3. Interpolation durch Polynome

" Interpolation ist die Kunst zwischen den Zeilen einer Tabelle zu lesen "
(Rutishauser)

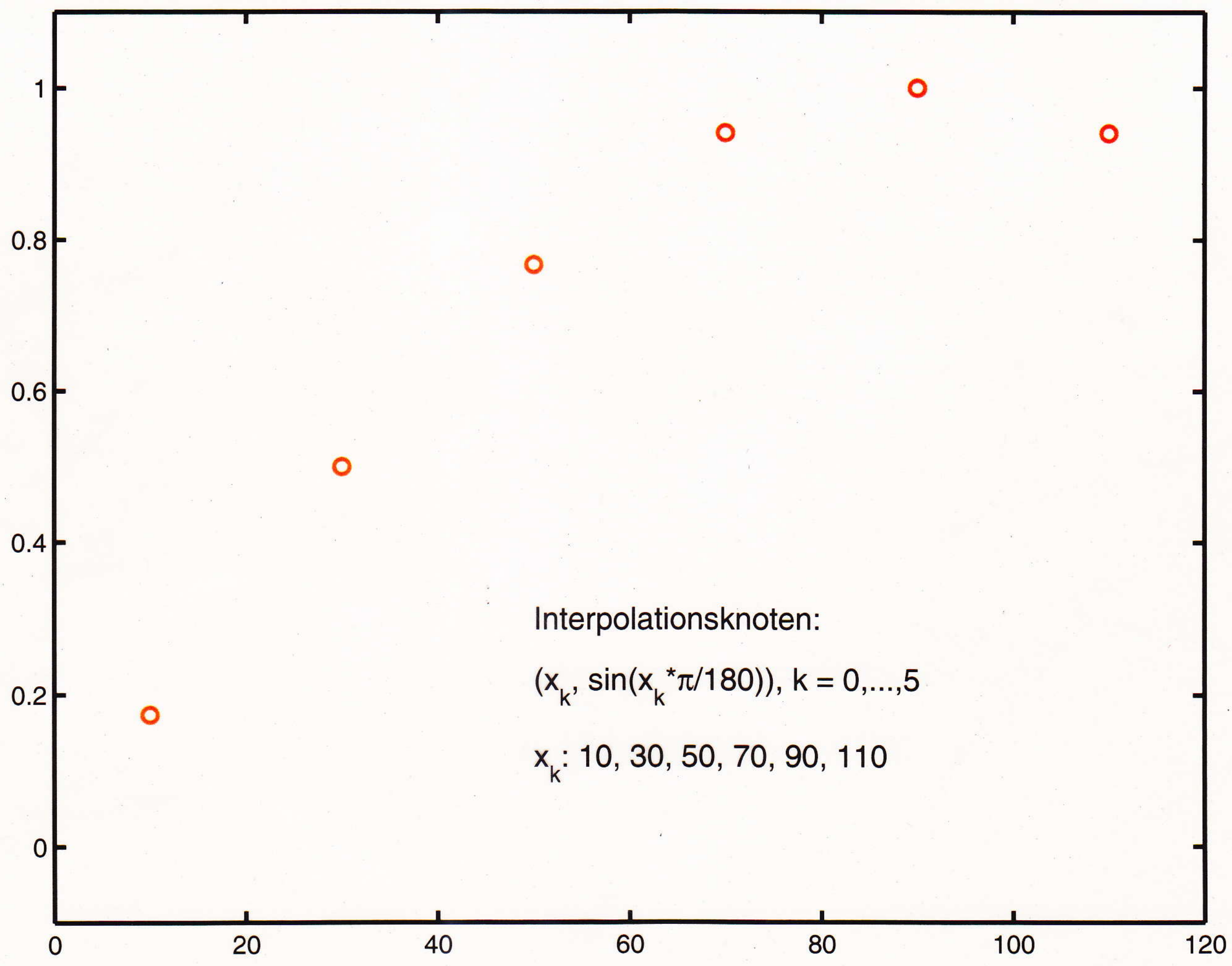
Beispiel:

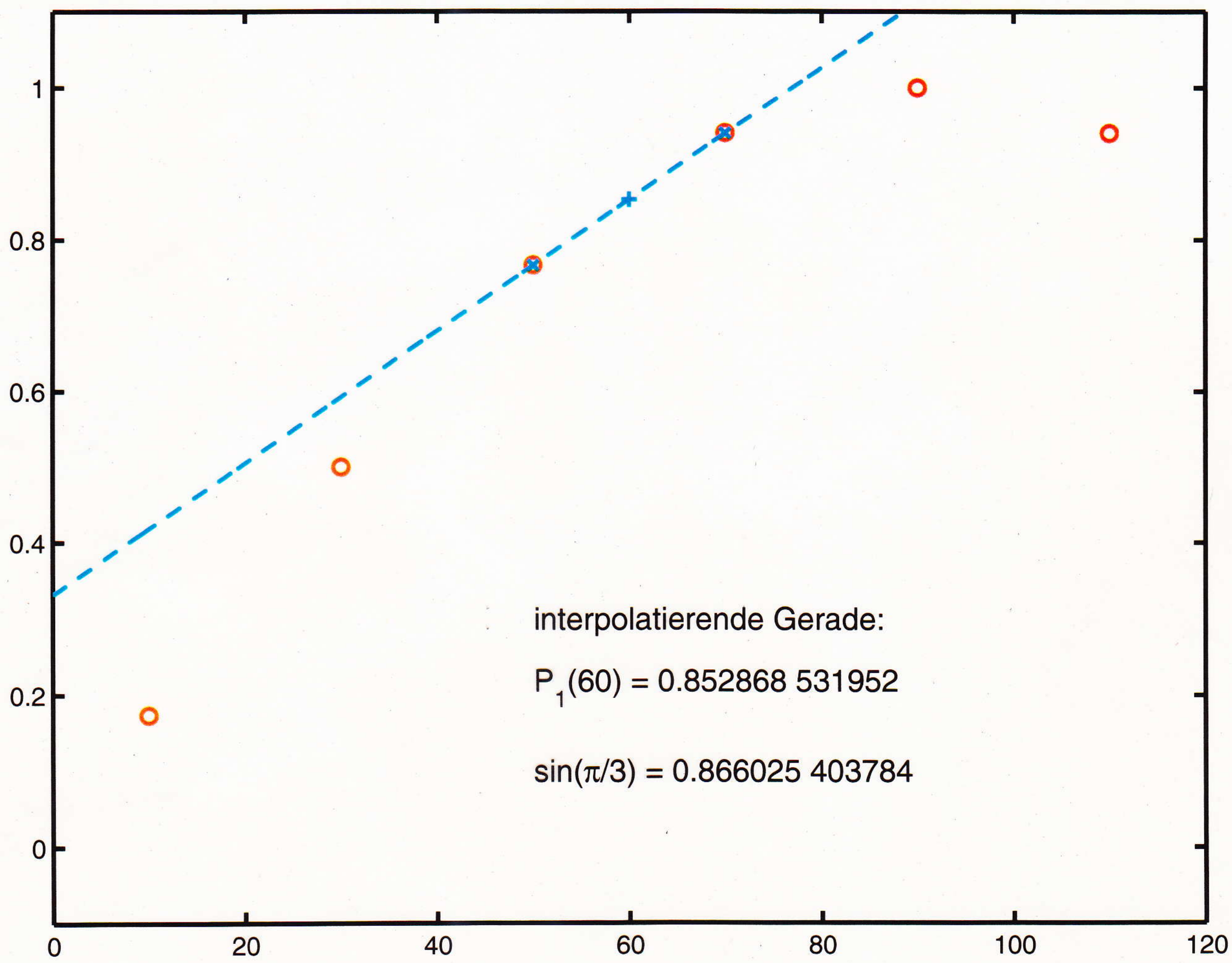
x_i	$\sin(x_i)$
50°	0.7660444431
51°	0.7771496145
52°	0.7880107536
53°	0.7986355100
54°	0.8090169943
55°	0.8191520442

Gesucht: $\sin(52.732°) = ?$

Idee:
- Gerade durch $(x, \sin x)$, $x = 52°, 53°$
- Parabel durch $(x, \sin x)$, $x = 52°/53°/54°$
- Kubisches Polynom durch $(x, \sin x)$ mit $x = 51°/52°/53°/54°$

