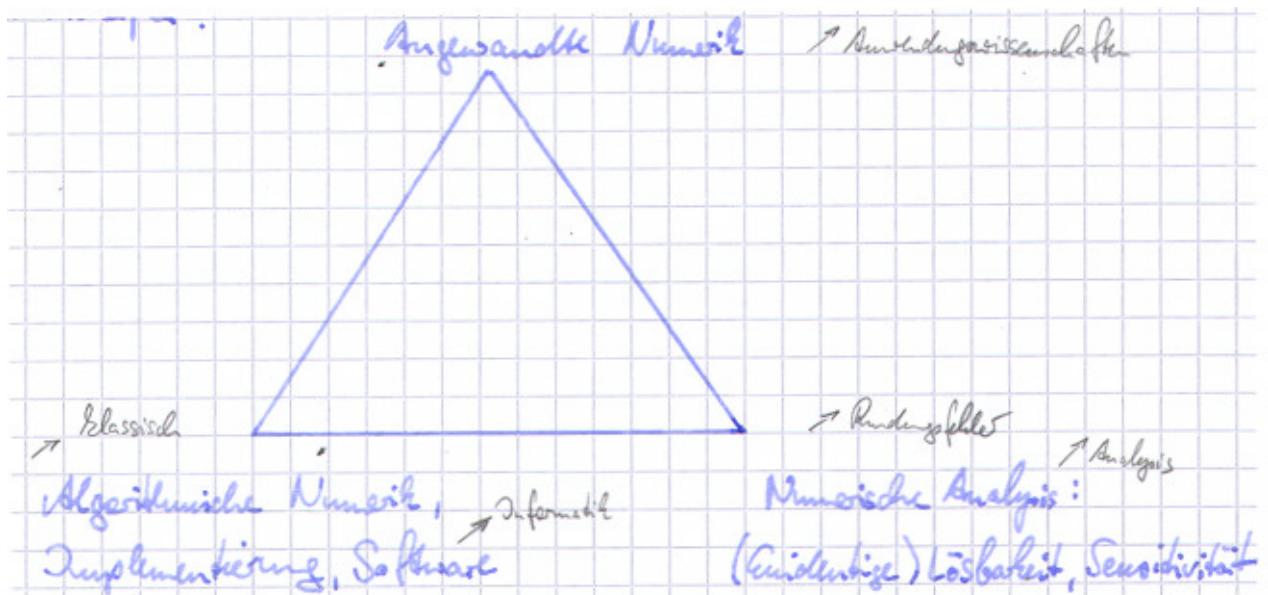


# 1 Einführung

## 1.1 Grundlagen

### Bemerkung 1.1 (Numerische Mathematik)

- a) **Im engeren Sinn:** zahlenmäßige Auswertung mathematischer Zusammenhänge  
z. B.
- Lösung von linearen und nichtlinearen Gleichungssystemen
  - Numerische Integration und Differentiation
  - Näherungsweise Auswertung reeller Funktionen
  - Numerische Lösung von Differentialgleichungen
  - Numerische Lösung von Optimierungsproblemen
- b) **Typisches Ziel:** Näherungen für die exakte Lösung eines mathematischen Problems, deren Fehler beliebig klein gemacht werden kann und für die verlässliche Fehlerschranken vorliegen.
- c) **praktisch:** wesentliche Komponente des Wissenschaftlichen Rechnens (engl.: scientific computing): Computersimulation auf der Grundlage mathematischer Modelle in den Anwendungswissenschaften: Naturwissenschaften, Ingenieurwissenschaften, Medizin, Wirtschaftswissenschaften.



### Bemerkung 1.2 (Entwicklung der Rechentechnik)

- Entwicklung der Numerik untrennbar verknüpft mit Entwicklung der Rechentechnik
- Grundlagen: verstärkt ab 18. Jahrhundert

## Bemerkung 1.2: Entwicklung der Rechentechnik

Z3 (1941)



ENIAC (1946)



Martin-Luther-Universität Halle-Wittenberg, FB Mathematik und Informatik  
Martin Arnold: Numerische Mathematik für Fachrichtung Informatik und Lehramt (WiS 2005/06)

Abbildung 1.1: Klassische Rechentechnik.

## Bemerkung 1.2: Entwicklung der Rechentechnik (II)

Hitachi SR8000-F1



Martin-Luther-Universität Halle-Wittenberg, FB Mathematik und Informatik  
Martin Arnold: Numerische Mathematik für Fachrichtung Informatik und Lehramt (WiS 2005/06)

Abbildung 1.2: Moderner Hochleistungsrechner am LRZ München.

- 1941 Z3 (K. Zuse)
- 1946 ENIAC (J. v. Neumann)
- 1958 erster Mikrochip
- 1967 erster Taschenrechner
- 1976 Home-Computer „Apple“
- 1981 erster Personal Computer (PC)
- heute: leistungsfähige Arbeitsplatzrechner (PC, Workstation)  
Vektor- und Parallelrechner für High performance computing
- seit 1971: Anzahl der elementaren Transistorfunktionen je Sekunde verdoppelt sich etwa nach jeweils 18 Monaten

### Bemerkung 1.3 (Literatur)

- [1] Th. Huckle and S. Schneider. *Numerik für Informatiker*. Springer-Verlag, Berlin, 2002.
- [2] J. Stoer. *Numerische Mathematik 1*. Springer-Verlag, Berlin Heidelberg New York, 8th edition, 1999.
- [3] J. Stoer and R. Bulirsch. *Numerische Mathematik 2*. Springer-Verlag, Berlin Heidelberg New York, 4th edition, 2000.
- [4] P. Deuffhard and A. Hohmann. *Numerische Mathematik I. Eine algorithmisch orientierte Einführung*. Walter de Gruyter, Berlin New York, 3rd edition, 2002.
- [5] A. Quarteroni, R. Sacco, and F. Saleri. *Numerische Mathematik 1*. Springer-Verlag, Berlin, 2002.
- [6] A. Quarteroni, R. Sacco, and F. Saleri. *Numerische Mathematik 2*. Springer-Verlag, Berlin, 2002.
- [7] G.H. Golub and Ch.F. van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore London, 3rd edition, 1996.

