & ... & (x_n - x_0) \cdot ... \cdot (x_n - x_{n-1}) \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix} \quad (1.7)$$

Um die Koeffizienten des Newton'schen Interpolationspolynoms zu berechenen, muss das Gleichungssystem (1.7) gelöst werden.
Die c_i werden als Steigungen bezeichnet.

Definition 1.1: k-te Steigung
Die k-te Steigung $f[x_i, ..., x_{i+k}]$ (auch: k-te dividierte Differenz) ist rekursiv definiert durch $f[x_i] := f(x_i)$ für i = 0, 1, ..., n und $f[x_i, ..., x_{i+k}] := \frac{f[x_{i+1},...,x_{i+k}] - f[x_i,...,x_{i+k-1}]}{x_{i+k} - x_i}$.

Es gilt $c_i = f[x_0, ..., x_i]$, i = 0, 1, ..., n.

Steigungs-/Differenzenschema:
Das Steigungs- oder Differenzenschema ist eine vorteilhafte Möglichkeit zur Berechnung der Steigungen/Differenzen.

i	x_i	0. Steigung $f[x_i]$	1. Steigung $f[x_i, x_{i+1}]$	...	n. Steigung $f[x_i, x_{i+1}, ..., x_{i+n}]$
0	x_0	$f[x_0] = f_0$			
			$f[x_0, x_1] = \frac{f_1 - f_0}{x_1 - x_0}$		
1	x_1	$f[x_1] = f_1$			
			$f[x_0, x_1] = \frac{f_2 - f_1}{x_2 - x_1}$		
2	x_2	$f[x_2] = f_2$			
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$f[x0, x_1, ..., x_n]$
$n-1$	x_{n-1}	$f[x_{n-1}] = f_{n-1}$			
			$f[x_{n-1}, x_n] = \frac{f_n - f_{n-1}}{x_n - x_{n-1}}$		
n	x_n	$f[x_n] = f_n$			

Doppeltes Steigungsschema:
Im Steigungsschema werden die Differenzen zwischen Stützstellen $(x_i - x_j)$ benötigt. Wenn man im Steigungsschema auf der linken Seite die Differenzen ergänzt, wird es übersichtlicher. Man erhält das doppelte Steigungsschema.

			x	$f(x)$			
			x_0	f_0			
		$x_1 - x_0$			$\frac{f_1-f_0}{x_1-x_0}$		
	$x_2 - x_0$		x_1	f_1		$\frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0}$	
$\ldots$		$x_2 - x_1$			$\frac{f_2-f_1}{x_2-x_1}$		$\ldots$
$\ldots$	$x_3 - x_1$		x_2	f_2		$\frac{f[x_2,x_3]-f[x_1,x_2]}{x_3-x_1}$	$\ldots$
$\ldots$		$\vdots$			$\vdots$		$\ldots$
		$x_n - x_{n-1}$			$\frac{f_n-f_{n-1}}{x_n-x_{n-1}}$		
			x_n	f_n			

Beispiel 1.3: Berechne die Koeffizienten des Newton-Interpolationspolynoms bei gegebenen Stützstellen ($x_0 = -1, f_0 = -1, x_1 = 0, f_1 = 3, x_2 = 2, f_2 = 11, x_3 = 3, f_3 = 27$) mit Hilfe des Steigungsschemas.

			x	$f(x)$			
			-1	-1			
		1			4		
	3		0	3		0	
4		2			4		1
	3		2	11		4	
		1			16		$\vdots$
			3	27			

$$p_3(x) = -1 + 4(x+1) + 0(x+1)(x-0) + 1(x+1)(x-0)(x-2)$$

Bezeichung: $p_{i,i+1,\ldots,i+k}(x)$ ist das eindeutig bestimmte Interpolationspolynom durch die Punkte $(x_i, f_i), (x_{i+1}, f_{i+1}), \ldots, (x_{i+k}, f_{i+k})$.

Satz 1.3: Ansatz (1.6) erfüllt die Interpolationsbedingung.
Beweis 1.3:
Behauptung:

$$p_{i,i+1,\ldots,i+k}(x) = \frac{(x - x_i)p_{i+1,i+2,\ldots,i+k}(x) - (x - x_{i+k})p_{i,i+1,\ldots,i+k-1}(x)}{x_{i+k} - x_i} \tag{1.8}$$

Fall 1: $x = x_i$

$$p_{i,i+1,i+2,\ldots,i+k}(x_i) = \frac{(x_i - x_i)p_{i+1,i+2,\ldots,i+k}(x_i) - (x_i - x_{i+k})p_{i,i+1,\ldots,i+k-1}(x_i)}{x_{i+k} - x_i}$$

$$p_{i,i+1,i+2,\ldots,i+k}(x_i) = \frac{(x_{i+k} - x_i)p_{i,i+1,\ldots,i+k-1}(x_i)}{x_{i+k} - x_i} = f_i$$

Da auch $p_{i,i+1,\ldots,i+k}(x_i) = f_i$ gilt, ist die Gleichung (1.8) für $x = x_i$ wahr.

Fall 2: $x = x_j$, j = i + 1, i + 2, ..., i + k - 1

Da sowohl $p_{i,i+1,\ldots,i+k-1}(x)$ als auch $p_{i+1,i+2,\ldots,i+k}(x)$ durch die Punkte (x_{i+1}, f_{i+1}), $(x_{i+2}, f_{i+2}), \ldots, (x_{i+k-1}, f_{i+k-1})$ geht, gilt ${}_{pi,i+1,\ldots,i+k-1}(x_j) = p_{i+1,i+2,\ldots,i+k}(x_j) = f_j$.
Daraus folgt

$$\frac{(x_j - x_i)p_{i+1,i+2,\ldots,i+k}(x_j) - (x_j - x_{i+k})_{pi,i+1,\ldots,i+k-1}(x_j)}{x_{i+k} - x_i} = \frac{(x_j - x_i - x_j + x_{i+k})f_j}{x_{i+k} - x_i}$$

Da auch $p_{i,i+1,...,i+k}(x_j) = f_j$ gilt, ist die Gleichung (1.8) für $x = x_j$, j = i + 1, i + 2, ..., i + k - 1 wahr.

Fall 3: $x = x_{i+k}$

$$p_{i,i+1,...,i+k}(x_{i+k}) = \frac{(x_{i+k} - x_i)p_{i+1,i+2,...,i+k}(x_{i+k}) - (x_{i+k} - x_{i+k})p_{i,i+1,...,i+k-1}(x_{i+k})}{x_{i+k} - x_i}$$

$$p_{i,i+1,...,i+k}(x_{i+k}) = \frac{(x_{i+k} - x_i)p_{i+1,i+2,...,i+k}(x_{i+k})f_{i+k}}{x_{i+k} - x_i}$$

Da auch $p_{i,i+1,...,i+k}(x_{i+k}) = f_{i+k}$ gilt, ist die Gleichung (1.8.) für $x = x_i$ wahr.
Da es keine weiteren Fälle gibt, ist die Gleichung (1.8) wahr.