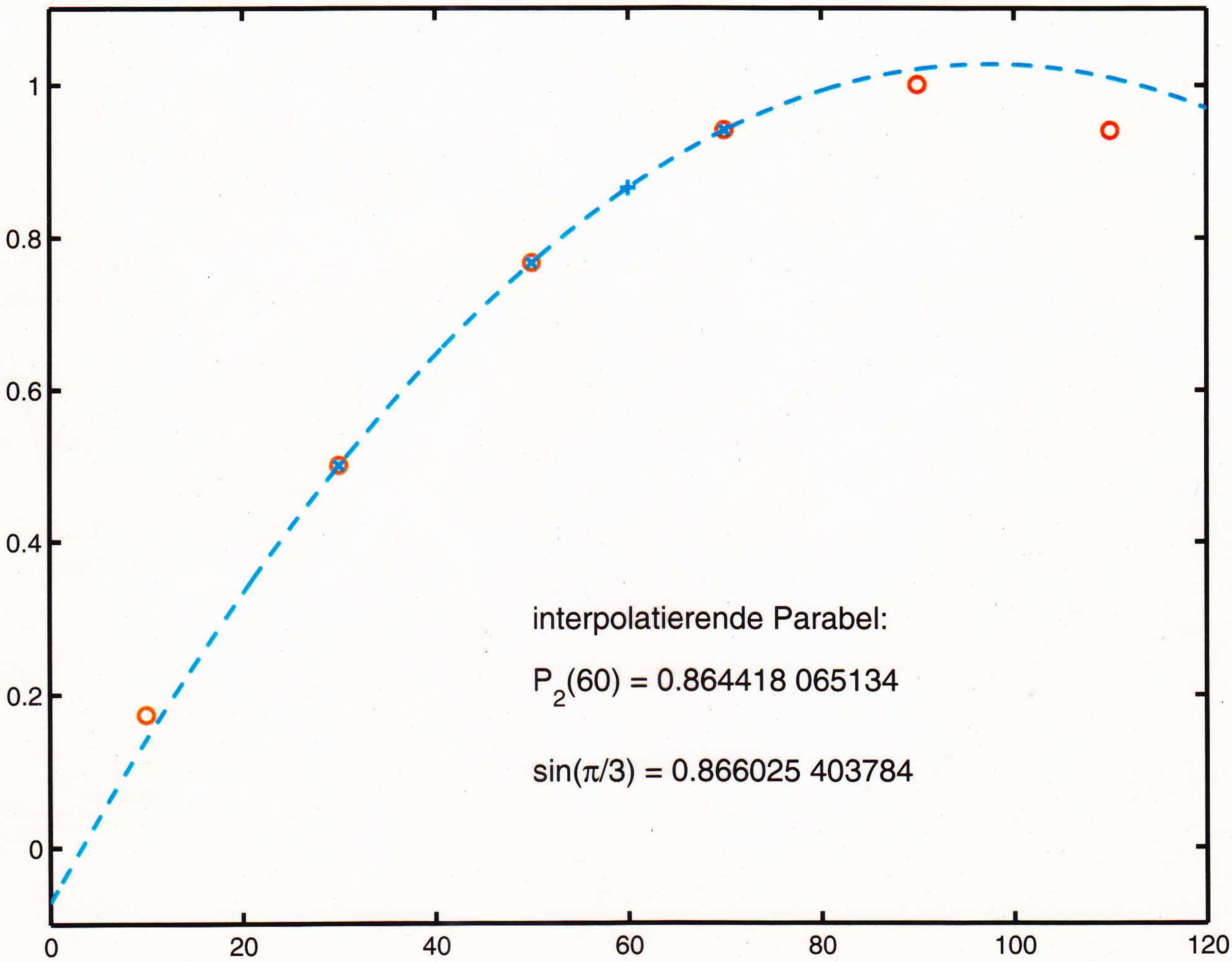
⋮


Interpolationsknoten:
$(x_k, \sin(x_k*\pi/180))$, $k = 0,\dots,5$
x_k: 10, 30, 50, 70, 90, 110
0
0.2
0.4
0.6
0.8
1
0
20
40
60
80
100
120

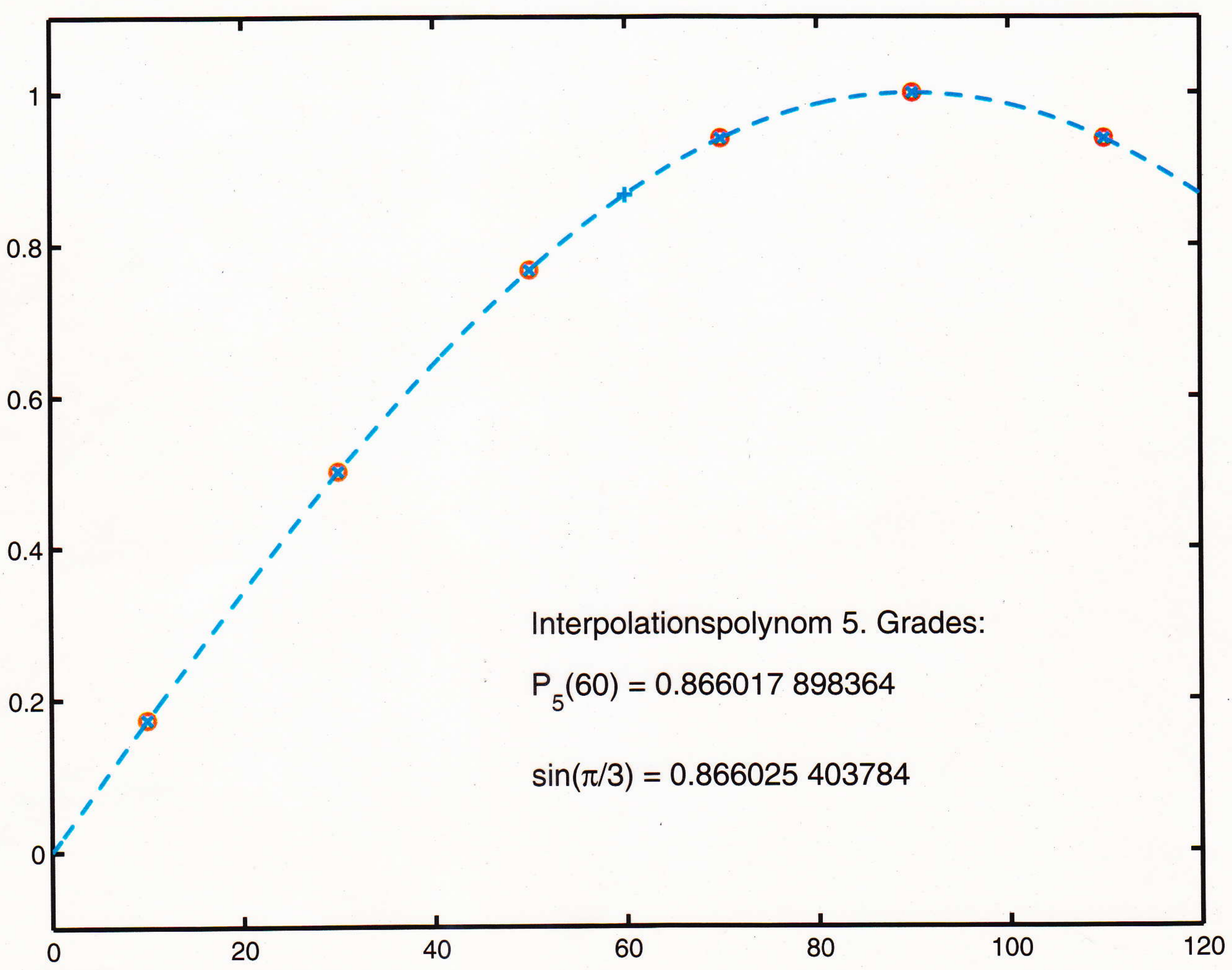


1
0.8
0.6
0.4
0.2
0
0
20
40
60
80
100
120
interpolatierende Gerade:
P1(60) = 0.852868 531952
sin(π/3) = 0.866025 403784

35c


interpolatierende Parabel:
$P_2(60) = 0.864418\,065134$
$\sin(\pi/3) = 0.866025\,403784$
1
0.8
0.6
0.4
0.2
0
0
20
40
60
80
100
120

35d



1
0.8
0.6
0.4
0.2
0
0
20
40
60
80
100
120
Interpolationspolynom 5. Grades:
P5(60) = 0.866017 898364
sin(π/3) = 0.866025 403784

Joseph Louis Lagrange
1736 – 1813

3.1 Lagrange-Darstellung*)

Problem: Zu vorgegebenen $(n+1)$ Stützstellen

$$(x_i, f_i), \quad i = 0, 1, \dots, n$$

mit $x_i \neq x_j$ für $i \neq j$ wird ein Polynom $P \in \Pi_n$ gesucht, das diese Daten interpoliert, d.h. $P(x_i) = f_i$, $i = 0, 1, \dots, n$.

Beachte: Die x_i, f_i können reell oder komplex sein!

> **Satz 3.1** Zu $(n+1)$ Stützstellen (x_i, f_i) mit $x_i \neq x_j$ $(i \neq j)$ gibt es genau ein Interpolationspolynom $P \in \Pi_n$.

Beweis:

a) Eindeutigkeit: Gäbe es zwei Interpolationspolynome $P_1, P_2 \in \Pi_n$, so wäre $Q := P_2 - P_1 \in \Pi_n$ mit $(n+1)$ Nullstellen

$$Q(x_i) = P_2(x_i) - P_1(x_i) = f_i - f_i = 0.$$

---

*) Joseph Louis Lagrange (1736-1813) Turin, Berlin, Paris

Nach (2.7) folgt $Q=0$, d.h. $P_2 = P_1$.

b) Existenz: Zu den Knoten x_i, $i=0,1,\dots,N$ werden die folgenden **Lagrange-Polynome** definiert:

$$L_k(x) := \prod_{\substack{j=0 \\ j\neq k}}^{n} \left(\frac{x - x_j}{x_k - x_j} \right) \in \Pi_n \qquad \underline{(3.2)}$$

Man sieht dann durch Einsetzen:

$$L_k(x_i) = \delta_{ki} = \begin{cases} 0, & \text{für } k \neq i \\ 1, & \text{für } k = i \end{cases}$$

Hiermit folgt, dass das Polynom

$$\boxed{P(x) := \sum_{k=0}^{n} f_k L_k(x) \in \Pi_n} \qquad \underline{(3.3)}$$

tatsächlich interpoliert! ▲

(3.3) heißt **Lagrange-Interpolations = formel**.

Beispiel

x_k	0	1	3
f_k	1	3	2

$$L_0(x) = \frac{(x-1)(x-3)}{(0-1)(0-3)} = \frac{1}{3}(x-1)(x-3)$$

$$L_1(x) = \frac{(x-0)(x-3)}{(1-0)(1-3)} = -\frac{1}{2}x(x-3)$$

$$L_2(x) = \frac{(x-0)(x-1)}{(3-0)(3-1)} = \frac{1}{6}x(x-1)$$

$\Longrightarrow$

$$\underline{P(x) = \frac{1}{3}(x-1)(x-3) - \frac{3}{2}x(x-3) + \frac{1}{3}x(x-1)}$$

Bemerkungen:

a) Die Lagrange-Darstellung (3.3) ist äquivalent zur „Normalform" $\sum_{k=0}^{n} a_k x^k$ (andere Basis). Eine Umrechnung in die Normalform ist i. Allg. nicht zu empfehlen!

Beispiel: Knoten: x_k: 10, 11, 12, ..., 20

Stützwerte: $f_k = \begin{cases} 0 & \text{für } x_k \neq 15 \\ 1 & \text{für } x_k = 15 \end{cases}$

$$\Longrightarrow P(x) = \frac{(x-10)(x-11)\ldots\widehat{(x-15)}\ldots(x-20)}{(15-10)(15-11)\ldots\widehat{(15-15)}\ldots(15-20)}$$

$$a_0 = P(0) = \frac{(-10)(-11)\ldots\widehat{(-15)}\ldots(-20)}{5\cdot 4\cdot 3\cdot 2\cdot 1\cdot(-1)\ldots(-5)} \approx \underline{3.1\cdot 10^7}$$

$\Longrightarrow$ Die Auswertung von $P(x)$ in Normalform ist mit Auslöschung verbunden!!

b) Störungen der Stützwerte f_k :

$$\Delta P(x) = \sum_{k=0}^{n} \tilde{f}_k L_k(x) - \sum_{k=0}^{n} f_k L_k(x)$$

$$= \sum_{k=0}^{n} L_k(x) \cdot \Delta f_k$$

$$\Longrightarrow |\Delta P(x)| \leq \left(\sum_{k=0}^{n} |L_k(x)| \right) \cdot \max_{k=0,..,n} |\Delta f_k|$$

Die Größe $\boxed{K_{abs}(x) := \sum_{k=0}^{n} |L_k(x)|}$ lässt sich also als **absolute Konditionszahl** des Interpolationsproblems interpretieren.