## 1.2 Klassische Polynominterpolation

### Bemerkung 1.4 (Problemstellung)

**geg.:**  $n + 1$  Stützpunkte  $(x_j, y_j)$ , ( $j = 0, 1, \dots, n$ ) mit Stützstellen  $x_j$  und Stützwerten  $y_j$ , zwischen denen ein (oft auch nur vermuteter) funktionaler Zusammenhang besteht:  $y_j = f(x_j)$ , ( $j = 0, 1, \dots, n$ ). Praktisch oft: Messdaten  $y_j$

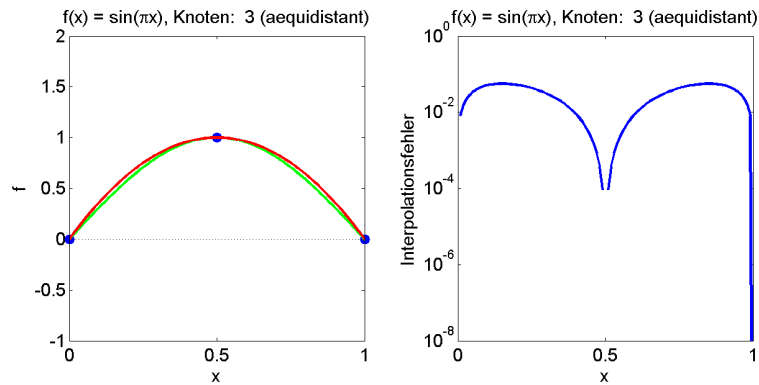
**ges.:** Polynom  $\Phi^{(n)}(x)$  höchstens  $n$ -ten Grades, das die  $n + 1$  Interpolationsbedingungen

$$y_j = \Phi^{(n)}(x_j), \quad (j = 0, 1, \dots, n)$$

erfüllt.

## Bemerkung 1.4: Polynominterpolation

**Beispiel**  $n = 2$ ,  $x_0 = 0$ ,  $x_1 = 1/2$ ,  $x_2 = 1$   
 $y_j = \sin \pi x_j$ , ( $j = 0, 1, 2$ )



Martin-Luther-Universität Halle-Wittenberg, FB Mathematik und Informatik  
 Martin Arnold: Numerische Mathematik für Fachrichtung Informatik und Lehramt (WiS 2005/06)

Abbildung 1.3: Interpolationspolynom  $\Phi^{(2)}(x)$  zu  $f(x) = \sin \pi x$ , ( $x \in [0, 1]$ ).

**Beispiel** Tabellierte Daten, z. B.

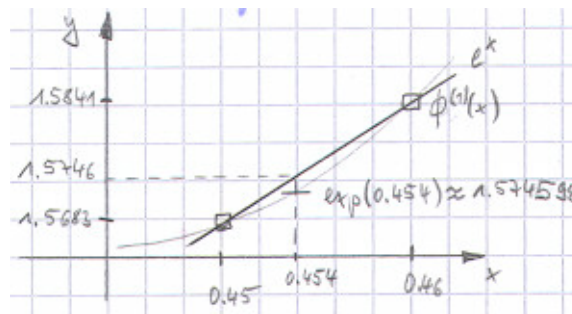
$n = 1$ ,  $f(x) = e^x$ ,  $x_0 = 0.45$ ,  $x_1 = 0.46$ ,  $y_0 = 1.5683 \approx \exp(0.45)$ ,  $y_1 = 1.5841 \approx \exp(0.46)$

### Lineare Interpolation

$$\exp(x) \approx \Phi^{(1)}(x) = 1.5683 + \frac{x - 0.45}{0.46 - 0.45}(1.5841 - 1.5683)$$

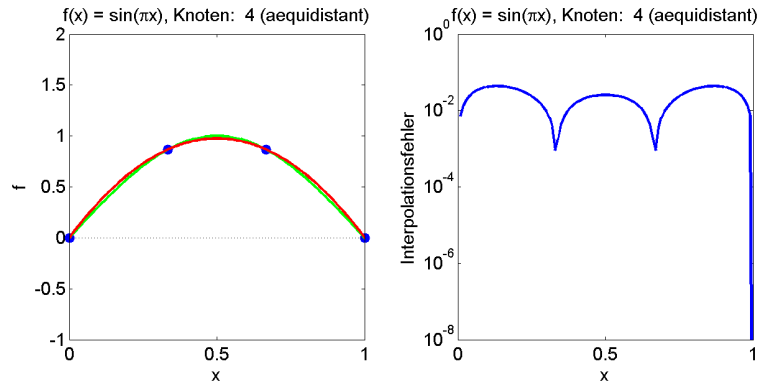
**Beispiel**

$$\Phi^{(1)}(0.454) = 1.5746$$



## Bemerkung 1.4: Polynominterpolation (II)

**Beispiel**  $n = 3$ ,  $x_0 = 0$ ,  $x_1 = 1/3$ ,  $x_2 = 2/3$ ,  $x_3 = 1$   
 $y_j = \sin \pi x_j$ , ( $j = 0, 1, 2, 3$ )

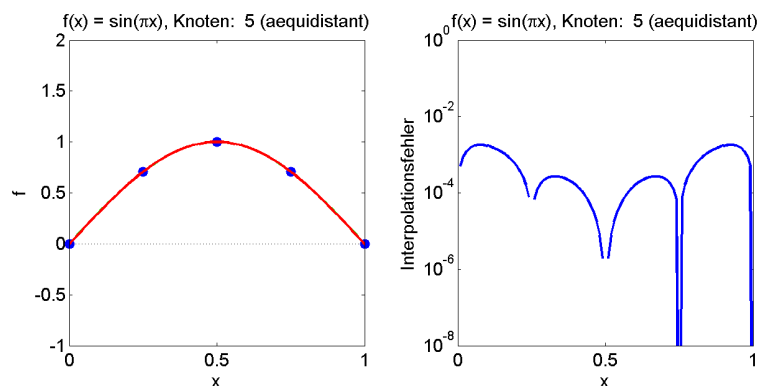


Martin-Luther-Universität Halle-Wittenberg, FB Mathematik und Informatik  
 Martin Arnold: Numerische Mathematik für Fachrichtung Informatik und Lehramt (WiS 2005/06)

Abbildung 1.4: Interpolationspolynom  $\Phi^{(3)}(x)$  zu  $f(x) = \sin \pi x$ , ( $x \in [0, 1]$ ).