□

Aus Gleichung (1.8) folgt

$$p_{i,i+1,...,i+k}(x) = \frac{(x - x_i)p_{i+1,i+2,...,i+k}(x) - (x - x_{i+k} + x_i - x_i)p_{i,i+1,...,i+k-1}(x)}{x_{i+k} - x_i}$$

$$\Rightarrow \quad p_{i,i+1,...,i+k}(x) = p_{i,i+1,...,i+k-1}(x) + (x - x_i)\frac{p_{i+1,i+2,...,i+k}(x) - p_{i,i+1,...,i+k-1}(x)}{x_{i+k} - x_i} \tag{1.9}$$

Gleichung (1.9) ist die rekursive Bildungsvorschrift des Neville-Algorithmusses zur Berechnung eines Funktionswertes, wenn weiterhin $p_i(x) = f_i$ und x fest gesetzt wird. Der Neville-Algorithmus ist dem Steigungsschema ähnlich.

Satz 1.4: Seien (x_q, f_q), q = 0, 1, ..., n, mit paarweise verschiedenen x_q gegeben. Dann ist

$$\begin{aligned} p_{i,i+1,...,i+k} = \; & f[x_i] + f[x_i, x_{i+1}](x - x_i) + ... \\ & + f[x_i, x_{i+1}, ..., x_{i+k}](x - x_i)(x - x_{i+1}) \cdot ... \cdot (x - x_{i+k-1}) \end{aligned}$$

mit $0 \le i \le i + k \le n$. Als Spezialfall ist $p_{0,1,...,n} = f[x_0] + f[x_0, x_1](x - x_0) + ... + f[x_0, x_1, ..., x_n](x - x_0)(x - x_1) \cdot ... \cdot (x - x_{n-1})$ das Polynom n-ten Grades, das für i = 0, 1, ..., n in den Punkten x_i den Wert f_i annimmt.

Beweis 1.4:

Induktionsanfang: Satz 1.4 gilt für $k = 0$, denn $p_i = f_i$ und $f[x_i] = f_i$.
Induktionsvoraussetzung: Satz 1.4 gilt für alle Polynome vom Grad $< k$.
Induktionsbehauptung: Satz 1.4 gilt für Polynome vom Grad k.
Induktionsbeweis:
Nach Gleichung (1.9) gilt

$$p_{i,i+1,...,i+k}(x) = p_{i,i+1,...,i+k-1}(x) + (x - x_i)\frac{p_{i+1,i+2,...,i+k}(x) - p_{i,i+1,...,i+k-1}(x)}{x_{i+k} - x_i}$$

Nach Induktionsvoraussetzung ist

$$p_{i,i+1,...,i+k-1}(x) = f[x_i] + ... + f[x_i, ..., x_{i+k-1}](x - x_i) \cdot ... \cdot (x - x_{i+k-2}),$$

somit ist nur noch folgende Gleichung zu zeigen:

$$f[x_i, ..., x_{i+k}](x - x_i) \cdot ... \cdot (x - x_{i+k-1}) = (x - x_i)\frac{p_{i+1,i+2,...,i+k}(x) - p_{i,i+1,...,i+k-1}(x)}{x_{i+k} - x_i}$$

$p_{i,i+1,...,i+k}(x)$ lässt sich als $p_{i,i+1,...,i+k}(x) = p_{i,i+1,...,i+k-1}(x) + c(x-x_i)\cdot...\cdot(x-x_{i+k-1})$ darstellen. Dabei ist c der Koeffizent bei x^k des Polynoms $p_{i,i+1,...,i+k}(x)$. Nach Induktionsvoraussetzung ist $f[x_i, x_{i+1}, ..., x_{i+k-1}]$ Koeffizient bei x^{k-1} für das Polynom $p_{i,i+1,...,i+k-1}(x)$ und $f[x_{i+1}, x_{i+2}, ..., x_{i+k}]$ Koeffizient bei x^{k-1} für das Polynom $p_{i+1,i+2,...,i+k}(x)$, da beide Polynome $(k-1)$-ten Grades sind.

$$\Rightarrow c = \frac{f[x_{i+1}, x_{i+2}, ..., x_{i+k}] - f[x_i, x_{i+1}, ..., x_{i+k-1}]}{x_{i+k} - x_i} = f[x_i, x_{i+2}, ..., x_{i+k}]$$

Aus Induktionsanfang und Induktionsschritt folgt nach dem Satz der vollständigen Induktion, dass Satz 1.4 für alle $k \in \mathbb{N}_0$ wahr ist.

$\square$

Demzufolge erhält man für die Newton'sche Darstellung folgenden Term:

$$\begin{aligned} p_n(x) &= f[x_0] + f[x_0, x_1](x-x_0) + f[x_0, x_1, x_2](x-x_0)(x-x_1) + ... \\ &\quad + f[x_0, x_1, ..., x_n](x-x_0)(x-x_1)\cdot ... \cdot(x-x_{n-1}) \end{aligned} \tag{1.10}$$

Falls die Sortierung der Stützwerte und -stellen in umgekehrter Reihenfolge erfolgt, so erhält man

$$\begin{aligned} p_n(x) &= f[x_n] + f[x_{n-1}, x_n](x-x_n) + f[x_{n-2}, x_{n-1}, x_n](x-x_n)(x-x_{n-1}) + ... \\ &\quad + f[x_0, x_1, ..., x_n](x-x_n)(x-x_{n-1})\cdot ... \cdot(x-x_1) \end{aligned} \tag{1.11}$$

Satz 1.5: Leibniz-Regel

Eine gegebene Funktion $f(x)$ sei als Produkt zweier anderer Funktionen $g(x)$ und $h(x)$ darstellbar: $f(x) = g(x)h(x)$.
Dann gilt die Leibniz-Regel:

$$f[x_0, x_1, ..., x_k] = \sum_{i=0}^{k} g[x_0, ..., x_i]h[x_i, ..., x_k] \tag{1.12}$$

Beweis 1.5:
Induktionsanfang: Gleichung (1.12) gilt für $k = 0$, denn $f[x_0] = f(x_0)$ und $\sum_{i=0}^{0} g[x_0, ..., x_i]h[x_i, ..., x_0] = g[x_0]h[x_0] = g(x_0)h(x_0) = f(x_0)$.
Induktionsvoraussetzung: Gleichung (1.12) gilt für $k = q$.
Induktionsbehauptung: Gleichung (1.12) gilt für $k = q + 1$.
Induktionsbeweis:

$$(x_{q+1} - x_0)f[x_0, x_1, ..., x_{q+1}] = f[x_1, x_2, ..., x_{q+1}] - f[x_0, x_1, ..., x_q]$$

$$(x_{q+1} - x_0)f[x_0, x_1, ..., x_{q+1}] = \sum_{i=1}^{q+1} g[x_1, ..., x_i]h[x_i, ..., x_{q+1}] - \sum_{i=0}^{q} g[x_0, ..., x_i]h[x_i, ..., x_q]$$

$$\begin{aligned} &= \sum_{i=1}^{q+1} g[x_1, ..., x_i]h[x_i, ..., x_{q+1}] - \sum_{i=0}^{q} g[x_0, ..., x_i]h[x_i, ..., x_q] \\ &\quad + \sum_{i=0}^{q} g[x_0, ..., x_i]h[x_{i+1}, ..., x_{q+1}] - \sum_{i=1}^{q+1} g[x_0, ..., x_{i-1}]h[x_i, ..., x_{q+1}] \end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{q} g[x_0,...,x_i](x_i - x0)h[x_i,...,x_{q+1}] + g[x_0,...,x_{q+1}](x_{q+1} - x0)h[x_{q+1}] \\
&\quad +g[x_0]h[x_0,...,x_{q+1}](x_{q+1} - x_0) + \sum_{i=1}^{q} g[x_0,...,x_i]h[x_i,...,x_{q+1}](x_{q+1} - x_i) \\
&= g[x_0,...,x_{q+1}](x_{q+1} - x_0)h[x_{q+1}] + \sum_{i=1}^{q} g[x_0,...,x_i](x_{q+1} - x_0)h[x_i,...,x_{q+1}] \\
&\quad +g[x_0]h[x_0,...,x_{q+1}](x_{q+1} - x_0)
\end{aligned}
$$

$$
(x_{q+1} - x_0)f[x_0, x_1, ..., x_{q+1}] = (x_{q+1} - x_0)\sum_{i=0}^{q+1} g[x_0,...,x_i]h[x_i,...,x_{q+1}]
$$