3.2 Verfahren von Aitken, Neville

Grundidee: Rekursive Berechnung des Interpolationspolynoms. Dabei wird zunächst nur über Teile der Stützstellenmenge (x_k, f_k), $k=0,1,..,n$ interpoliert.

$P_{i,k}(x)$: Interpol.polynom zu den Stützstellen $(x_i, f_i), (x_{i+1}, f_{i+1}), \ldots, (x_{i+k}, f_{i+k})$ $(0 \leq i \leq i+k \leq n)$; $\in \Pi_k$

Lemma von Aitken (3.4)

$$P_{i0}(x) = f_i$$

$$P_{ik}(x) = P_{i+1,k-1}(x) + \frac{x - x_{i+k}}{x_i - x_{i+k}} \left(P_{i,k-1}(x) - P_{i+1,k-1}(x) \right)$$

Beweis: (vollst. Induktion über k)

$\underline{k=0}$: klar

$\underline{k-1 \Rightarrow k}$: $Q(x) :=$ rechte Seite der Rekursion

Nach Ind. vor. gilt $Q \in \Pi_k$. Ferner

$Q(x_i) = f_i$ (da sich $P_{i+1,k-1}(x)$ weghebt!)

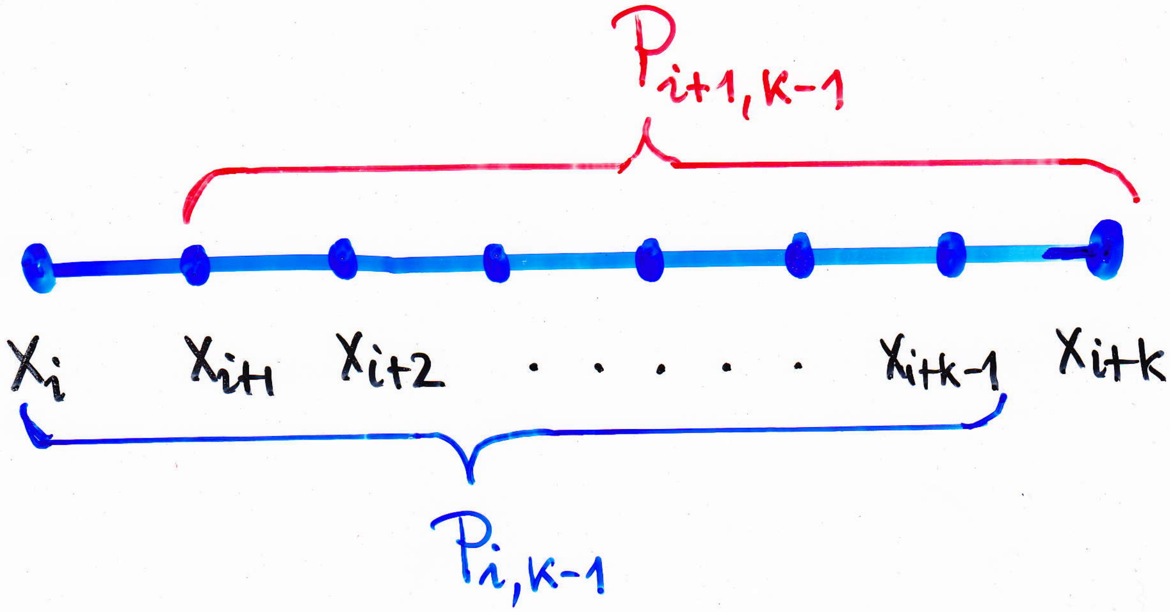
für $j = i+1, \dots, i+k-1$:

$Q(x_j) = f_j$ (da $P_{i+1,k-1}$ und $P_{i,k-1}$ interpolieren)

$Q(x_{i+k}) = f_{i+k}$ (da $P_{i+1,k-1}$ in x_{i+k} interpoliert) $\blacktriangle$

Schema von Neville:

$$\begin{array}{ccccc}
P_{00} & & & & \\
P_{10} & P_{01} & & & \\
P_{20} & P_{11} & P_{02} & & \\
\vdots & \vdots & \vdots & \ddots & \\
P_{n0} & P_{n-1,1} & P_{n-2,2} & \cdots & P_{0n} = P(x)
\end{array}$$


$P_{i+1,k-1}$
x_i
x_{i+1}
x_{i+2}
. . . . . .
x_{i+k-1}
x_{i+k}
$P_{i,k-1}$

Speicherung: Zeilenweise unter Verwendung eines eindimensionalen Feldes:

$$
\begin{array}{lllll}
P_0 & & & & \\
P_1 & P_0 & & & \\
P_2 & P_1 & P_0 & & \\
\vdots & \vdots & \vdots & \ddots & \\
P_n & P_{n-1} & P_{n-2} & \cdots & P_0
\end{array}
$$

Algorithmus (3.5)

$P_0 := f_0$,

für $i = 1, 2, \ldots n$

  $P_i := f_i$, $z := x - x_i$

  für $k = i-1, i-2, \ldots, 0$

    $P_k := P_{k+1} + \dfrac{z}{x_k - x_i}(P_k - P_{k+1})$

Bem.: Der Algorithmus von Aitken, Neville ist zu empfehlen, falls $P(x)$ nur an einer Stelle x ausgewertet werden soll (Tabelle, Extrapolation).

Beispiel: $\sin(52.732°)$ aus $\sin(50°), \sin(51°), \ldots, \sin(55°)$

52°	0.78801				
53°	0.79863	0.79578 80753			
51°	0.77714	0.79575	0.79581 16199		
54°	0.80901	0.79554	0.79581	0.79581 18032	
50°	0.76604	0.79539	0.79580	0.79581	0.79581 18045
55°	0.81915	0.79506	0.79581	0.79581	0.79581

0.79581 18045

exakt: $\sin(52.732°) = 0.79581\,18045\,420162$

Sir Isaac Newton
1643 - 1727

3.3 Darstellung nach Newton

Def. (3.6) Für $0 \le i \le i+k \le n$ bezeichne $f[x_i, x_{i+1}, .., x_{i+k}]$ den höchsten Koeffizienten des Interpolationspolynoms $P_{i,k}(x)$. Die $f[x_i, x_{i+1}, .., x_{i+k}]$ heißen Newtonsche dividierte Differenzen. Sie hängen nur von der Stützstellenmenge $\{(x_i, f_i), \ldots, (x_{i+k}, f_{i+k})\}$ ab!

Es werde nun die Newton-Basis

$$\omega_k(x) := \prod_{j=0}^{k-1} (x - x_j) \in \Pi_k \tag{3.7}$$

definiert. Dann folgt aus den Definitionen:

$$\begin{aligned} P(x) &= P_{0n}(x) \\ &= f[x_0, .., x_n]\, x^n + \ldots \\ &= f[x_0, .., x_n]\, \omega_n(x) + Q(x) \end{aligned}$$

Dabei: $Q \in \Pi_{n-1}$, Q interpoliert in $x_0, .., x_{n-1}$

$$\Longrightarrow \quad Q(x) = P_{0,n-1}(x)$$

$\Longrightarrow$

$$P(x) = P_{0n}(x) = f[x_0,..,x_n]\,\omega_n(x) + P_{0,n-1}(x)$$

Iteriert man diesen Schritt (vollst. Induktion), so erhält man die **Newton-Darstellung**:

$$P(x) = \sum_{k=0}^{n} f[x_0,..,x_k]\,\omega_k(x) \tag{3.8}$$

$$= \sum_{k=0}^{n} f[x_0,..,x_k]\,(x-x_0)\cdot\ldots\cdot(x-x_{k-1})$$

Wie berechnet man die Koeffizienten?

Aus dem Aitken-Lemma (3.4) folgt:

$$P_{ik}(x) = f[x_i,\ldots,x_{i+k}]\,x^k + \ldots$$

$$= P_{i+1,k-1} + \frac{x - x_{i+k}}{x_i - x_{i+k}}\left(P_{i,k-1} - P_{i+1,k-1}\right)$$

$$= \frac{f[x_i,..x_{i+k-1}] - f[x_{i+1},..,x_{i+k}]}{x_i - x_{i+k}}\,x^k + \ldots$$

$\Longrightarrow$ **Rekursion:**

$$f[x_i] := f_i \tag{3.9}$$

$$f[x_i\ldots x_{i+k}] = \frac{f[x_i\ldots x_{i+k-1}] - f[x_{i+1}\ldots x_{i+k}]}{x_i - x_{i+k}}$$

Newton-Schema:

$$
\begin{array}{llll}
f[x_0] & & & \\
f[x_1] & f[x_0,x_1] & & \\
\vdots & & \ddots & \\
f[x_n] & f[x_{n-1},x_n] & \cdots & f[x_0 \ldots x_n]
\end{array}
$$

Speicherung: Spaltenweise!

$$
\begin{array}{llll}
d_0 & & & \\
d_1 & d_1 & & \\
\vdots & \uparrow & \ddots & \\
d_n & d_n & \cdots & d_n
\end{array}
$$

Algorithmus (3.10)

für $j = 0, 1, \ldots, n$: $d_j = f_j$

für $k = 1, 2, \ldots, n$

- für $i = n, n-1, \ldots, k$

  - $$d_i = \frac{d_i - d_{i-1}}{x_i - x_{i-k}}$$

Die $d_0, \ldots, d_n$ enthalten dann die in (3.8) benötigten dividierten Differenzen.

Beispiel:

x_k	0	1	2	3
f_k	1	2	0	1

Dividierte Differenzen:

$$\begin{array}{c|cccc} 0 & 1 & & & \\ 1 & 2 & 1 & & \\ 2 & 0 & -2 & -3/2 & \\ 3 & 1 & 1 & 3/2 & 1 \end{array}$$

$$\Longrightarrow \quad P(x) = 1 + 1\,(x-0) - \tfrac{3}{2}(x-0)(x-1) + {}$$
$$+ 1\,(x-0)(x-1)(x-2)$$

Auswertung:

Die Auswertung von (3.8) (nicht umformen!) erfolgt mit einem angepassten Horner-Schema:

$$p := f[x_0, \dots, x_n]$$
$$\text{für } k = n-1, \dots, 0 \qquad \text{(3.11)}$$
$$\quad p := p \cdot (x - x_k) + f[x_0, \dots, x_k]$$

3.4 Interpolationsfehler

Wir betrachten ein reelles Interpolations = problem mit Interpolationsknoten

$$x_0, .., x_n \in [a,b] \subset \mathbb{R} .$$

Satz (3.12)

Es sei $f : [a,b] \to \mathbb{R}$ $(n+1)$ fach stetig differenzierbar, $f_k := f(x_k)$, $k = 0,1,..,n$ und $P \in \Pi_n [a,b]$ das zugehörige Interpolationspolynom.