## Bemerkung 1.4: Polynominterpolation (III)

**Beispiel**  $n = 4$ ,  $x_0 = 0$ ,  $x_1 = 1/4$ ,  $x_2 = 1/2$ ,  $x_3 = 3/4$ ,  $x_4 = 1$   
 $y_j = \sin \pi x_j$ , ( $j = 0, 1, 2, 3, 4$ )

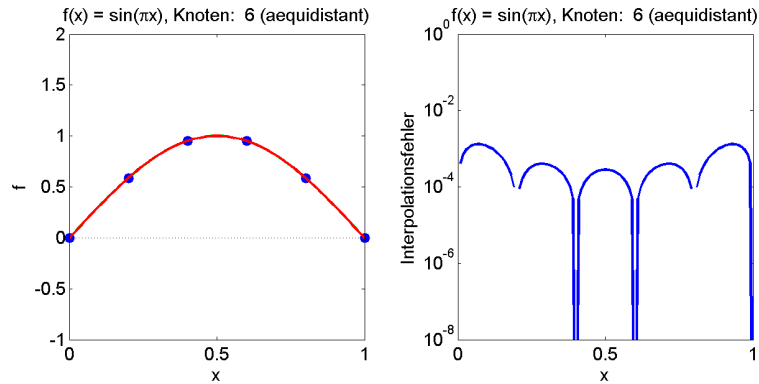


Martin-Luther-Universität Halle-Wittenberg, FB Mathematik und Informatik  
 Martin Arnold: Numerische Mathematik für Fachrichtung Informatik und Lehramt (WiS 2005/06)

Abbildung 1.5: Interpolationspolynom  $\Phi^{(4)}(x)$  zu  $f(x) = \sin \pi x$ , ( $x \in [0, 1]$ ).

### Bemerkung 1.4: Polynominterpolation (IV)

**Beispiel**  $n = 5$ ,  $x_j = j/5$ , ( $j = 0, 1, 2, 3, 4, 5$ )  
 $y_j = \sin \pi x_j$ , ( $j = 0, 1, 2, 3, 4, 5$ )

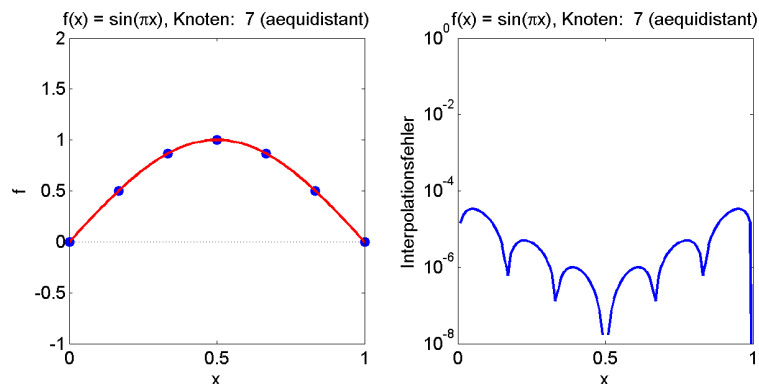


Martin-Luther-Universität Halle-Wittenberg, FB Mathematik und Informatik  
 Martin Arnold: Numerische Mathematik für Fachrichtung Informatik und Lehramt (WiS 2005/06)

Abbildung 1.6: Interpolationspolynom  $\Phi^{(5)}(x)$  zu  $f(x) = \sin \pi x$ , ( $x \in [0, 1]$ ).

### Bemerkung 1.4: Polynominterpolation (V)

**Beispiel**  $n = 6$ ,  $x_j = j/6$ , ( $j = 0, 1, 2, 3, 4, 5, 6$ )  
 $y_j = \sin \pi x_j$ , ( $j = 0, 1, 2, 3, 4, 5, 6$ )



Martin-Luther-Universität Halle-Wittenberg, FB Mathematik und Informatik  
 Martin Arnold: Numerische Mathematik für Fachrichtung Informatik und Lehramt (WiS 2005/06)

Abbildung 1.7: Interpolationspolynom  $\Phi^{(6)}(x)$  zu  $f(x) = \sin \pi x$ , ( $x \in [0, 1]$ ).

**Bemerkung 1.5 (Eindeutigkeit des Interpolationspolynoms)**

Sind die Stützstellen  $x_0, x_1, \dots, x_n$  paarweise voneinander verschieden, so ist das Interpolationspolynom  $\Phi^{(n)}(x)$  aus Bemerkung 1.4 eindeutig bestimmt, denn gilt für zwei Polynome  $\Phi_1^{(n)}, \Phi_2^{(n)} \in \Pi_n$

$$\Phi_1^{(n)}(x_j) = \Phi_2^{(n)}(x_j) = y_j, \quad (j = 0, 1, \dots, n),$$

so ist  $\Phi_1^{(n)} - \Phi_2^{(n)} \in \Pi_n$  ein Polynom mit den  $n + 1$  Nullstellen  $x_0, x_1, \dots, x_n$   
 $\Rightarrow \Phi_1^{(n)} - \Phi_2^{(n)} = 0$  (Fundamentalsatz der Algebra),  $\Phi_1^{(n)}(x) \equiv \Phi_2^{(n)}(x)$ .

**Bemerkung 1.6 (Elementarer Zugang)**

Sei  $\Phi^{(n)}(x) = \sum_{i=0}^n a_i x^i$  mit den zunächst unbekanntenen Koeffizienten  $a_0, a_1, \dots, a_n$ . Die Interpolationsbedingungen  $y_j = \Phi^{(n)}(x_j)$ , ( $j = 0, 1, \dots, n$ ) sind äquivalent zu dem linearen Gleichungssystem

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Lösung mit dem Gaußschen Algorithmus, vgl. Abschnitt 2.1.

**Ziel:** Transformation in ein äquivalentes lineares Gleichungssystem mit Dreiecksgestalt

**Schritt 1** Addiere Vielfache der 1. Zeile zu Zeilen  $2, \dots, n + 1$  so, dass in der 1. Spalte alle Elemente unterhalb der Hauptdiagonale verschwinden:

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 0 & x_1 - x_0 & x_1^2 - x_0^2 & \cdots & x_1^n - x_0^n \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & x_n - x_0 & x_n^2 - x_0^2 & \cdots & x_n^n - x_0^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 - y_0 \\ \vdots \\ y_n - y_0 \end{pmatrix}$$

Rechenaufwand:  $\approx n^2$  Rechenoperationen zur Transformation von  $n$  Zeilen mit je  $n$  Spalten

**Schritt  $k$**  Addiere Vielfache der  $k$ -ten Zeile zu Zeilen  $k + 1, \dots, n + 1$  so, dass in der  $k$ -ten Spalte alle Elemente unterhalb der Hauptdiagonale verschwinden.