Aus Induktionsanfang und Induktionsschritt folgt nach dem Satz der vollständigen Induktion, dass Gleichung (1.12) für alle k gilt.

$\square$

Bemerkung 1.6: Die Newton'sche Interpolationsdarstellung hat gegenüber der Lagrange'schen Interpolationsdarstellung den Vorteil, dass beim Hinzunehmen einer weiteren Stützstelle die bekannten $c_0, c_1, ..., c_n$ weiterverwendet werden und nur c_{n+1} neu berechnet werden muss.

1.3.1 Newton-Interpolation mit äquidistanten Stützstellen

In den folgenden Betrachtungen wird angenommen, dass äquidistante Stützstellen vorliegen, d.h.

$$
x_{i+1} - x_i = h, \quad i = 0, 1, ..., n-1, \; h \equiv konstant. \tag{1.13}
$$

Absteigende Differenzen: Seien $x_i = x_0 + ih$, i = 0, 1, ..., n und f_i mit fester Schrittweite h gegeben.

Definition 1.2: Absteigende Differenz
Die k-te vorwärtsgenommene (absteigende) Differenz ist rekursiv definiert durch $\triangle^0 f_i := f_i$ und $\triangle^k f_i := \triangle^{k-1} f_{i+1} - \triangle^{k-1} f_i$ für $k > 0$.

Beispiel 1.4:

$$
\begin{aligned}
f[x_0, x_1] &= \frac{f_1 - f_0}{x_1 - x_2} = \frac{\triangle^0 f_1 - \triangle^0 f_0}{1h} = \frac{\triangle^1 f_0}{1!h} \\
f[x_0, x_1, x_2] &= \frac{\triangle^1 f_1 - \triangle^1 f_0}{1!h(x_2 - x_0)} = \frac{\triangle^2 f_0}{2!h}
\end{aligned}
$$

Satz 1.6: Für alle $f[x_i, x_{i+1}, ..., x_{i+k}]$ gilt folgende Bildungsvorschrift

$$
f[x_i, x_{i+1}, ..., x_{i+k}] = \frac{1}{k!}\frac{\triangle^k f_i}{h^k} \tag{1.14}
$$

Beweis 1.6:
Induktionsanfang: Gleichung (1.14) gilt für $k = 0$, denn $\triangle^0 f_i = f_i$.
Induktionsvoraussetzung: Gleichung (1.14) gilt für $k = q$.

Induktionsbehauptung: Gleichung (1.14) gilt für $k = q + 1$.
Induktionsbeweis:

$$
\begin{aligned}
f[x_i, x_{i+1}, ..., x_{i+q+1}] &= \frac{f[x_{i+1}, x_{i+2}, ..., x_{i+q+1}] - f[x_i, x_{i+1}, ..., x_{i+q}]}{x_{i+q+1} - x_i} \\
&= \frac{\frac{1}{q!h^q}\triangle^q f_{i+1} - \frac{1}{q!h^q}\triangle^q f_i}{(q+1)h} \\
&= \frac{\triangle^q f_{i+1} - \triangle^q f_i}{(q+1)!h^{q+1}} \\
&= \frac{\triangle^{q+1} f_i}{(q+1)!h^{q+1}}
\end{aligned}
$$

Aus Induktionsanfang und Induktionsschritt folgt, dass Gleichung (1.14) für alle $k \in \mathbb{N}$ gilt.

□

Durch geeignete Substitution kann man damit die Darstellung des Interpolationspolynoms vereinfachen.
Bemerkung 1.7: Die Variablentransformation mit $t = \frac{x-x_0}{h}$ führt zur folgenden Darstellung:

$$p_n(t) = \frac{\triangle^0 f_0}{0!} + \frac{\triangle^1 f_0}{1!}t + ... + \frac{\triangle^n f_0}{n!}t(t-1) \cdot ... \cdot (t-(n-1))$$

Aufsteigende Differenzen: Sei $h = x_{i+1} - x_i$ für i = 0, 1, ..., n - 1, d.h. $x_i = x_0 + ih$ gilt für alle i = 0, 1, ..., n.

Definition 1.3: Aufsteigende Differenz
Die k-te rückwärtsgenommene (aufsteigende) Differenz ist rekursiv definiert durch $\nabla^0 f_{n-i} := f_{n-i}$ und $\nabla^k f_{n-i} := \nabla^{k-1} f_{n-i} - \nabla^{k-1} f_{n-i-1}$ für $k > 0$.

Beispiel 1.5:

$$
\begin{aligned}
f[x_{n-1}, x_n] &= \frac{f_n - f_{n-1}}{x_n - x_{n-1}} = \frac{\nabla^0 f_n - \nabla^0 f_{n-1}}{h} = \frac{\nabla^1 f_n}{h} \\
f[x_{n-2}, x_{n-1}, x_n] &= \frac{f[x_{n-1}, x_n] - f[x_{n-2}, x_{n-1}]}{x_n - x_{n-2}} = \frac{\frac{1}{h}\nabla^1 f_n - \frac{1}{h}\nabla^1 f_{n-1}}{2h} = \frac{\nabla^2 f_n}{2!h}
\end{aligned}
$$

Satz 1.7: Die Gleichung

$$f[x_{n-i-k}, ..., x_{n-i-0}] = \frac{\nabla^k f_{n-i}}{k!h^k} \tag{1.15}$$

gilt für alle $k \in \mathbb{N}$.

Beweis 1.7: Der Beweis für die Gleichung (1.15) läuft analog zum Beweis der Gleichung (1.14).

Mit Hilfe der Bildungsvorschrift für aufsteigende Differenzen und einer Substitution lässt sich die Formel des Interpolationspolynoms vereinfachen.

Bemerkung 1.8: Die Variablentransformation mit $t = \frac{x-x_n}{h}$ führt zur folgenden Darstellung:

$$p_n(t) = \frac{\nabla^0 f_n}{0!} + \frac{\nabla^1 f_n}{1!}t + ... + \frac{\nabla^n f_n}{n!}t(t+1) \cdot ... \cdot (t+(n-1))$$

1.4 Hermite-Interpolation

In manchen Fällen hat man nicht nur die Funktionswerte der Stützstellen gegeben, sondern auch die k-ten Ableitungn der Funktion an den Stützstellen.
Gegeben: $(x_i, f_i^{(k)})$ mit i = 0, 1, ..., m und $k = 0, 1, ..., n_i - 1$. Es gilt $\sum_{i=0}^{m} n_i = n + 1$.
Die Koeffizienten des Polynoms, das die Interpolationsbedingung $p_n^{(k)}(x_i) = f^{(k)}(x_i)$ erfüllt, können mit Hilfe eines Gleichungssystems bestimmt werden.