Dann gilt für $a \leq x \leq b$:

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

mit $\xi \in \,]\min(x, x_0, .., x_n), \max(x, x_0, .., x_n)[$.

Beweis: Für $x \notin \{x_0, ..., x_n\}$ betrachte man die Funktion

$$g(t) := \big(f(t) - P(t)\big) - \big(f(x) - P(x)\big) \prod_{j=0}^{n} \left(\frac{t - x_j}{x - x_j} \right)$$

Dann ist $g : [a,b] \to \mathbb{R}$ $(n+1)$-fach stetig differenzierbar und hat (wenigstens) die Nullstellen $x, x_0, .., x_n$

Wir wenden nun auf g den Satz von Rolle an (→ **Analysis**) : Zwischen je zwei Nullstellen von $g(x)$ gibt es eine von $g'(x)$.

$\Longrightarrow$ g' hat wenigstens $n+1$ Nullstellen in I,
g'' hat wenigstens n Nullstellen in I,
$\vdots$ $\vdots$ $\vdots$
$g^{(n+1)}$ hat wenigstens eine Nullstelle in I

– wobei $I :=]\min(x, x_0, \ldots, x_n), \max(x, x_0, \ldots x_n)[$

$\Longrightarrow$ Es gibt ein $\xi \in I$ mit

$$0 = g^{(n+1)}(\xi) = f^{(n+1)}(\xi) - (f(x) - P(x)) \frac{(n+1)!}{\prod (x - x_j)}$$

$$\Longrightarrow \quad f(x) - P(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi) \cdot \prod_{j=0}^{n} (x - x_j) \qquad \blacktriangle$$

Bem.:

a) Es gibt i. Allg. keine weiteren Informationen über die Lage von ξ.

b) Man beachte die Ähnlichkeit des Satzes mit dem Taylorschen Satz (→ Analysis).

Beispiel (3.13)

$f(x) := \cos x$ soll äquidistant tabelliert werden, so dass kubische Interpolation einen absoluten Fehler $\leq 10^{-5}$ liefert. Wie klein ist die Schrittweite h zu wählen?

Sinnvolle Wahl: $x_{k-1} < x_k \leq x < x_{k+1} < x_{k+2}$

$$\Longrightarrow \quad |f(x) - P(x)| \leq \frac{1}{4!} |\cos^{(4)}(\xi)| \, |(x-x_{k-1})(x-x_k)(x-x_{k+1}) \cdot (x-x_{k+2})|$$

$$\leq \frac{1}{24} \cdot 1 \cdot 2h \cdot h \cdot h \cdot 2h$$

$$= \frac{h^4}{6} \overset{!}{\leq} 10^{-5}$$

$$\Longrightarrow \quad \underline{h \leq 0.088}$$

Aufgabe: Eine vorgegebene stetige Funktion $f: [a,b] \to \mathbb{R}$ soll unter gegebenen Genauigkeitsanforderungen durch ein Polynom P approximiert werden:

$$\max_{a \leq x \leq b} |f(x) - P(x)| \overset{!}{\leq} \varepsilon, \quad \varepsilon > 0 \text{ gegeben}$$

Daß diese Aufgabe stets lösbar ist, besagt ein grundlegender Satz der Approximations-theorie, der auf Weierstraß[*)] zurückgeht
(→ Heuser: Analysis 2. Teil)

Andererseits ist die naheliegende Idee, ein solches Polynom durch Interpolation zu bestimmen, häufig nicht erfolgreich – selbst dann nicht, wenn man von äquidistanter Knotenwahl abweicht!
(Vgl. Satz von Faber in Stoer: Numerik I)

Beispiel (3.14) (Runge)

$$f(x) := \frac{1}{1+x^2}, \quad -5 \le x \le 5$$

Bei äquidistanter Knotenwahl

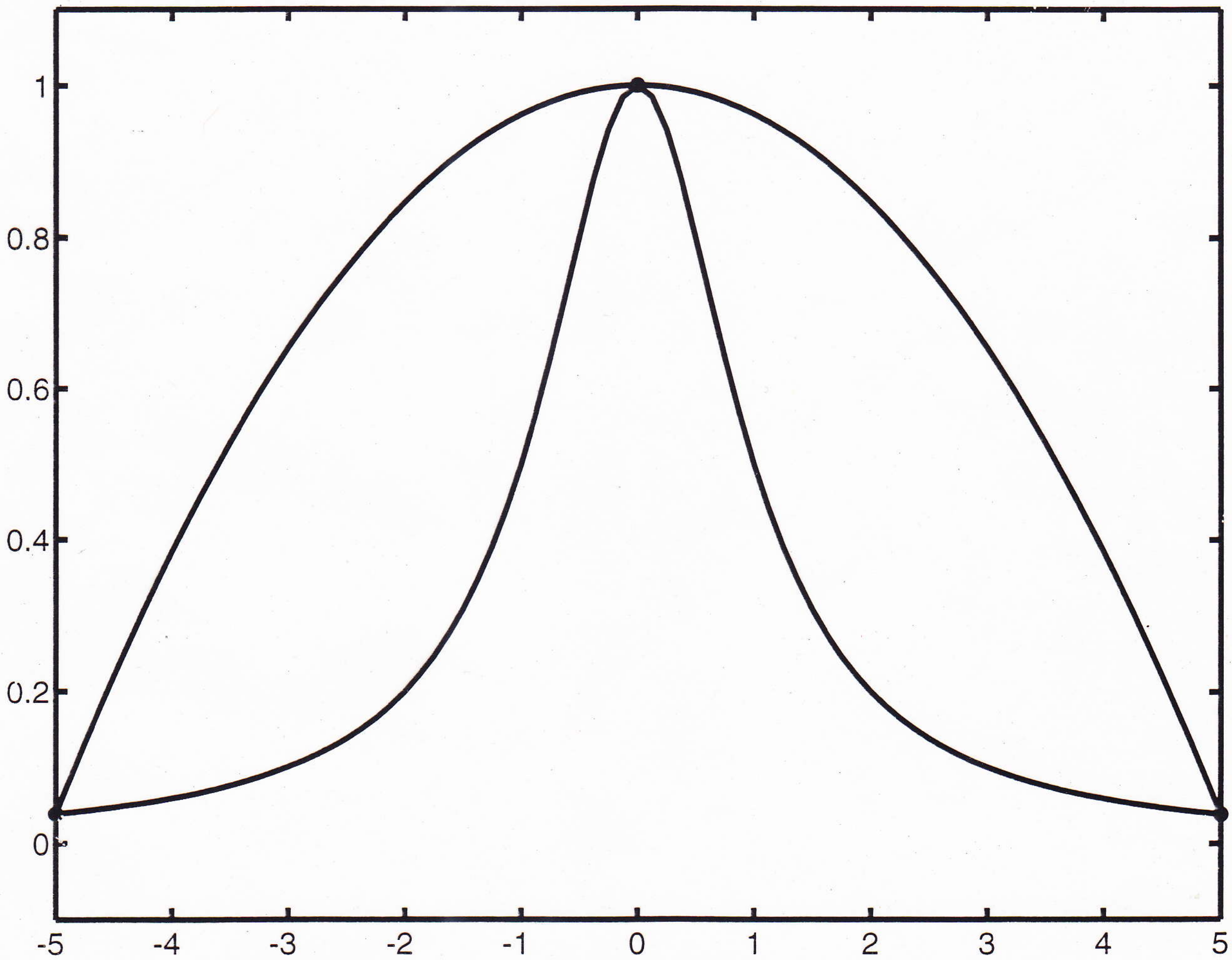
$$x_k^{(n)} = -5 + k\,\frac{10}{n}, \quad k = 0, 1, \ldots, n$$

wächst der Interpolationsfehler mit wachsendem n. Grund: Anwachsen von $\omega_{n+1}(x)$ in (3.12).

---

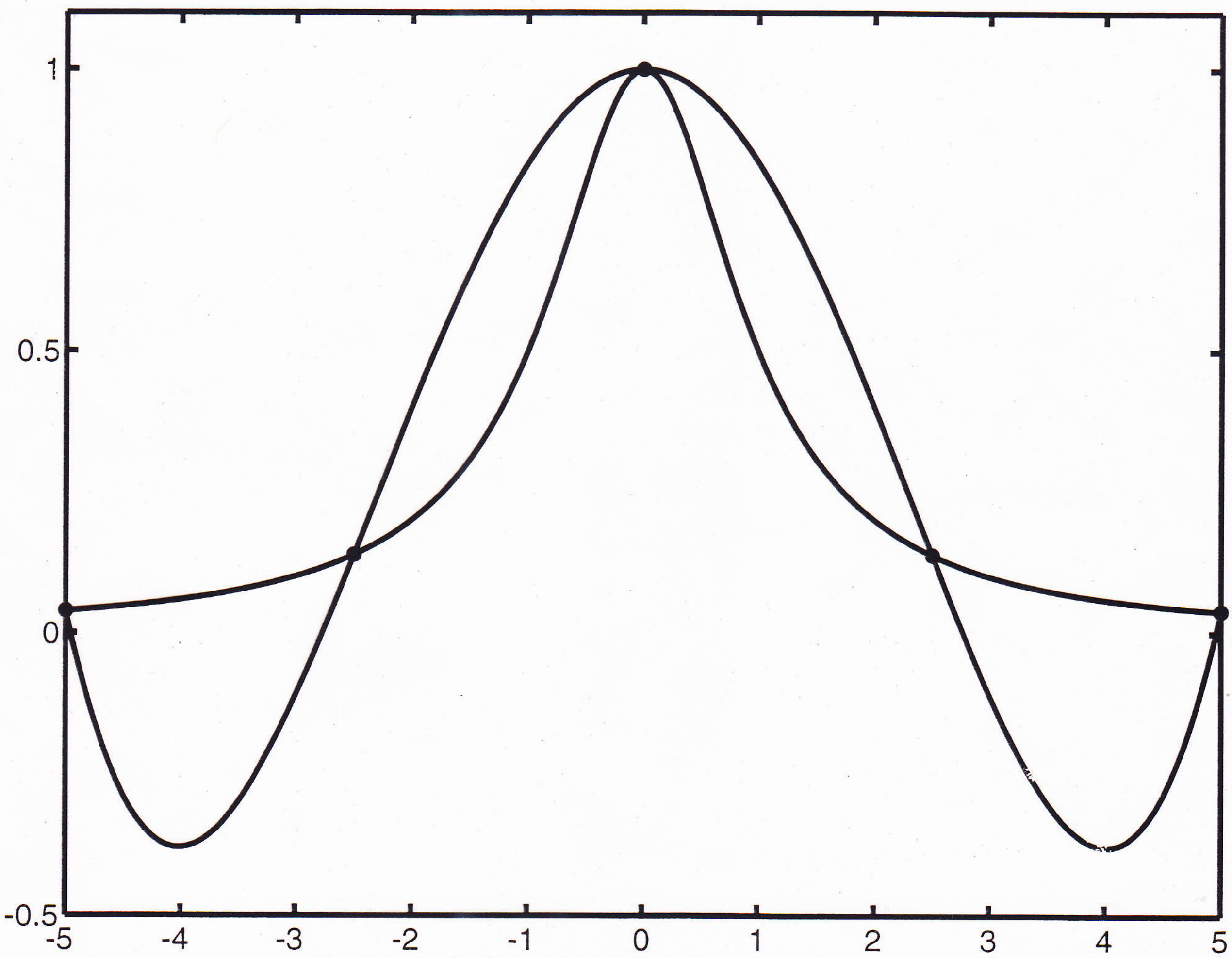
*) Karl Weierstraß (1815 – 1897) Münster, Braunschweig, Berlin