Rechenaufwand:  $\approx (n + 1 - k)^2$  Rechenoperationen zur Transformation von  $n + 1 - k$  Zeilen mit je  $n + 1 - k$  Spalten

**gesamt**  $n$  Gaußschritte mit insgesamt

$$\sum_{k=1}^n (n + 1 - k)^2 = \sum_{l=1}^n l^2 = \frac{n(n + 1)(2n + 1)}{6}$$

Rechenoperationen.

**Ergebnis:** Rechenaufwand  $\frac{n^3}{3} + \mathcal{O}(n^2)$  Rechenoperationen, wächst kubisch mit  $n$ .

**Bemerkung 1.7 (Landau–Symbole)**

Sei  $x_0 \in \mathbb{R}$  und  $g : I \rightarrow \mathbb{R}$  in einer Umgebung von  $x_0$  definiert. Gibt es für ein  $p \in \mathbb{R}$ ,  $p \geq 0$  eine positive Konstante  $\bar{c} \in \mathbb{R}$ , so dass für alle  $x$  in einer hinreichend kleinen Umgebung von  $x_0$  die Abschätzung

$$|g(x)| \leq \bar{c} \cdot |x - x_0|^p$$

erfüllt ist, so schreibt man

$$g(x) = \mathcal{O}((x - x_0)^p), \quad (x \rightarrow x_0)$$

**spricht:** „ $g(x)$  ist groß O von  $(x - x_0)^p$ “.

**Beispiel**  $\sin x = \mathcal{O}(x)$ ,  $(x \rightarrow 0)$

Existiert  $\lim_{x \rightarrow x_0} \frac{g(x)}{(x - x_0)^p}$  und ist  $\lim_{x \rightarrow x_0} \frac{g(x)}{(x - x_0)^p} = 0$ , so schreibt man

$$g(x) = o((x - x_0)^p), \quad (x \rightarrow x_0)$$

**spricht:** „klein o“.

**Beispiel**  $\sqrt{x^3} = o(x)$ ,  $(x \rightarrow 0)$

Entsprechend bedeutet

$$v(n) = \mathcal{O}(n^p), \quad (n \rightarrow \infty)$$

für eine Funktion  $v : \mathbb{N} \rightarrow \mathbb{R}$ , dass für alle  $n \geq n_0$  die Abschätzung  $|v(n)| \leq \bar{c}n^p$  mit einer gewissen positiven Konstanten  $\bar{c} \in \mathbb{R}$  erfüllt ist, und

$$v(n) = o(n^p), \quad (n \rightarrow \infty)$$

steht für

$$\lim_{n \rightarrow \infty} \frac{v(n)}{n^p} = 0.$$

**Beispiel**  $v(n) = \frac{n(n+1)(2n+1)}{6}$

Es gilt  $v(n) < n \cdot 2n \cdot 3n / 6 = n^3$ , also  $v(n) = \mathcal{O}(n^3)$ . Genauer gilt  $v(n) = n^3/3 + w(n)$  mit  $w(n) = \frac{1}{2}n^2 + \frac{1}{6}n = \mathcal{O}(n^2)$ , man schreibt kurz:  $v(n) \doteq n^3/3$ .

**Bemerkung 1.8 (Klassische Polynominterpolation: Lagrange–Darstellung)**

Elementarer Zugang aus Bemerkung 1.6  $\Rightarrow$  Interpolationspolynom  $\Phi^{(n)}$  in monomialer Basis  $\{1, x, x^2, \dots, x^n\}$ :

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



## Lagrange-Darstellung

$$\Phi^{(n)}(x) = \sum_{j=0}^n y_j L_j^{(n)}(x)$$

mit den *Lagrangeschen Basispolynomen*

$$L_j^{(n)}(x) := \prod_{\substack{i=0 \\ i \neq j}}^n \frac{x - x_i}{x_j - x_i} = \frac{(x - x_0)(x - x_1) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n)}{(x_j - x_0)(x_j - x_1) \cdots (x_j - x_{i-1})(x_j - x_{i+1}) \cdots (x_j - x_n)},$$

die die Interpolationsbedingungen

$$L_j(x_k) = \delta_{kj} = \begin{cases} 1 & \text{falls } k = j, \\ 0 & \text{sonst,} \end{cases} \quad (k = 0, 1, \dots, n)$$

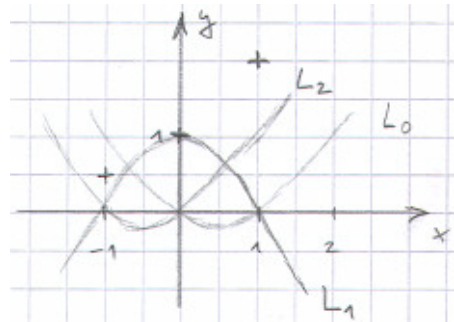
erfüllen.

**Beispiel**  $n = 2$ ,  $x_0 = -1$ ,  $x_1 = 0$ ,  $x_2 = 1$

$$L_0^{(2)}(x) = \frac{(x - 0)(x - 1)}{(-1 - 0)(-1 - 1)}$$

$$L_1^{(2)}(x) = \frac{(x + 1)(x - 1)}{(0 + 1)(0 - 1)}$$

$$L_2^{(2)}(x) = \frac{(x + 1)(x - 0)}{(1 + 1)(1 - 0)}$$



## Bemerkung 1.9 (Horner-Schema)

Auswertung des Interpolationspolynoms mit  $\mathcal{O}(n)$  Rechenoperationen:

$$\begin{aligned} \Phi^{(n)}(x) &= \sum_{j=0}^n a_j x^j = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n \\ &= a_0 + x(a_1 + x(a_2 + \dots + x(a_{n-1} + x a_n) \dots)) \end{aligned}$$

## Horner-Schema

	$a_n$	$a_{n-1}$	$a_{n-2}$	$\dots$	$a_0$
$x$		$x \cdot a_n$	$x \cdot (a_{n-1} + x a_n)$	$\dots$	$x \cdot (\dots)$
	$x \cdot a_n$	$a_{n-1} + x a_n$	$a_{n-2} + x(a_{n-1} + x a_n)$	$\dots$	$\dots = \Phi^{(n)}(x)$

**Beispiel** Auswertung von  $\Phi(x) = (x - 1)^3 = x^3 - 3x^2 + 3x - 1$  an der Stelle  $x = 5$ :

	1	-3	3	-1
5		$5 \cdot 1 = 5$	$5 \cdot 2 = 10$	$5 \cdot 13 = 65$
	1	$-3 + 5 = 2$	$3 + 10 = 13$	$-1 + 65 = 64 = \Phi(5)$