Beispiel 1.6: *Gegeben* sind zwei Funktionswerte und zwei Ableitungswerte einer Funktion: $f(2) = 3, f'(2) = 4, f''(2) = 8, f(-1) = 0$.
Gesucht ist ein Polynom, das die Interpolationsbedingung erfüllt.
Lösung: Ansatz: $p_3(x) = a_0 + a_1 x + a_2 x^2 + a_3 - x^3$

$$\begin{pmatrix} 1 & 2 & 4 & 8 \\ 0 & 1 & 4 & 12 \\ 0 & 0 & 2 & 12 \\ 1 & -1 & 1 & -1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \\ 8 \\ 0 \end{pmatrix}$$

Das gesuchte Polynom, das die Interpolationsforderung erfüllt, ist $p_3(x) = 3 - 2x^2 + x^3$.

Der einfachste Fall der Hermite'schen Interpolationsaufgabe ist gegeben, wenn von m paarweise verschiedenen Stützstellen x_i die Funktionswerte f_i und die Ableitungen f_i' mit i = 0, 1, ..., m bekannt sind. Dann ist $p(x)$ ein Polynom $(2m + 1)$-ten Grades.
Das Interpolationspolynom kann mittels folgenden Ansatzes bestimmt werden:
$p(x) = \sum_{j=0}^{m} (f_j g_j(x) + f_j' h_j(x))$ mit den Polynomen $(2m + 1)$-ten Grades $g_j(x)$ und $h_j(x)$, j = 0, 1, ..., m, so dass gilt

$$g_j(x_i) = \delta_{ij},$$
$$g_j'(x_i) = 0,$$
$$h_j(x_i) = 0,$$
$$h_j'(x_i) = \delta_{ij}.$$

Dabei ist δ_{ij} das Kronecker-Symbol:

$$\delta_{ij} = \begin{cases} 0, & \text{falls } i \neq j \\ 1, & \text{falls } i = j \end{cases}$$

Es sei $h_j(x) = L_j^2(x)(x - x_j)$ mit den Lagrange'schen Multiplikator $L_j(x) := \prod_{k=0;k\neq j}^{m} \frac{x - x_k}{x_j - x_k}$.
Für $i \neq j$ gilt $L_j(x_i) = \prod_{k=0;k\neq j}^{m} \frac{x_i - x_k}{x_j - x_k} = \frac{x_i - x_i}{x_j - x_i} \prod_{k=0;k\neq i,j}^{m} \frac{x_i - x_k}{x_j - x_k} = 0$.
Für $i = j$ gilt $L_j(x_i) = L_j(x_j) = \prod_{k=0;k\neq j}^{m} \frac{x_j - x_k}{x_j - x_k} = 1$.
Folgender Ansatz wird für $g_j(x)$ gewählt: $g_j(x) = L_j^2(x)(c_j x + d_j)$. Dann gilt

$$g_j(x_i) = \begin{cases} L_j^2(x_i)(c_j x_i + d_j) = 0, & \text{falls } i \neq j \\ L_j^2(x_j)(c_j x_j + d_j) = c_j x_j + d_j, & \text{falls } i = j \end{cases}$$

$$\Rightarrow c_j x_j + d_j = 1 \tag{1.16}$$

$$\begin{aligned} g_j'(x_i) &= (L_j^2(x_i))'(c_j x_i + d_j) + L_j^2(x_i)(c_j x_i + d_j)' \\ &= 2L_j(x_i)L_j'(x_i)(c_j x_i + d_j) + L_j^2(x_i)c_j \\ &= L_j(x_i)(2L_j'(x_i)(c_j x_i + dj) + L_j(x_i)c_j) \end{aligned}$$

$$g_j'(x_i) = \begin{cases} L_j(x_i)(2L_j'(x_i)(c_jx_i + d_j) + L_j(x_i)c_j) = 0, & \text{falls } i \neq j \\ 2L_j'(x_j)(c_jx_j + dj) + L_j(x_j)c_j, & \text{falls } i = j \end{cases}$$

$$2L_j'(x_j)(c_jx_j + d_j) + c_j = 0 \tag{1.17}$$

Aus (1.16) folgt

$$d_j = 1 - c_jx_j. \tag{1.18}$$

Aus (1.17) und (1.18) folgt nun $2L_j'(x_j)(c_jx_j + 1 - c_jx_j) + c_j = 0$.

$$2L_j'(x_j) + c_j = 0$$
$$\Rightarrow c_j = -2L_j'(x_j)$$
$$\Rightarrow d_j = 1 + 2L_j'(x_j)x_j$$

Satz 1.8: Das Polynom $p(x) = \sum_{j=0}^{m} (f_jg_j(x) + f_j'h_j(x))$ erfüllt die Interpolationsforderung.
Beweis 1.8:
Behauptung:
a) $p(x_q) = f(x_q)$, q = 0, 1, ..., m
b) $p'(x_q) = f'(x_q)$, q = 0, 1, ..., m
Beweis:

$$\begin{aligned}
a) \quad p(x_q) &= \sum_{j=0}^{m} (f_jg_j(x_q) + f_j'h_j(x_q)) \\
&= f_qg_q(x_q) + f_q'h_q(x_q) + \sum_{j=0;j\neq q}^{m} (f_jg_j(x_q) + f_j'h_j(x_q)) \\
&= f_qL_q^2(x_q)(c_qx_q + d_q) + f_q'L_q^2(x_q)(x_q - x_q) \\
&\quad + \sum_{j=0;j\neq q}^{m} (f_jL_j^2(x_q)(c_jx_q + d_j) + f_j'L_j^2(x_q)(x_q - x_j)) \\
&= f_q(c_qx_q + (1 - c_qx_q)) + 0 \\
&= f_q \\
p(x_q) &= f(x_q) \\
b) \quad p'(x_q) &= \sum_{j=0}^{m} (f_jg_j'(x) + f_j'h_j'(x)) \\
&= f_qg_q'(x_q) + f_q'h_q'(x_q) + \sum_{j=0}^{m} (f_jg_j'(x_q) + f_j'h_j'(x_q)) \\
&= f_q(L_q^2(x_q)(c_qx_q + d_q))' + f_q'(L_q^2(x_q)(x_q - x_q))' \\
&\quad + \sum_{j=0}^{m} (f_j(2L_j(x_q)L_j'(x_q)(c_jx_q + d_j) + L_j^2(x_q)c_j) + f_j'h_j'(x_q)) \\
&= f_q(L_q^2(x_q)(c_qx_q + d_q))' + f_q'(L_q^2(x_q)(x_q - x_q))' \\
&\quad + \sum_{j=0}^{m} 0f_j + f_j'(2L_j(x_q)L_j'(x_q)(x_q - x_j) + L_j^2(x_q))) \\
&= f_q(L_q^2(x_q)(c_qx_q + d_q))' + f_q'(L_q^2(x_q)(x_q - x_q))' + \sum_{j=0}^{m} 0f_j'
\end{aligned}$$

$$
\begin{aligned}
&= f_q(2L_q(x_q)L_q'(x_q)(c_qx_q+d_q)+L_q^2(x_q)c_q)+f_q'((L_q^2(x_q))'(x_q-x_q)+L_q^2(x_q)) \\
&= f_q(2L_q'(x_q)\cdot 1-2L_q'(x_q))+f_q'(2L_q'(x_q)(x_q-x_q)+1) \\
&= f_q' \\
p'(x_q) &= f'(x_q)
\end{aligned}
$$

Wegen der Gültigkeit der Behauptungen a) und b) ist die Interpolationsforderung erfüllt.