2.5a



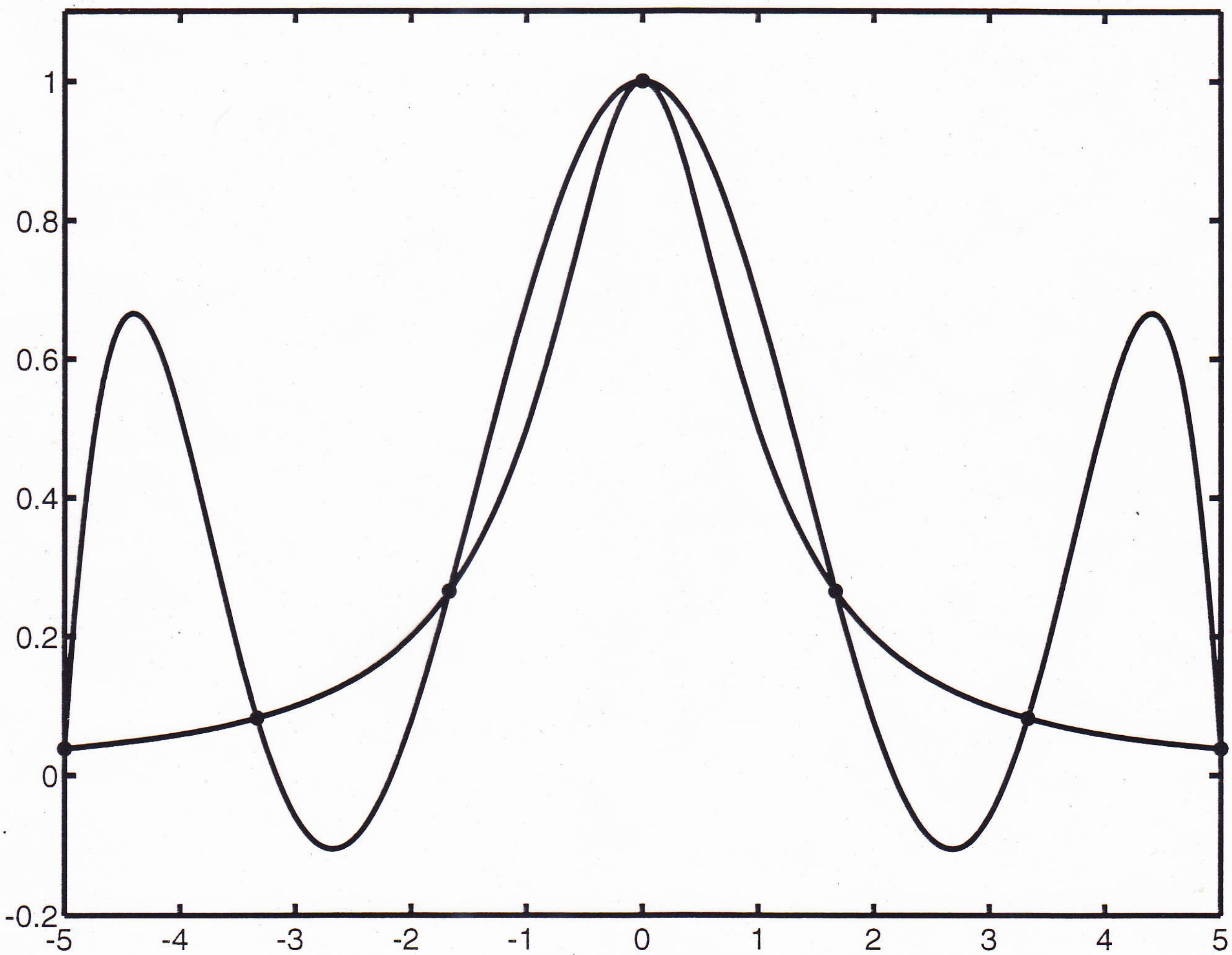
Beispiel von Runge; aequidistante Knoten, n = 2

2.5b



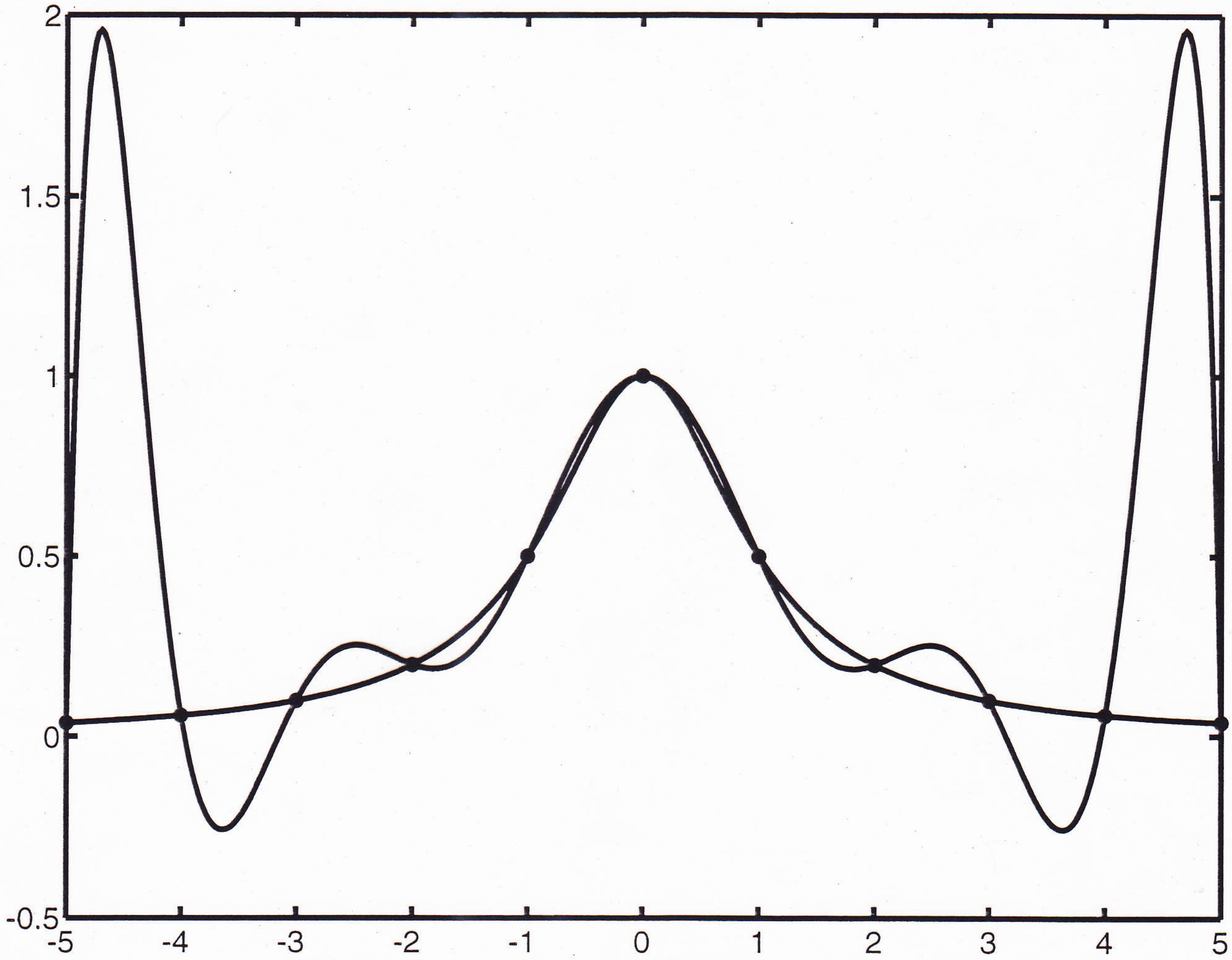
Beispiel von Runge; aequidistante Knoten, n = 4

2.5c



Beispiel von Runge; aequidistante Knoten, n = 6

2.5d



Beispiel von Runge; aequidistante Knoten, n = 10

3.5 Tschebyscheff-Polynome*)

Idee: Zu einem vorgegebenen Intervall $[a,b]$ und $n \in \mathbb{N}$ sollen Interpolationsknoten

$$a \leq x_0 < x_1 < \ldots < x_n \leq b$$

bestimmt werden, so dass das Knotenpolynom

$\omega_{n+1}(x) := \prod_{k=0}^{n} (x - x_k)$ möglichst klein wird:

$$\max_{a \leq x \leq b} |\omega_{n+1}(x)| \quad \text{minimal !}$$

Warnung: Hierdurch wird nicht der Interpol.-fehler $|f(x) - P(x)|$ minimiert; vgl. (3.12), sondern nur der Teil $|\prod (x - x_k)|$.