## Algorithmus

```

p := a_n
for i = n : -1 : 1
  | p := a_{i-1} + x * p

```

**Matlab-Code** Speicherschema für Vektoren in Matlab  $(a_0, a_1, \dots, a_n) = \mathbf{a}(1:(n+1))$

```

p = a(n+1);
for i=n:-1:1,
  p = a(i) + x * p;
end;

```

### Bemerkung 1.10 (Rekursive Auswertung des Interpolationspolynoms)

Sei  $\Phi_{i,\dots,i+l} \in \Pi_l$  das Interpolationspolynom zu Stützpunkten

$$(x_k, y_k), \quad (k = i, i + 1, \dots, i + l).$$

Dann gilt

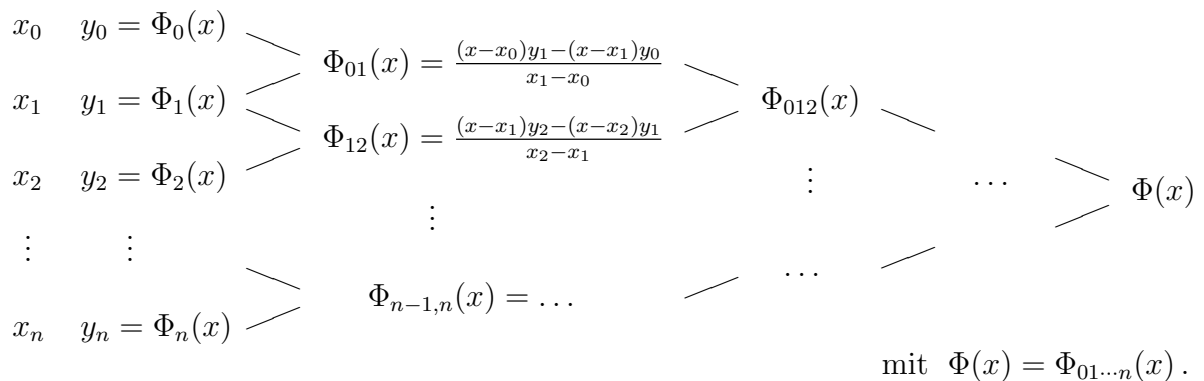
$$\begin{aligned} \Phi_{i+1,\dots,i+l}(x_k) &= y_k, \quad (k = i + 1, \dots, i + l), \\ \Phi_{i,\dots,i+l-1}(x_k) &= y_k, \quad (k = i, \dots, i + l - 1), \end{aligned}$$

also

$$\Phi_{i,\dots,i+l}(x) = \frac{(x - x_i)\Phi_{i+1,\dots,i+l}(x) - (x - x_{i+l})\Phi_{i,\dots,i+l-1}(x)}{x_{i+l} - x_i}, \quad (*)$$

denn der rechts stehende Ausdruck erfüllt  $\Phi_{i,\dots,i+l}(x_k) = y_k$  für  $k = i$ , für  $k = i + l$  und für  $k = i + 1, \dots, i + l - 1$ .

### Neville-Schema



**Beispiel 1.11 (Polynominterpolation: Neville–Schema)**

**Idee** Approximiere  $\sqrt{2} = 2^{1/2}$  durch  $\Phi(1/2)$  mit dem Interpolationspolynom  $\Phi \in \Pi_2$  zu  $(-1, 1/2)$ ,  $(0, 1)$  und  $(1, 2)$ .

**Lagrange–Darstellung**

$$\Phi\left(\frac{1}{2}\right) = \sum_{j=0}^2 y_j L_j^{(2)}\left(\frac{1}{2}\right) = \frac{1}{2} \cdot L_0^{(2)}\left(\frac{1}{2}\right) + 1 \cdot L_1^{(2)}\left(\frac{1}{2}\right) + 2 \cdot L_2^{(2)}\left(\frac{1}{2}\right)$$

mit den Lagrangeschen Basispolynomen  $L_j^{(2)}(x)$  aus Bemerkung 1.8.

**Neville**

$$\begin{array}{rcl} -1 & \frac{1}{2} & \searrow \\ & & \frac{(\frac{1}{2} - (-1)) \cdot 1 - (\frac{1}{2} - 0) \cdot \frac{1}{2}}{0 - (-1)} = \frac{5}{4} \\ 0 & 1 & \nearrow \\ & & \searrow \\ & & \frac{(\frac{1}{2} - 0) \cdot 2 - (\frac{1}{2} - 1) \cdot 1}{1 - 0} = \frac{3}{2} \\ 1 & 2 & \nearrow \end{array} \quad \begin{array}{l} \searrow \\ \nearrow \end{array} \quad \frac{(\frac{1}{2} - (-1)) \cdot \frac{3}{2} - (\frac{1}{2} - 1) \cdot \frac{5}{4}}{1 - (-1)} = \frac{23}{16} \approx 1.4375$$

**Bemerkung 1.12 (Klassische Polynominterpolation: Newton–Darstellung)**

**Problem** Neville–Schema erfordert  $\mathcal{O}(n^2)$  Rechenoperationen, ebenso die Auswertung von  $\sum_j y_j L_j^{(n)}(x)$ , aber

$$\sum_{j=0}^n a_j x^j = a_0 + x(a_1 + x(a_2 + \dots + x(a_{n-1} + x a_n) \dots))$$

kann mit  $\mathcal{O}(n)$  Rechenoperationen ausgewertet werden.

**gesucht** Darstellung des Interpolationspolynoms, die man effizient bestimmen kann und die mit Horner–artigem Schema ausgewertet werden kann.

**Ansatz** Es gilt

$$\Phi_{i,i+1,\dots,i+l}(x) = \Phi_{i,i+1,\dots,i+l-1}(x) + (x - x_i)(x - x_{i+1}) \cdots (x - x_{i+l-1}) f_{i,i+1,\dots,i+l}$$