□

1.5 Inverse Interpolation

Häufig sind in den Anwendungen nicht Näherungen für Funktionswerte einer gewissen Funktion f, sondern für deren Umkehrfunktion f^{-1} gesucht. Falls f eine eindeutig umkehrbare reelle Funktion ist und eine Nullstelle $x = x_0$ besitzt, so kann man x_0 mit Hilfe der Umkehrsfunktion (auch: inverse Funktion) bestimmen: $x_0 = f^{-1}(0)$.
Eine Tabelle von Funktionswerten ist gleichzeitig eine Tabelle von Werten der Umkehrfunktion, wenn man Stützstelle und Stützwert vertauscht.

Beispiel 1.7: Gegeben sei folgende Tabelle für die Langevin-Funktion $f(x) = \frac{1}{x} - \ln(x)$

x	0.7	0.8	0.9	1.0	1.1	1.2	1.3
$f(x)$	1.7852	1.4731	1.2165	1.0	0.81378	0.65102	0.50687

Gesucht ist der Wert von $f^{-1}(y)$ für $y = 1.05$, wenn eine kubische Interpolation angewendet wird. Wählt man die Werte 1.4731, 1.2165, 1.0 und 0.81378 als Stützstellen aus, so erhält man z.B. mit Hilfe der Lagrange-Interpolation oder der Newton-Interpolation den Näherungswert $f^{-1} \approx 0.9753$.

Inverse quadratische Interpolation:
Die sogenannte inverse quadratische Interpolation ist ein Verfahren zur genäherten Lösung der Gleichung $f(x) = 0$.
Die Umkehrfunktion f^{-1} von f wird in drei Stützpunkten $(y_{k-2}, x_{k-2}), (y_{k-1}, x_{k-1})$ und (y_k, x_k) interpoliert und der Funktionswert des so erhaltenen Polynoms an der Stelle $y = 0$ als neuer Näherungswert x_{k+1} verwendet wird.
Ansatz: $p_2(y) = a_0 + a_1y + a_2y^2$

$$
\begin{aligned}
a_0 + a_1y_{k-2} + a_2y_{k-2}^2 &= x_{k-2} \\
a_0 + a_1y_{k-1} + a_2y_{k-1}^2 &= x_{k-1} \\
a_0 + a_1y_k + a_2y_k^2 &= x_k
\end{aligned}
$$

$$
\begin{aligned}
a_1(y_{k-2}-y_{k-1}) + a_2(y_{k-2}^2-y_{k-1}^2) &= x_{k-2}-x_{k-1} \\
a_1(y_{k-2}-y_k) + a_2(y_{k-2}^2-y_k^2) &= x_{k-2}-x_k
\end{aligned}
$$

$$
\begin{aligned}
a_1 &= \frac{(y_{k-2}^2-y_k^2)(x_{k-2}-x_{k-1})-(y_{k-2}^2-y_{k-1}^2)(x_{k-2}-x_k)}{(y_{k-2}^2-y_k^2)(y_{k-2}-y_{k-1})-(y_{k-2}^2-y_{k-1}^2)(y_{k-2}-y_k)} \\
a_2 &= \frac{(y_{k-2}-y_k)(x_{k-2}-x_{k-1})-(y_{k-2}-y_{k-1})(x_{k-2}-x_k)}{(y_{k-2}^2-y_k^2)(y_{k-2}-y_{k-1})-(y_{k-2}^2-y_{k-1}^2)(y_{k-2}-y_k)} \\
a_0 &= x_{k-2}-a_1y_{k-2}-a_2y_{k-2}^2 = x_{k+1}
\end{aligned}
$$

Nun wird das oben beschriebene Verfahren auf die drei Stützpunkte $(y_{k-1}, x_{k-1}), (y_k, x_k)$ und (y_{k+1}, x_{k+1}) angewendet. Dieses Verfahren kann bis zur der gewünschten Genauigkeit wiederholt werden.

1.6 Interpolationsfehler

Manchmal ist eine Funktion $f(x)$ bekannt, aber sehr schwer auszuwerten. Dann besteht die Möglichkeit $(n+1)$ geeignete Stützstellen zu wählen und ein Polynom n-ten Grades durch die Stützpunkte zu legen. Nun stellt sich aber die Frage, wie gut approximiert $p_n(x)$ die Funktion $f(x)$ an Zwischenstellen $x \neq x_i$.
O.B.d.A. gilt $a \leq x_0 < x_1 < x_2 < \ldots < x_{n-1} < x_n \leq b$. Sei $f(x) = p_n(x) + R_{n+1}(x)$, dann stellt sich die Frage nach dem Wert des Restgliedes $R_{n+1}(x) = f(x) - p_n(x)$.

Satz 1.9: Sei p_n ein Polynom n-ten Grades, das Interpolationspolynom einer Funktion $f \in C^{n+1}[a,b]$ zu den paarweise verschiedenen Knoten $a \leq x_0 < x_1 < \ldots < x_n \leq b$, dann gibt es für jedes $x \in [a,b]$ ein $\xi \in (\min\{x_0, x\}, \max\{x_n, x\})$ mit

$$R_{n+1}(x) = f(x) - p_n(x) = \frac{f_{(n+1)}(\xi)}{(n+1)!} \prod_{k=0}^{n} (x - x_k). \tag{1.19}$$

Bemerkung 1.9: $C^q[a,b]$ ist die Menge aller q-mal stetig differenzierbaren Funktion auf dem Intervall $[a,b]$. Eine Funktion ist stetig differenzierbar, wenn ihre erste Ableitung existiert und stetig ist.

Beweis 1.9:
Fall 1: $x = x_i$
$p_n(x_i) = f(x_i) \Rightarrow R_{n+1}(x_i) = 0$
und $\frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{k=0}^{n} (x_i - x_k) = 0$.
Fall 2: $x \neq x_i$ und x ist fest
Es werden folgende Funktionen definiert:

$$w_n(t) := (t - x_0)(t - x_1) \cdot \ldots \cdot (t - x_n) \tag{1.20}$$

$$g(t) := f(t) - p_n(t) - \frac{f(x) - p_n(x)}{w_n(x)} w_n(t) \tag{1.21}$$

Es gilt $g(t) \in C^{n+1}[a,b]$, weil $f(t) \in C^{n+1}[a,b]$ nach Voraussetzung und $p_n(t)$ und $w_n(t)$ Polynome sind. Die Funktion $g(t)$ besitzt $(n+2)$ verschiedene Nullstellen in $[a,b]$. Diese Nullstellen sind $x_0, x_1, \ldots, x_n, x$. Demzufolge besitzt $g^{(n+1)}(t)$ mindestens eine Nullstelle ξ mit $\xi \in (\min\{x_0, x\}, \max\{x_n, x\})$. Aus Gleichung (1.21) folgt:

$$0 = g^{(n+1)}(\xi) = f^{(n+1)}(\xi) - p_n(\xi) - \frac{(f(x) - p_n(x)) w_n^{(n+1)}(\xi)}{w_n(x)}$$

Daraus folgt wegen $p_n^{(n+1)}(\xi) = 0$:

$$f^{(n+1)}(\xi) = \frac{R_{n+1}(x)(n+1)!}{w_n(x)}$$

Umstellen nach $R_{n+1}(x)$ liefert:

$$R_{n+1}(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{k=0}^{n} (x - x_k)$$

Da es keinen weiteren Fall gibt, ist Satz 1.9 beweisen.