Reduktion: O.E.d.A. kann $[a,b] = [-1,1]$ angenommen werden. Sind $(x_0, .., x_n)$ optimal bzgl. $[-1,1]$, so erhält man optimale Knoten bzgl. $[a,b]$ durch lineare Transformation:

$$\bar{x}_k := \frac{b-a}{2} x_k + \frac{a+b}{2} . \qquad \underline{(3.15)}$$

---

*) Pafnuti Lwowitsch Tschebyscheff (1821-1894)
Moskau, St. Petersburg

Def. (3.16) Die Funktionen

$$T_n(x) := \cos\left[n \cdot \arccos x\right] \quad , -1 \leq x \leq 1$$

heißen **Tschebyscheff-Polynome (erster Art).**

Satz (3.17) (Eigenschaften der $T_n(x)$)

a) Dreiterm-Rekursion:

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

$$T_0(x) := 1 \quad , \quad T_1(x) := x \quad .$$

Insbesondere folgt hieraus: $T_n \in \Pi_n$

b) Normierung:

$$T_n(x) = 2^{n-1} \cdot x^n + \ldots$$

c) Nullstellen von $T_n(x)$:

$$x_k = \cos\left[\frac{1+2(n-1-k)}{2n}\pi\right] \quad , \quad k=0,1,\ldots,n-1$$

mit $-1 < x_0 < x_1 < \ldots < x_{n-1} < 1$

d) Extrema von $T_n(x)$ auf $[-1,1]$:

$$x_k^E = \cos\left[\frac{n-k}{n}\pi\right] \quad , \quad k=0,1,\ldots,n$$

mit $-1 = x_0^E < x_1^E < \ldots < x_n^E = 1$

und $T_n(x_k^E) = (-1)^{n-k} \quad , \quad k=0,1,\ldots,n$

Beweis:

zu a): Additionstheorem:

$$\varphi := \arccos x \iff x = \cos\varphi$$

$$
\begin{aligned}
T_{n+1}(x) &= \cos[(n+1)\varphi] \\
&= \cos\varphi\cos(n\varphi) - \sin\varphi\sin(n\varphi) \\
&= 2\cos\varphi\cos(n\varphi) - [\cos\varphi\cos n\varphi + \sin\varphi\sin n\varphi] \\
&= 2\cos\varphi\cos(n\varphi) - \cos[(n-1)\varphi] \\
&= 2x\,T_n(x) - T_{n-1}(x)
\end{aligned}
$$

zu b): Aus a) per vollst. Induktion

zu c): $T_n(x) = 0 \iff n\varphi = (2j+1)\frac{\pi}{2},\ j \in \mathbb{Z}$

$$\iff x = \cos\left(\frac{2j+1}{2n}\pi\right),\quad j = 0, 1, \dots, n-1$$

zu d): $T_n'(x) = -\sin(n\varphi)\cdot\frac{n}{\sqrt{1-x^2}},\quad -1 < x < 1$

$$T_n'(x) = 0 \iff n\varphi = j\pi \iff x = \cos\left(\frac{j\pi}{n}\right)$$

▲

Folgerung (3.18) (Minimax-Eigenschaft)

Das Polynom $\frac{1}{2^n}T_{n+1}(x)$ minimiert

$$\|p\|_\infty := \max_{-1\le x\le 1} |p(x)|$$ für alle normierten

Polynome $p(x) = x^{n+1} + \dots \in \Pi_{n+1}$.

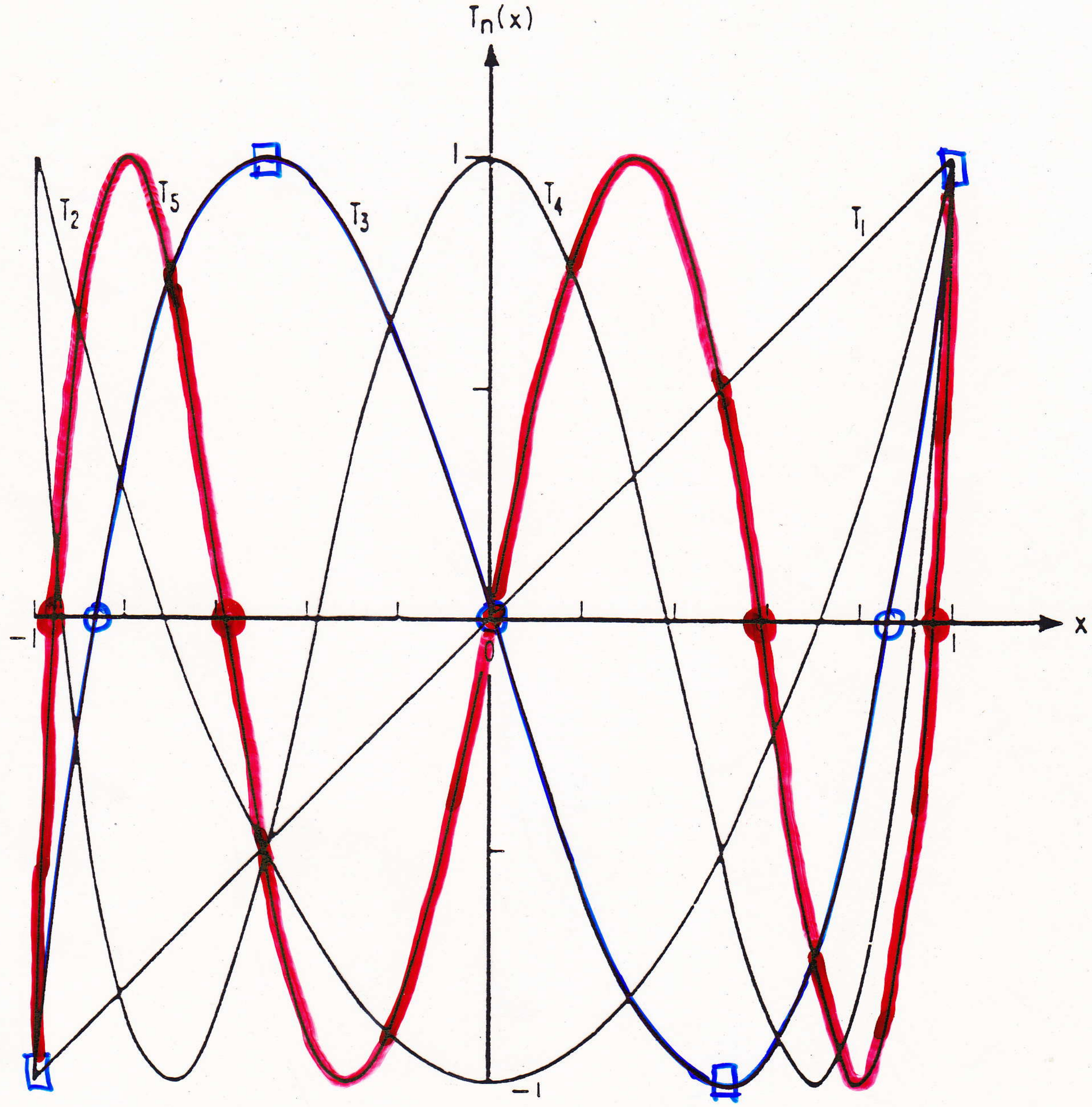


FIGURE 22.6. *Chebyshev Polynomials* $T_n(x)$, $n=1(1)5$.

Beweis: Nach (3.17) hat $p^*(x) := \frac{1}{2^n} T_{n+1}(x)$ auf $[-1,1]$ $(n+2)$ Extrema:

$$-1 = x_0^E < x_1^E < \ldots < x_{n+1}^E = 1$$

mit $p^*(x_k^E) = (-1)^{n+1-k} \cdot \frac{1}{2^n}$, $k = 0, 1, \ldots, n+1$

Gäbe es nun ein normiertes Polynom $p(x) = x^{n+1} + \ldots$ mit $\|p\|_\infty < \frac{1}{2^n}$, so wäre $q := p^* - p \in \Pi_n$ (x^{n+1} hebt sich weg)

mit $q(x_k^E) \begin{cases} > 0, & \text{falls } p^*(x_k^E) = \frac{1}{2^n} \\ < 0, & \text{falls } p^*(x_k^E) = -\frac{1}{2^n} \end{cases}$

Nach dem Zwischenwertsatz hat q daher wenigstens $(n+1)$ Nullstellen $\Longrightarrow$ $q = 0$ im Widerspruch zu $\|p\|_\infty < \|p^*\|_\infty$ ▲

Folgerung (3.19) (T-Knoten)

Die gesuchten Interpolationsknoten sind gerade die Nullstellen von $T_{n+1}(x)$, also

$$x_k = \cos\left[\frac{1+2(n-k)}{2(n+1)}\pi\right], \quad k = 0, 1, \ldots, n$$

wobei $-1 < x_0 < x_1 < \ldots < x_n < 1$.

Beispiel (3.14) (Runge)

$$f(x) := \frac{1}{1+x^2} \quad , -5 \le x \le 5$$

Diesmal Tschebyscheff-Knoten:

$$x_k^{(n)} = 5 \cdot \cos\left[\frac{1+2(n-k)}{2(n+1)} \pi \right]$$

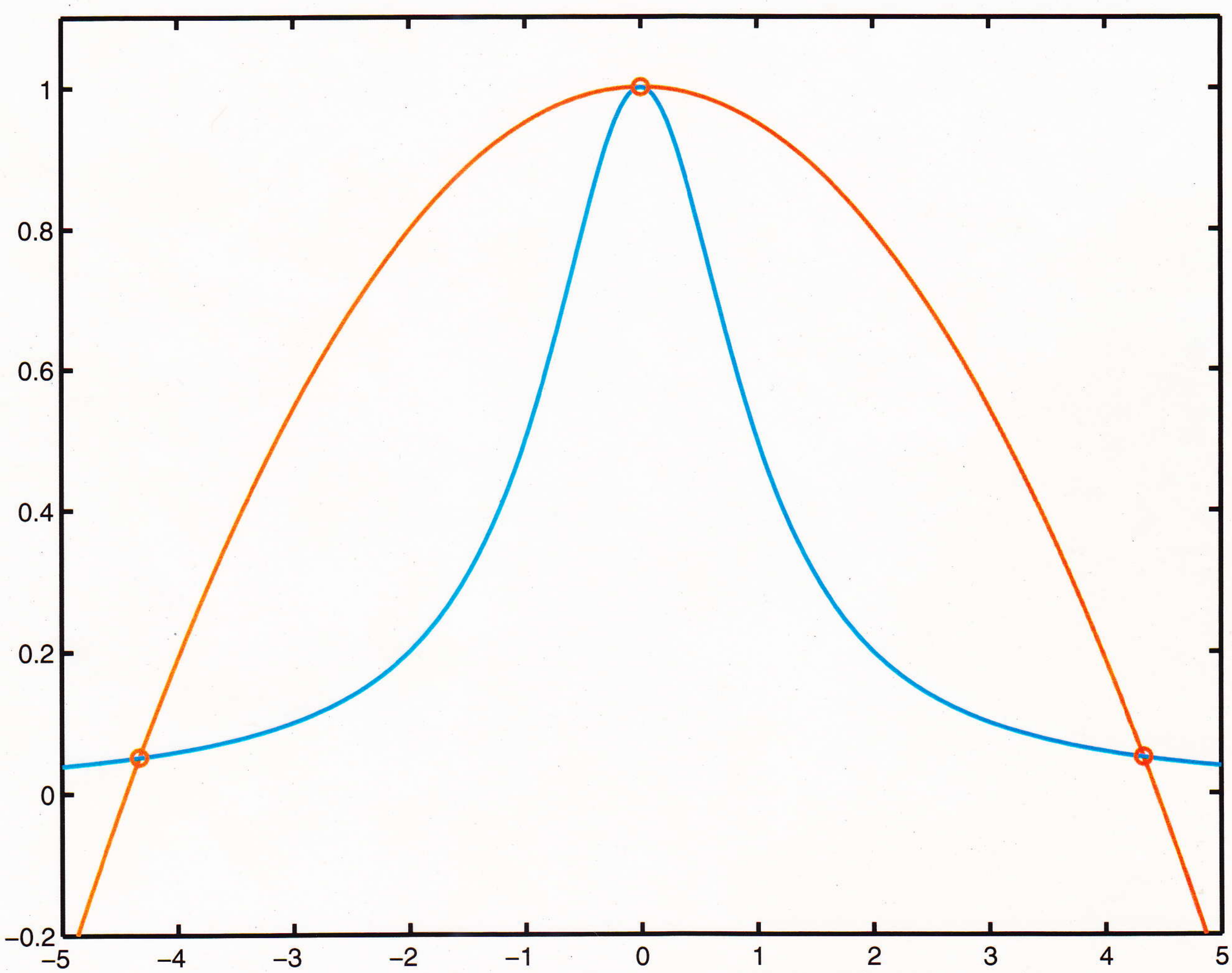
$$k = 0, 1, \ldots, n$$

$\Rightarrow$ Erheblich bessere Approximationsgüte für große n.

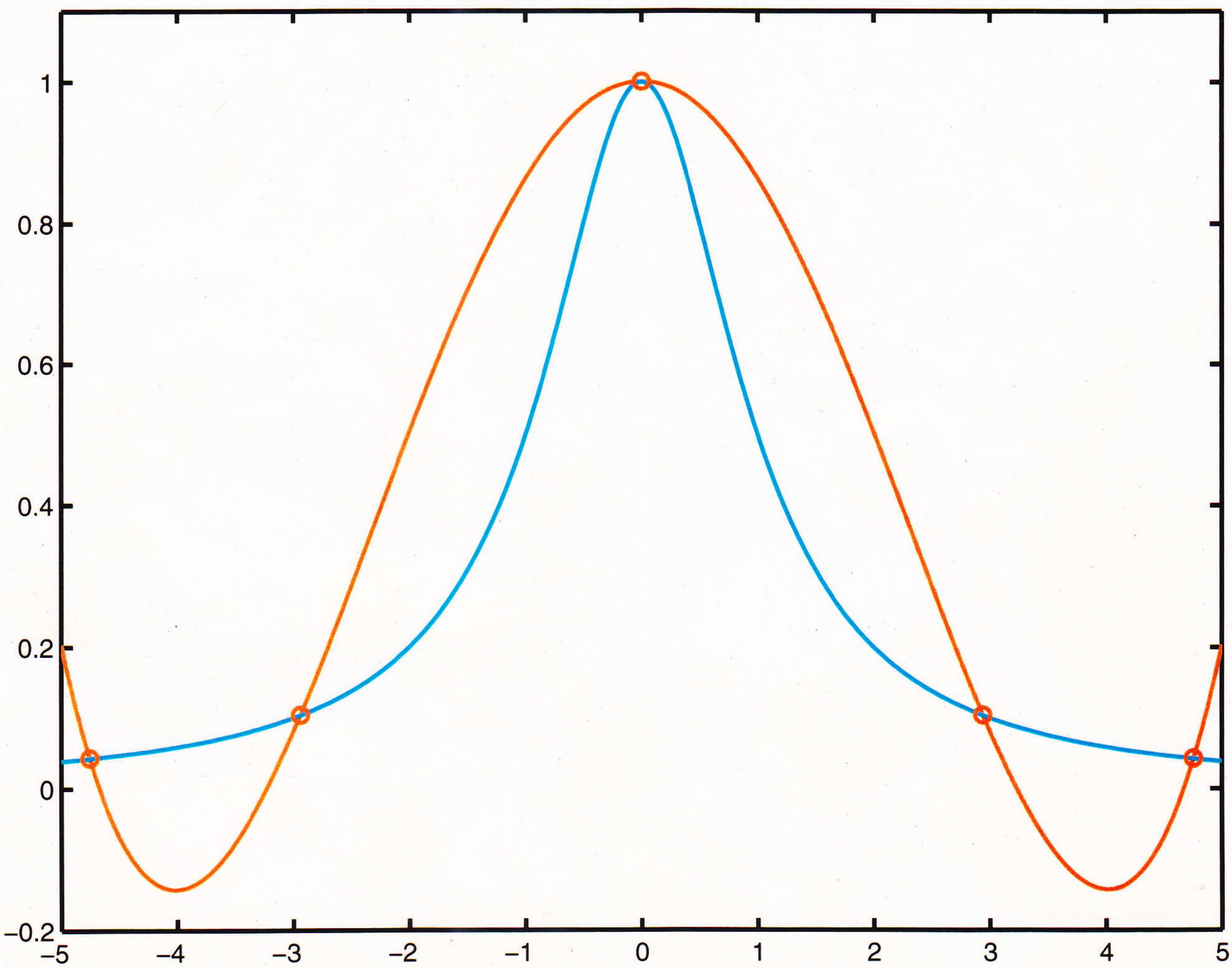
Warnung: Auch für T-Knoten kann nicht sichergestellt werden, dass

$$\| f - p_n \|_\infty := \max_{a \le x \le b} | f(x) - p_n(x) |$$

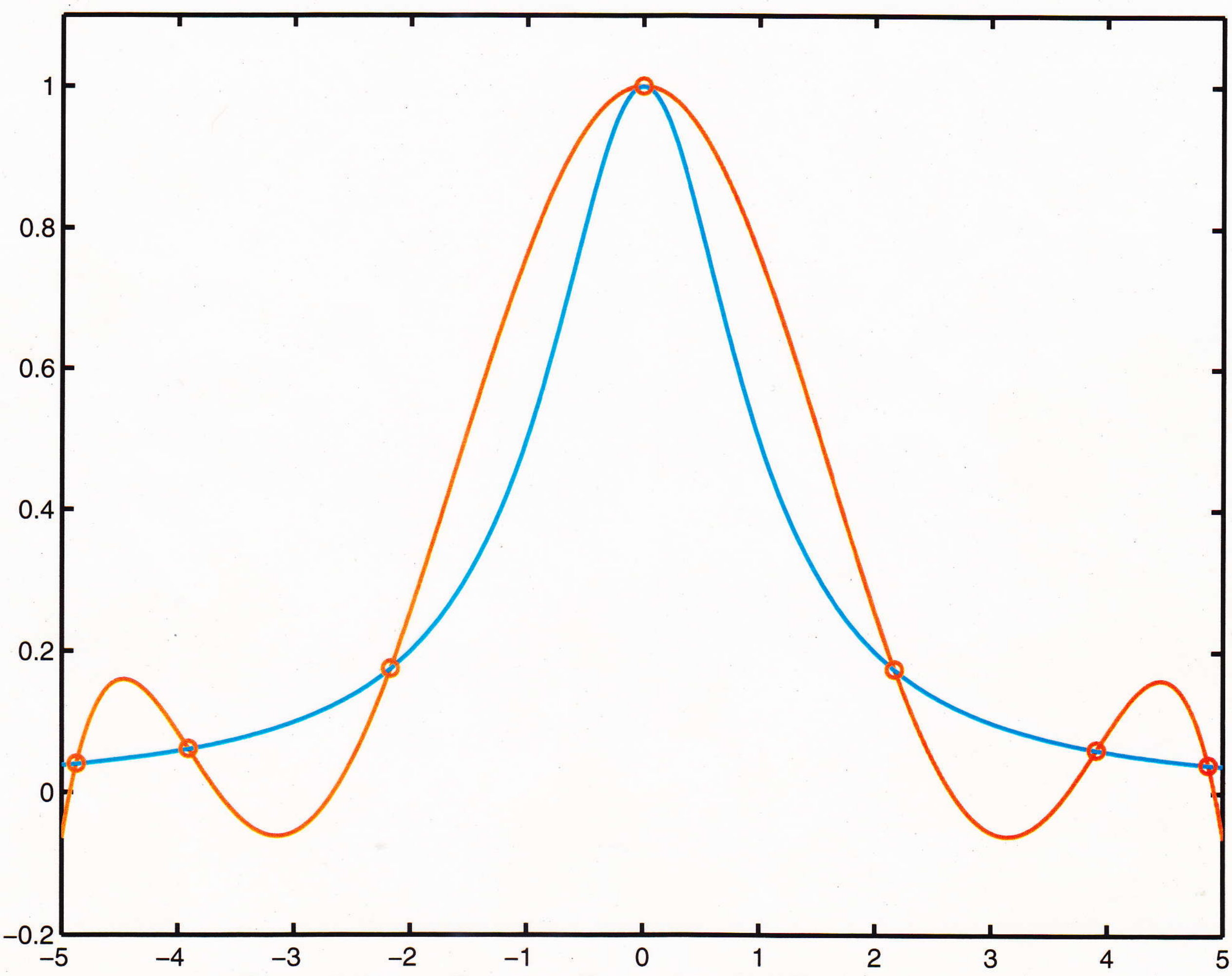
für $n \to \infty$ gegen Null geht!!