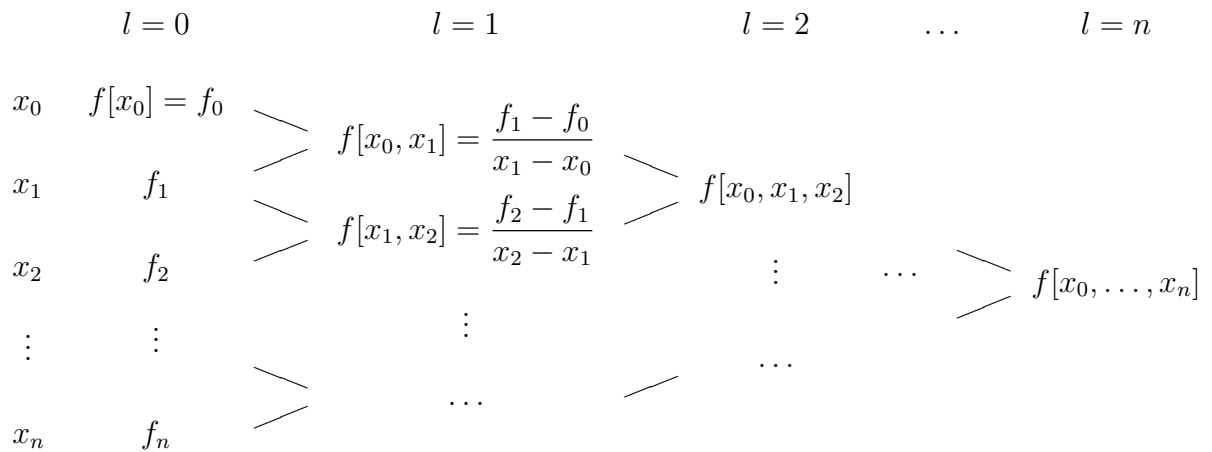
mit einem  $f_{i,i+1,\dots,i+l} \in \mathbb{R}$ , denn

$$\Phi_{i,i+1,\dots,i+l}(x_k) = \Phi_{i,i+1,\dots,i+l-1}(x_k), \quad (k = i, i + 1, \dots, i + l - 1).$$

Durch Vergleich der Koeffizienten von  $x^l$  in (\*) aus Bemerkung 1.10 folgt

$$f_{i,i+1,\dots,i+l} = \frac{f_{i+1,\dots,i+l} - f_{i,\dots,i+l-1}}{x_{i+l} - x_i}.$$

Für  $f_k = f(x_k)$  bezeichnet man diese *dividierten Differenzen* mit  $f[x_i, \dots, x_{i+l}]$ .  
 Rekursive Berechnung mittels *Steigungsschema*



mit

$$f[x_0, x_1, x_2] = \frac{\frac{f_2 - f_1}{x_2 - x_1} - \frac{f_1 - f_0}{x_1 - x_0}}{x_2 - x_0}, \dots$$

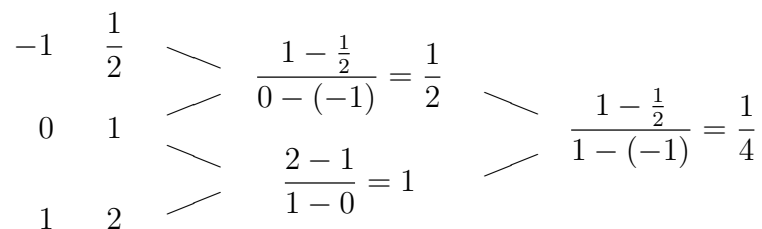
### Newtonsche Darstellung

$$\Phi(x) = f[x_0] + (x - x_0) f[x_0, x_1] + (x - x_0)(x - x_1) f[x_0, x_1, x_2] + \dots + (x - x_0)(x - x_1) \dots (x - x_{n-1}) f[x_0, x_1, \dots, x_n]$$

### Newton-Horner-Schema

$$\Phi(x) = f[x_0] + (x - x_0) (f[x_0, x_1] + (x - x_1) (f[x_0, x_1, x_2] + \dots + (x - x_{n-2}) (f[x_0, x_1, \dots, x_{n-1}] + (x - x_{n-1}) f[x_0, x_1, \dots, x_n]) \dots))$$

**Beispiel** vgl. Beispiel 1.11



$$\Phi(x) = \frac{1}{2} + (x - (-1)) \cdot \frac{1}{2} + (x - (-1))(x - 0) \cdot \frac{1}{4} = \frac{1}{2} + (x + 1) \cdot \left(\frac{1}{2} + x \cdot \frac{1}{4}\right)$$

$$\Phi\left(\frac{1}{2}\right) = \frac{1}{2} + \left(\frac{1}{2} + 1\right) \cdot \left(\frac{1}{2} + \frac{1}{2} \cdot \frac{1}{4}\right) = \frac{1}{2} + \frac{3}{2} \cdot \frac{5}{8} = \frac{23}{16}$$

## Algorithmen

Steigungsschema

```
for i = n : -1 : 0
  | a_i := f_i
  | for j = i + 1 : n
  |   | a_j := (a_j - a_{j-1}) / (x_j - x_i)
```

Newton–Horner–Schema

```
p := a_n
for i = n - 1 : -1 : 0
  | p := a_i + (x - x_i) p
```

### Satz 1.13 (Restglied der Polynominterpolation)

Sei  $f \in C^{n+1}[a, b]$  und  $\Phi$  das Interpolationspolynom zu den Stützstellen

$$a \leq x_0 < x_1 < \dots < x_n \leq b,$$

d. h.  $\Phi(x_k) = f(x_k)$ , ( $k = 0, 1, \dots, n$ ). Dann gibt es zu jedem  $\bar{x} \in [a, b]$  ein  $\xi \in [a, b]$  mit

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

**Beweis** Die Behauptung ist trivial für  $x = x_k$ , ( $k = 0, 1, \dots, n$ ).

Andernfalls betrachtet man

$$g(x) := f(x) - \Phi(x) - K(x - x_0)(x - x_1) \cdots (x - x_n)$$

mit

$$K := \frac{f(\bar{x}) - \Phi(\bar{x})}{(\bar{x} - x_0)(\bar{x} - x_1) \cdots (\bar{x} - x_n)}.$$