□

Die Funktion $f^{(n+1)}(x)$ kann "groß" werden im Vergleich zu der ursprünglichen Funktion $f(x)$, so dass $p_n(x)$ an Zwischenstellen stark oszilliert.

Kapitel 2

Spline-Interpolation

Die Grundidee der sogenannten Spline-Interpolation besteht darin, Polynome geringen Grades durch je zwei benachbarte Stützstellen zu legen, um so Zwischenstellen leicht interpolierten zu können.
O.B.d.A. gilt $x_0 < x_1 < x_2 < \ldots < x_{n-1} < x_n$.

Wenn man linear interpoliert, erhält man folgenden Spline $s(x)$:

$$s(x) = \begin{cases} \frac{f_1 - f_0}{x_1 - x_0}(x - x_0) + f_0 & \text{für } x \in [x_0, x_1] \\ \frac{f_2 - f_1}{x_2 - x_1}(x - x_1) + f_0 & \text{für } x \in [x_1, x_2] \\ \frac{f_3 - f_2}{x_3 - x_2}(x - x_2) + f_0 & \text{für } x \in [x_2, x_3] \\ \vdots & \\ \frac{f_n - f_{n-1}}{x_n - x_{n-1}}(x - x_{n-1}) + f_{n-1} & \text{für } x \in [x_{n-1}, x_n] \end{cases}$$

Definition 2.1: Spline
Sei $\triangle_n := a = x_0 < x_1 < x_2 < \ldots < x_{n-1} < x_n = b$ ein Gitter von $n + 1$ paarweise verschiedenen Knoten.
Ein Spline vom Grad m (oder der Ordnung $m + 1$) bezüglich $\triangle_n$ ist eine Funktion $s \in C^{m-1}[a, b]$, die auf jedem Intervall $[x_i, x_{i+1}]$ mit einen Polynom s_i (vom Grad m) übereinstimmt.
Der Raum der Splines vom Grad m wird mit $S_m(\triangle_n)$ bezeichnet.

Definition 2.2: Interpolierender Spline
Ist f eine auf [a,b] gegebene reellwertige Funktion bzw. $f_0, f_1, \ldots, f_n \in \mathbb{R}$ gegebene Stützwerte, so heißt $s_f \in S_m(\triangle_n)$ ein f bzw. $f_0, f_1, \ldots, f_n$ interpolierender Spline, falls $s_f(x_i) = f(x_i)$ bzw. $s_f(x_i) = f_i$, i = 0, 1, ..., n, gilt.

2.1 Kubische Spline-Interpolation

Das menschliche Auge nimmt keine Unstetigkeiten ("Spitzen") in der Krümmung, d. h. der zweiten Ableitung, wahr. Also werden zweimal stetig differenzierbare, kubische Splines als "glatt" wahrgenommen.
$\Rightarrow s_i$ soll ein Polynom dritten Grades sein.
O. B. d. A. gilt $x_0 < x_1 < x_2 < \ldots < x_{n-1} < x_n$.

Ansatz der Interpolation durch kubische Splines:
Im Intervall $[x_i, x_{i+1}]$ mit i = 0, 1, ..., n - 1 soll

$$s_i(x) = a_i + b_i(x - x_i) + \frac{c_i}{2}(x - x_i)^2 + \frac{d_i}{6}(x - x_i)^3 \tag{2.1}$$

gelten.

Der Spline s wird also durch n Splines s_i, i = 0, 1, ..., n - 1, mit je vier Koeffizienten (a_i, b_i, c_i, d_i) charakterisiert, d.h. es müssen $4n$ Koeffizienten bestimmt werden.
Zusätzlich zu der Interpolationsbedingung $s(x_i) = f_i$ werden noch folgende Verheftungsbedingungen gefordert:

$$\left.\begin{array}{l} s_i(x_{i+1}) = s_{i+1}(x_{i+1}) \\ s_i'(x_{i+1}) = s_{i+1}'(x_{i+1}) \\ s_i''(x_{i+1}) = s_{i+1}''(x_{i+1}) \end{array}\right\} i = 0, 1, \ldots, n-2$$

Aus der Interpolationsbedingung und den Verheftungsbedingungen ergeben sich $(n+1) + 3(n-1) = 4n - 2$ Gleichungen. Also werden zwei zusätzliche Bedingungen gebraucht.
Es können folgende Fälle vorliegen:

- natürliche Randbedingungen:
  $s''(x_0) = 0$
  $s''(x_n) = 0$
- Hermite-Randbedingungen:
  $s'(x_0) = f_0'$
  $s'(x_n) = f_n'$
- periodische Randbedingungen:
  $s'(x_0) = s'(x_n)$
  $s''(x_0) = s''(x_n)$

Wegen $s \in C^2[a, b]$ folgt aus Gleichung (2.1):

$$s_i'(x) = b_i + c_i(x - x_i) + \frac{d_i}{2}(x - x_i)^2, \quad i = 0, 1, \ldots, n-1 \tag{2.2}$$

$$s_i''(x) = c_i + d_i(x - x_i), \quad i = 0, 1, \ldots, n-1 \tag{2.3}$$

Für $x = x_i$ folgt nun aus (2.1), (2.2) und (2.3):

$$s_i(x_i) = a_i + b_i(x_i - x_i) + \frac{c_i}{2}(x_i - x_i)^2 + \frac{d_i}{6}(x_i - x_i)^3 = a_i$$

$$\Rightarrow a_i = f_i \tag{2.4}$$

$$s_i'(x_i) = b_i + c_i(x_i - x_i) + \frac{d_i}{2}(x_i - x_i)^2$$

$$\Rightarrow s_i'(x_i) = b_i \tag{2.5}$$

$$s_i''(x_i) = c_i + d_i(x_i - x_i)$$

$$\Rightarrow s_i''(x_i) = c_i \tag{2.6}$$

In den nun folgenden Betrachtungen werden die Abstände zwischen den Stützstellen als äquidistant angenommen, damit die Umformungen kompakter dargestellt werden können. Also existiert eine konstante Schrittweite h:

$$h := x_{i+1} - x_i$$

Für $x = x_{i+1}$ folgt nun aus (2.1), (2.2) und (2.3):

$$s_i(x_{i+1}) = a_i + b_i(x_{i+1} - x_i) + \frac{c_i}{2}(x_{i+1} - x_i)^2 + \frac{d_i}{6}(x_{i+1} - x_i)^3$$

$$\Rightarrow s_i(x_{i+1}) = a_i + b_i h + \frac{c_i}{2}h^2 + \frac{d_i}{6}h^3 \tag{2.7}$$