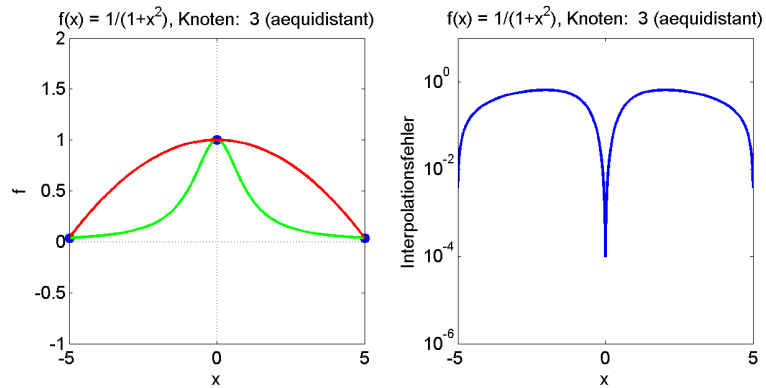
Die Funktion  $g$  hat in  $[a, b]$  (mindestens)  $n+2$  Nullstellen:  $x_0, x_1, \dots, x_n, \bar{x}$ . Nach dem Satz von Rolle hat  $g'$  mindestens  $n+1$  Nullstellen usw. und schließlich  $g^{(n+1)}(x)$  mindestens eine Nullstelle  $\xi \in [a, b]$ . Wegen

$$\begin{aligned} 0 &= g^{(n+1)}(\xi) \\ &= f^{(n+1)}(\xi) - \underbrace{\frac{d^{n+1}\Phi(x)}{dx^{n+1}} \Big|_{x=\xi}}_{= 0, \text{ da } \Phi \in \Pi_n} - K \cdot \frac{d^{n+1}}{dx^{n+1}} \left( (x - x_0)(x - x_1) \cdots (x - x_n) \right) \Big|_{x=\xi} \\ &= f^{(n+1)}(\xi) - K \cdot (n+1)! \end{aligned}$$

folgt schließlich  $K = \frac{f^{(n+1)}(\xi)}{(n+1)!}$  und hieraus die Behauptung. ■

## Beispiel 1.15: Funktion von Runge

Beispiel  $n = 2$ ,  $x_0 = -5$ ,  $x_1 = 0$ ,  $x_2 = 5$   
 $y_j = 1/(1 + x_j^2)$ , ( $j = 0, 1, 2$ )

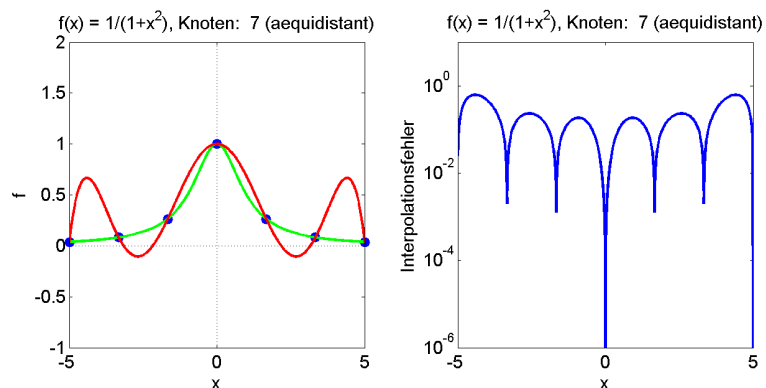


Martin-Luther-Universität Halle-Wittenberg, FB Mathematik und Informatik  
Martin Arnold: Numerische Mathematik für Fachrichtung Informatik und Lehramt (WiS 2005/06)

Abbildung 1.8: Interpolation der Funktion von Runge:  $\Phi^{(2)}(x)$ , Stützstellen äquidistant.

## Beispiel 1.15: Funktion von Runge (II)

Beispiel  $n = 6$ ,  $x_j = -5 + 10j/6$ , ( $j = 0, 1, \dots, 6$ )  
 $y_j = 1/(1 + x_j^2)$ , ( $j = 0, 1, \dots, 6$ )

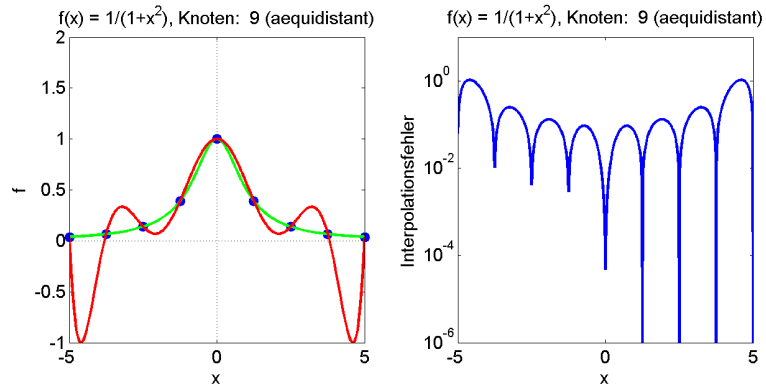


Martin-Luther-Universität Halle-Wittenberg, FB Mathematik und Informatik  
Martin Arnold: Numerische Mathematik für Fachrichtung Informatik und Lehramt (WiS 2005/06)

Abbildung 1.9: Interpolation der Funktion von Runge:  $\Phi^{(6)}(x)$ , Stützstellen äquidistant.

### Beispiel 1.15: Funktion von Runge (III)

Beispiel  $n = 8$ ,  $x_j = -5 + 10j/8$ , ( $j = 0, 1, \dots, 8$ )  
 $y_j = 1/(1 + x_j^2)$ , ( $j = 0, 1, \dots, 8$ )

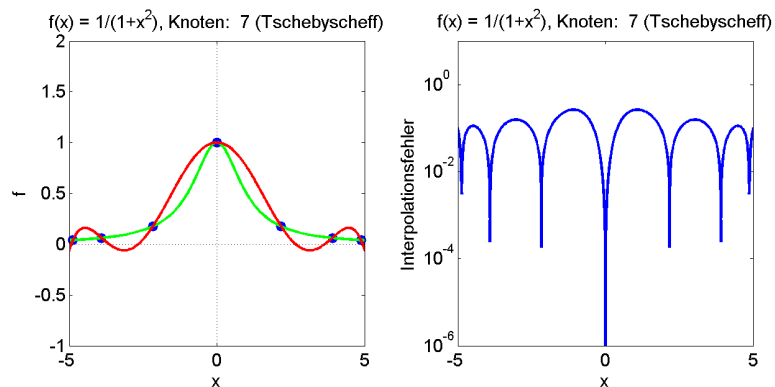


Martin-Luther-Universität Halle-Wittenberg, FB Mathematik und Informatik  
 Martin Arnold: Numerische Mathematik für Fachrichtung Informatik und Lehramt (WiS 2005/06)

Abbildung 1.10: Interpolation der Funktion von Runge:  $\Phi^{(8)}(x)$ , Stützstellen äquidistant.

### Beispiel 1.15: Funktion von Runge (IV)

Beispiel  $n = 6$ ,  $x_j = 5 \cos \frac{(2j+1)\pi}{14}$ , ( $j = 0, 1, \dots, 6$ )  
 $y_j = 1/(1 + x_j^2)$ , ( $j = 0, 1, \dots, 6$ )

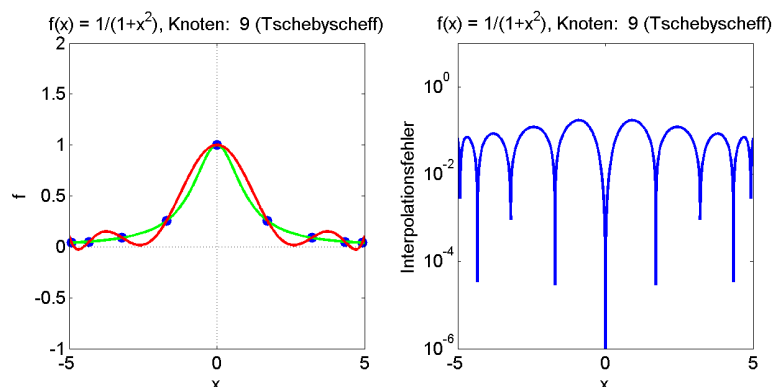


Martin-Luther-Universität Halle-Wittenberg, FB Mathematik und Informatik  
 Martin Arnold: Numerische Mathematik für Fachrichtung Informatik und Lehramt (WiS 2005/06)

Abbildung 1.11: Interpolation der Funktion von Runge:  $\Phi^{(6)}(x)$ , Tschebyscheff-Stützstellen.

## Beispiel 1.15: Funktion von Runge (V)

**Beispiel**  $n = 8$ ,  $x_j = 5 \cos \frac{(2j+1)\pi}{18}$ , ( $j = 0, 1, \dots, 8$ )  
 $y_j = 1/(1+x_j^2)$ , ( $j = 0, 1, \dots, 8$ )



Martin-Luther-Universität Halle-Wittenberg, FB Mathematik und Informatik  
 Martin Arnold: Numerische Mathematik für Fachrichtung Informatik und Lehramt (WiS 2005/06)

Abbildung 1.12: Interpolation der Funktion von Runge:  $\Phi^{(8)}(x)$ , Tschebyscheff-Stützstellen.

### Bemerkung 1.14 (Wahl der Stützstellen)

a) Nach Satz 1.13 sollte man die Stützstellen möglichst so wählen, dass

$$\max_{x \in [a,b]} |(x - x_0)(x - x_1) \cdots (x - x_n)| \rightarrow \min$$

Typische Stützstellen:

- äquidistant:  $x_j := a + jh$ , Schrittweite  $h := \frac{b-a}{n}$
- Tschebyscheff-Nullstellen:  $[a, b] = [-1, 1]$

$$x_j := \cos \frac{(2j+1)\pi}{2n+2}, \quad (j = 0, 1, \dots, n)$$

- b) Zu jeder Folge von Stützstellen lässt sich ein  $f \in C[a, b]$  angeben, so dass die zugehörige Folge der Interpolationspolynome nicht gleichmäßig konvergiert (Satz von Faber).
- c) Praktische Erfahrung: Polynome hohen Grades neigen zu Oszillationen und sollten vermieden werden.



**Beispiel 1.15 (Funktion von Runge)**

Äquidistante Stützstellen sind ungeeignet zur Interpolation der *Funktion von Runge*

$$f(x) = \frac{1}{1+x^2}, \quad (x \in [-5, 5])$$

(vgl. Abb. 1.8, 1.9 und 1.10). Bessere Ergebnisse für Tschebyscheff–Stützstellen (vgl. Abb. 1.11 und 1.12).

**Bemerkung 1.16 (Hermite–Interpolation)**

**Klassische Polynominterpolation** Bestimme zu gegebenen Stützstellen  $x_j$  und gegebenen Stützwerten  $y_j$  ein Polynom  $\Phi \in \Pi_n$ , das die  $n + 1$  Interpolationsbedingungen

$$\Phi(x_j) = y_j, \quad (j = 0, 1, \dots, n)$$

erfüllt.

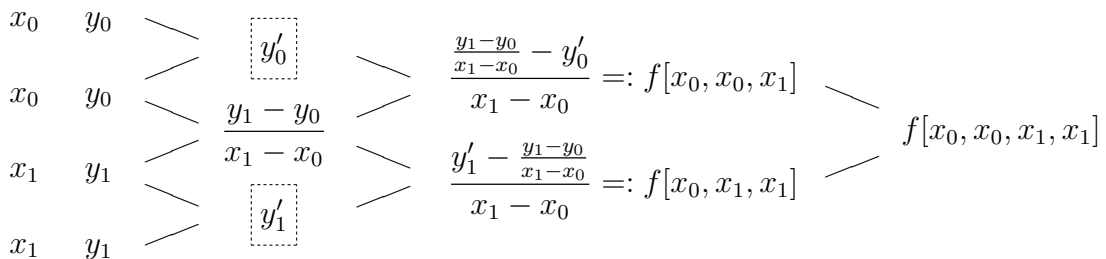
**(Klassische) Hermite–Interpolation** Bestimme zu gegebenen Stützstellen  $x_j$  und gegebenen Stützwerten  $(y_j, y'_j)$  ein Polynom  $\Phi \in \Pi_{2n+1}$ , das die  $2(n + 1)$  Interpolationsbedingungen

$$\Phi(x_j) = y_j, \quad \Phi'(x_j) = y'_j, \quad (j = 0, 1, \dots, n)$$

erfüllt.

**Newtonsche Darstellung** Füge die Stützpunkte  $(x_j, y_j)$ ,  $(j = 0, 1, \dots, n)$  jeweils zweimal in das Steigungsschema ein und ersetze auftretende Quotienten „0/0“ durch die vorgegebenen Funktionswerte  $y'_j$  der Ableitung  $\Phi'(x)$ .

**Beispiel**  $n = 1$



mit

$$f[x_0, x_0, x_1, x_1] := \frac{f[x_0, x_1, x_1] - f[x_0, x_0, x_1]}{x_1 - x_0}$$

**Ergebnis**

$$\Phi(x) = y_0 + (x - x_0) \cdot y'_0 + (x - x_0)^2 \cdot f[x_0, x_0, x_1] + (x - x_0)^2(x - x_1) \cdot f[x_0, x_0, x_1, x_1]$$

Effiziente Auswertung mit Newton–Horner–Schema.

**Höhere Ableitungen** Interpolationsbedingungen

$$\frac{d^k \Phi}{dx^k}(x_j) = y_j^{(k)}, \quad (j = 0, 1, \dots, n; k = 0, 1, \dots, c_j)$$

bestimmen das Hermite-Interpolationspolynom  $\Phi \in \Pi_r$  mit  $r + 1 = \sum_{j=0}^n (1 + c_j)$ .