$$s_i'(x_{i+1}) = b_i + c_i(x_{i+1} - x_i) + \frac{d_i}{2}(x_{i+1} - x_i)^2$$

$$\Rightarrow s_i'(x_{i+1}) = b_i + c_i h + \frac{d_i}{2}h^2 \tag{2.8}$$

$$s_i''(x_{i+1}) = c_i + 1 + d_i(x_i - x_i)$$

$$\Rightarrow s_i''(x_{i+1}) = c_i + d_i h \tag{2.9}$$

Nun wird versucht die Koeffizienten a_i, b_i und d_i mit i = 0, 1, ..., n - 1 als Funktion von $c_0, c_1, \ldots, c_{n-1}$ und c_n darzustellen. Dabei ist es unwichtig, dass c_n in keinem Spline s_i als Koeffizient verwendet wird, wenn c_n trotzdem eindeutig bestimmbar ist.
Alle a_i sind durch Gleichung (2.4) eindeutig als Funktion von c_k, k = 0, 1, ..., n, bestimmt. So dass nur noch b_i und d_i als Funktion von c_k dargestellt werden müssen.

Aus $s_i''(x_{i+1}) = s_{i+1}''(x_{i+1})$ mit i = 0, 1, ..., n - 1 und den Gleichungen (2.6) und (2.9) folgt:

$$c_i + d_i h = c_{i+1}$$

$$\Rightarrow d_i = \frac{c_{i+1} - c_i}{h} \tag{2.10}$$

Aus $s_i(x_{i+1}) = s_{i+1}(x_{i+1})$ und den Gleichungen (2.4) und (2.7) folgt:

$$a_i + b_i h + \frac{c_i}{2}h^2 + \frac{d_i}{6}h^3 = a_{i+1}$$

$$\Rightarrow f_i + b_i h + \frac{c_i}{2}h^2 + \frac{d_i}{6}h^3 = f_{i+1}$$

Unter Verwendung von Gleichung (2.10) folgt daraus:

$$b_i h = (f_{i+1} - f_i) - \frac{c_i}{2}h^2 - \frac{c_{i+1} - c_i}{6}h^2$$

$$b_i = \frac{f_{i+1} - f_i}{h} - \frac{c_{i+1} + 2c_i}{6}h \tag{2.11}$$

Jetzt müssen noch die c_k mit k = 0, 1, ..., n bestimmt werden. Aus $s_i'(x_{i+1}) = s_{i+1}'(x_{i+1})$ mit i = 0, 1, ..., n - 2 folgt:

$$b_i + c_i h + \frac{d_i}{2}h^2 = b_{i+1}$$

Wegen den Gleichungen (2.10) und (2.11) folgt daraus:

$$\frac{f_{i+1}-f_i}{h} - \frac{h(c_{i+1}+2c_i)}{6} + c_i h + \frac{h(c_{i+1}-c_i)}{2} = \frac{f_{i+2}-f_{i+1}}{h} - \frac{h(c_{i+2}+2c_{i+1})}{6}$$

$$\Rightarrow c_i + 4c_{i+1} + c_{i+2} = 6\frac{f_i - 2f_{i+1} + f_{i+2}}{h^2} \tag{2.12}$$

Natürliche Randbedingungen:
Die natürlichen Randbedingungen einer Spline-Funktion sind:

$$\begin{aligned} s''(x_0) &= 0 \\ s''(x_n) &= 0 \end{aligned}$$

Wegen Gleichung (2.6) ist $c_0 = s_0''(x_0)$.

$$\Rightarrow c_0 = 0 \tag{2.13}$$

Aus $s_i''(x_{i+1}) = s_{i+1}''(x_{i+1})$ mit i = 0, 1, ..., n - 1 und Gleichungen (2.6) folgt $s_{n-1}''(x_n) = c_n$.

$$\Rightarrow c_n = 0 \tag{2.14}$$

Aus den Gleichungen (2.12), (2.13) und (2.14) ergibt sich folgendes Gleichungssytem:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-1} \\ c_n \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{6(f_0-2f_1+f_2)}{h^2} \\ \frac{6(f_1-2f_2+f_3)}{h^2} \\ \vdots \\ \frac{6(f_{n-2}-2f_{n-1}+f_n)}{h^2} \\ 0 \end{pmatrix} \tag{2.15}$$

Hermite'sche Randbedingungen:
Die Hermite'schen Randbedingungen einer Spline-Funktion sind:

$$\begin{aligned} s'(x_0) &= f_0' \\ s'(x_n) &= f_n' \end{aligned}$$

Aus Gleichung (2.11) folgt

$$c_1 + 2c_0 = \frac{6(f_1 - f_0 + f_0'h)}{h^2} \tag{2.16}$$

Aus $s_{n-1}'(x_n) = f_n'$ folgt wegen Gleichung (2.8)

$$b_{n-1} + c_{n-1}h + \frac{d_{n-1}^2}{2} = f_n'$$

$$\Rightarrow \frac{f_n - f_{n-1}}{h} - \frac{(c_n + 2c_{n-1})h}{6} + \frac{6c_{n-1}h}{6} + \frac{(3c_n - 3c_{n-1})h}{6} = f_n'$$

$$\Rightarrow c_{n-1} + 2c_n = \frac{6(f_{n-1} - f_n + f_n'h)}{h^2} \tag{2.17}$$

Aus den Gleichungen (2.12), (2.16) und (2.17) ergibt sich folgendes Gleichungssytem:

$$\begin{pmatrix} 2 & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-1} \\ c_n \end{pmatrix} = \begin{pmatrix} \frac{6(f_1-f_0+f_0'h)}{h^2} \\ \frac{6(f_0-2f_1+f_2)}{h^2} \\ \frac{6(f_1-2f_2+f_3)}{h^2} \\ \vdots \\ \frac{6(f_{n-2}-2f_{n-1}+f_n)}{h^2} \\ \frac{6(f_{n-1}-f_n+f_n'h)}{h^2} \end{pmatrix} \tag{2.18}$$

Periodische Randbedingungen:
Die periodischen Randbedingungen einer Spline-Funktion sind:

$$\begin{aligned} s'(x_0) &= s'(x_n) \\ s''(x_0) &= s''(x_n) \end{aligned}$$

Aus $s'(x_0) = s'(x_n)$ folgt wegen den Gleichungen (2.5) und (2.8)

$$b_{n-1}h = b_0h - c_{n-1}h^2 - \frac{d_{n-1}}{2}h^3$$

$$\Rightarrow b_{n-1}h = b_0h - \frac{c_{n-1}+c_n}{2}h^2 \tag{2.19}$$

Aus $s''(x_0) = s''(x_n)$ folgt wegen den Gleichungen (2.9) und (2.10)

$$c_0 = c_{n-1} + d_{n-1}h$$

$$\Rightarrow c_0 = c_n \tag{2.20}$$

Wegen Gleichung (2.7) gilt

$$a_{n-1} + b_{n-1}h + \frac{c_{n-1}}{2}h^2 + \frac{d_{n-1}}{6}h^3 = f_n$$

Daraus folgt wegen den Gleichungen (2.4), (2.10), (2.11), (2.19) und (2.20)

$$f_{n-1} + b_{n-1}h + \frac{c_{n-1}}{2}h^2 + \frac{d_{n-1}}{6}h^3 = f_n$$

$$\Rightarrow b_0h - \frac{c_{n-1}+c_n}{2}h^2 + \frac{c_{n-1}}{2}h^2 + \frac{c_n - c_{n-1}}{6}h^2 = f_n - f_{n-1}$$

$$\Rightarrow (f_1 - f_0) - \frac{2c_0 + c_1}{6}h^2 - \frac{3c_n}{6}h^2 + \frac{c_n - c_{n-1}}{6}h^2 = f_n - f_{n-1}$$

$$\Rightarrow -(2c_0 + c_1 + c_{n-1} + 2c_n)h^2 = 6(f_n - f_{n-1}) - 6(f_1 - f_0)$$

$$4c_0 + c_1 + c_{n-1} = \frac{6(f_1 - f_0) - 6(f_n - f_{n-1})}{h^2} \tag{2.21}$$

Aus den Gleichungen (2.12), (2.20) und (2.21) ergibt sich folgendes Gleichungssytem:

$$\begin{pmatrix} 4 & 1 & 0 & 0 & \dots & 0 & 1 & 0 \\ 1 & 4 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & 4 & 1 \\ 1 & 0 & 0 & 0 & \dots & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-1} \\ c_n \end{pmatrix} = \begin{pmatrix} \frac{6(f_1-f_0)-6(f_n-f_{n-1})}{h^2} \\ \frac{6(f_0-2f_1+f_2)}{h^2} \\ \frac{6(f_1-2f_2+f_3)}{h^2} \\ \vdots \\ \frac{6(f_{n-2}-2f_{n-1}+f_n)}{h^2} \\ 0 \end{pmatrix} \tag{2.22}$$

Anhang A

Ableitungsberechnung

Hinweis:

$$\binom{n}{k} + \binom{n}{k+1} = \binom{n+1}{k+1}, \text{denn}$$

$$\frac{n(n-1)\cdot ...\cdot(n-k+1)}{k!} + \frac{n(n-1)\cdot ...\cdot(n-k)}{(k+1)!} = \frac{n(n-1)\cdot ...\cdot(n-k)}{k!}\,\frac{(k+1)+(n-k+1)}{k+1}$$

Es wird $x^0 := 1 \quad \forall x$ festgelegt.
Beziehungen im Horner-Schema:

$$\begin{array}{cccccc}
a_n & a_{n-1} & ... & a_2 & a_1 & a_0 \\
b_{0,n} & b_{0,n-1} & ... & b_{0,2} & b_{0,1} & b_{0,0} \\
b_{1,n} & b_{1,n-1} & ... & b_{1,2} & b_{1,1} & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \\
b_{n-2,n} & b_{n-2,n-1} & b_{n-2,n-2} & & & \\
b_{n-1,n} & b_{n-1,n-1} & & & & \\
b_{n,n} & & & & &
\end{array}$$

$$b_{i,n-k} = xb_{i,n-k+1} + b_{i-1,n-k}, \qquad k > 0, i > -1$$

$$b_{i,n-k} = b_{i-1,n-k} = a_{n-k}, \qquad k = 0, i > -1$$

$$b_{i,n-k} = xb_{i,n-k+1} + an - k, \qquad k > -1, i = 0$$

Behauptung:

$$b_{i,n-k} = \sum_{q=0}^{k} \binom{i+q}{i} a_{n-k+q} x^q \tag{A.1}$$

Induktionsanfang:
Gleichung (A.1) gilt für $k = 0$, denn $b_{i,n-0} = a_n$ und $\sum_{q=0}^{0} \binom{i+q}{i} a_{n+q} x^q = a_n$.
Gleichung (A.1) gilt für $i = 0$, denn $b_{0,n-k} = \sum_{q=0}^{k} a_{n-k+q} x^q$ und $\sum_{q=0}^{k} \binom{q}{0} a_{n-k+q} x^q = \sum_{q=0}^{k} a_{n-k+q} x^q$.
Induktionsvoraussetzungen: Gleichung (A.1) gilt für $k < s + 1$ oder $i < t + 1$.
Induktionsbehauptungen: Gleichung (A.1) gilt für

a) $k = s + 1$
b) $i = t + 1$.

Induktionsbeweis:
a) $b_{t,n-(s+1)} = xb_{t,n-(s+1)+1} + b_{t-1,n-(s+1)}$

$$= x\sum_{q=0}^{s}\binom{t+q}{q}a_{n-s+q}x^q + \sum_{q=0}^{s+1}\binom{t-1+q}{q}a_{n-s-1+q}x^q$$

$$= \sum_{q=0}^{s}\binom{t+q}{q}a_{n-s+q}x^{q+1} + \sum_{q=1}^{s+1}\binom{t-1+q}{q}a_{n-s-1+q}x^q + \binom{t-1}{t-1}a_{n-s-1}x^0$$

$$= x\sum_{q=1}^{s+1}\binom{t+q-1}{q-1}a_{n-s+q-1}x^q + \sum_{q=1}^{s+1}\binom{t-1+q}{q}a_{n-s-1+q}x^q + \binom{t}{t}a_{n-s-1}x^0$$

$$= x\sum_{q=1}^{s+1}\left(\binom{t+q-1}{q-1}a_{n-s-1+q}x^q + \binom{t-1+q}{q}a_{n-s-1+q}x^q\right) + \binom{t}{t}a_{n-s-1}x^0$$

$$= x\sum_{q=1}^{s+1}\binom{t+q}{q}a_{n-s-1+q}x^q + \binom{t+0}{t}a_{n-s-1}x^0$$

$$= x\sum_{q=0}^{s+1}\binom{t+q}{q}a_{n-s-1+q}x^q$$

b) $b_{t+1,n-s} = xb_{t+1,n-s+1} + b_{t,n-s}$

$$= x\sum_{q=0}^{s-1}\binom{t+1+q}{t+1}a_{n-s+1+q}x^q + \sum_{q=0}^{s}\binom{t+q}{t}a_{n-s+q}x^q$$

$$= \sum_{q=1}^{s}\binom{t+q}{t+1}a_{n-s+1+q-1}x^q + \sum_{q=1}^{s}\binom{t+q}{t}a_{n-s+q}x^q + \binom{t}{t}a_{n-s}x^0$$

$$= \sum_{q=1}^{s}\left(\binom{t+q}{t+1}a_{n-s+1+q-1}x^q + \binom{t+q}{t}a_{n-s+q}x^q\right) + \binom{t+1}{t+1}a_{n-s}x^0$$

$$= \sum_{q=1}^{s}\binom{t+1+q}{t+1}a_{n-s+q}x^q + \binom{t+1}{t+1}a_{n-s}x^0$$

$$= \sum_{q=0}^{s}\binom{t+1+q}{t+1}a_{n-s+q}x^q$$

Aus Induktionsanfang und Induktionsschritt folgt, dass Gleichung (A.1) für alle $b_{i,n-k}$ gilt.

$\square$

Bezug zur Ableitung:
Aus $f'(x) = \sum_{q=0}^{n-1} a_q x^q$ folgt $f^{(i)}(x) = \sum_{q=0}^{n-i}(q+1)(q+2)\cdot\ldots\cdot(q+i)a_{q+i}x^q$. Also ist $i!b_{i,i} = f^{(i)}(x)$.