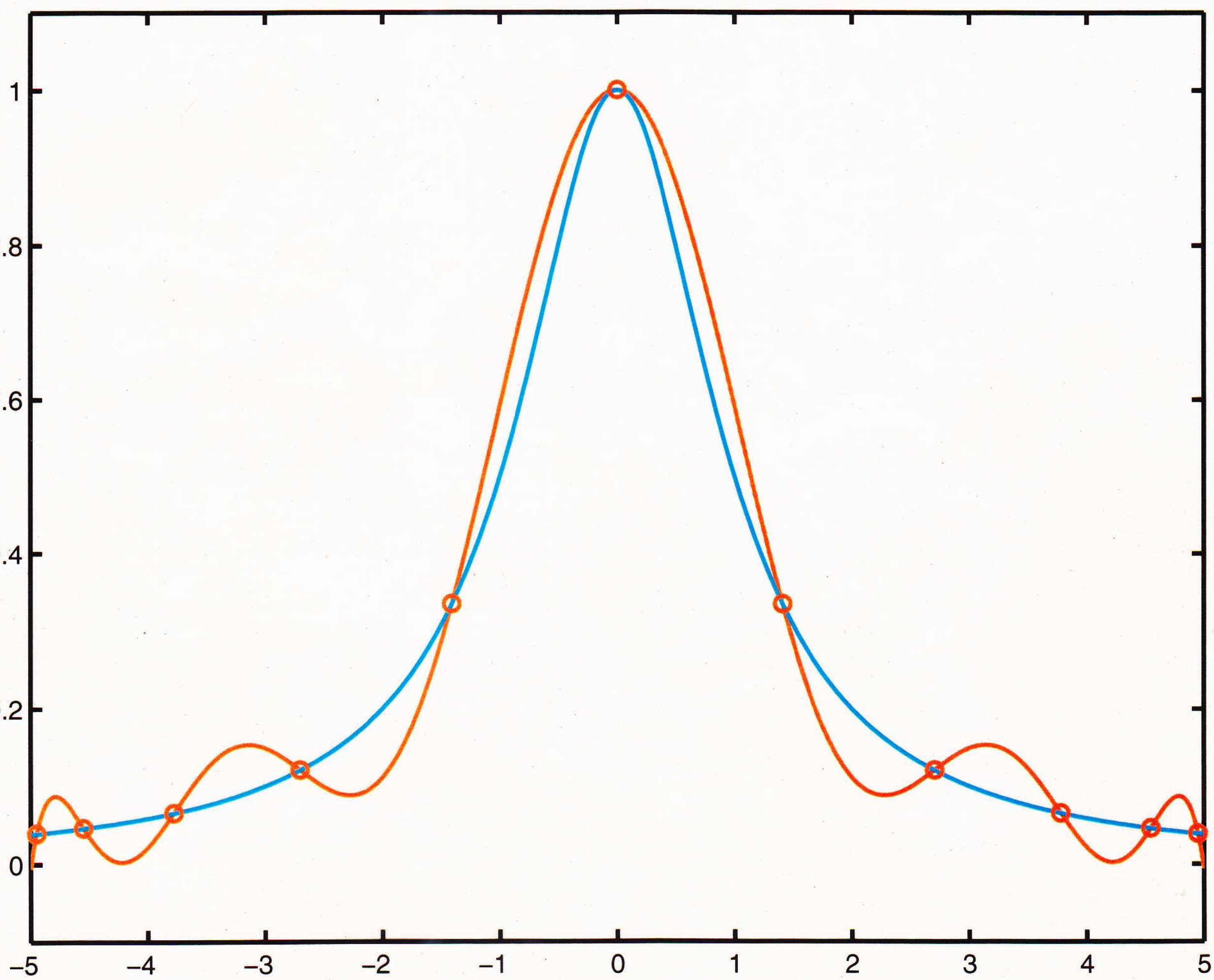
Beispiel von Runge; Tschebyscheff–Knoten, n = 2



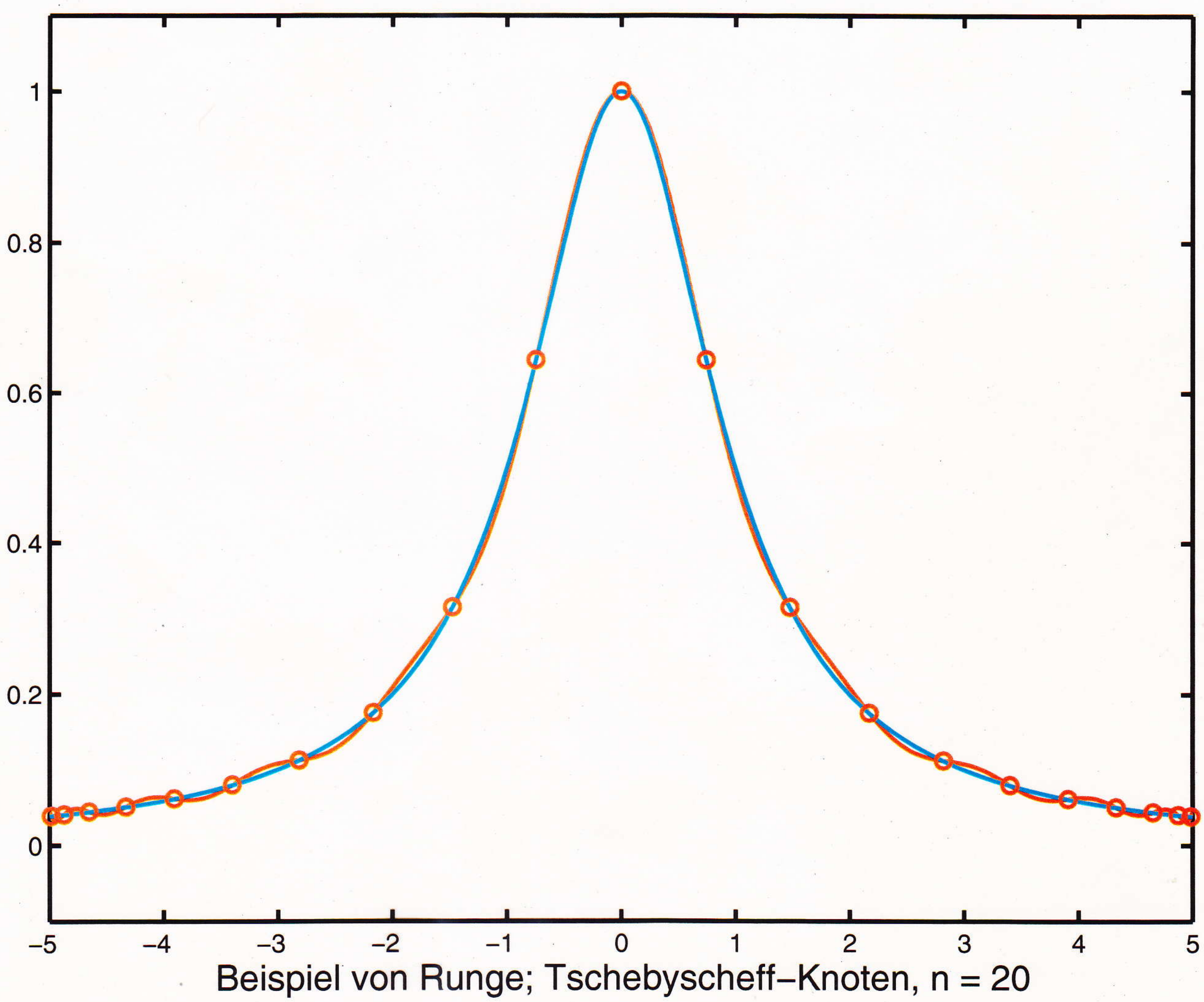
Beispiel von Runge; Tschebyscheff−Knoten, n = 4



Beispiel von Runge; Tschebyscheff−Knoten, n = 6



Beispiel von Runge; Tschebyscheff−Knoten, n = 10



Beispiel von Runge; Tschebyscheff–Knoten, n = 20

3.6 Hermite-Interpolation *)

Idee: Neben den Funktionswerten sollen auch die ersten Ableitungen interpoliert werden!

Problemstellung: Gegeben seien $n \in \mathbb{N}$,
$x_0, .., x_n \in \mathbb{R}$ paarw. verschiedene Knoten,
Daten: $f_0, .., f_n, f_0', .., f_n'$.

Gesucht: $p \in \Pi_{2n+1}$ mit
$\forall i: \quad p(x_i) = f_i \wedge p'(x_i) = f_i'$.

Satz (3.20) (Existenz + Eindeutigkeit)

a) Mit den Lagrange-Polynomen $L_k(x)$ gilt:

$$p(x) := \sum_{k=0}^{n} (1 - 2L_k'(x_k)(x-x_k))\, L_k^2(x)\, f_k + \sum_{k=0}^{n} (x-x_k)\, L_k^2(x)\, f_k'$$

löst das Hermite-Interpolationsproblem

b) Die Lösung ist eindeutig bestimmt.

---

*) Charles Hermite (1822-1901) Paris

Beweis:

zu a): Es ist tatsächlich $p \in \Pi_{2n+1}$.
Die Interpol. bedingungen sind nachzurechnen!

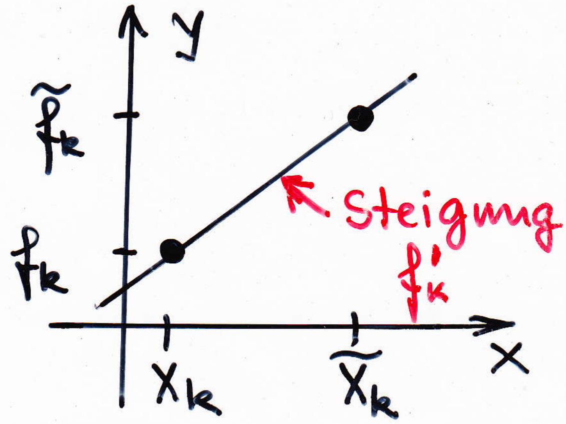
zu b): Gäbe es ein zweites Polynom $q \in \Pi_{2n+1}$, das die Interpol. bedingungen erfüllt, so hätte $r(x) := p(x) - q(x)$ in allen x_k eine doppelte Nullstelle! Damit hätte r $(2n+2)$ Nullstellen (der Vielfachheit nach gezählt).
Andererseits: $r \in \Pi_{2n+1}$. Damit: $r = 0$, also $p = q$. ▲

Zur numerischen Berechnung des Hermite-Interpolationspolynoms ist die Lagrange-Darstellung (3.20) nicht geeignet. Statt dessen wird man versuchen, eine **Newton-Darstellung** analog zu (3.8) zu finden.

Idee: Man wähle zu jedem Knoten x_k einen weiteren benachbarten Knoten $\tilde{x}_k$ und wähle den Funktionswert $\tilde{f}_k$ so, dass die

Gerade durch (x_k, f_k), $(\tilde{x}_k, \tilde{f}_k)$ gerade die Steigung f_k' besitzt, also

$$f[x_k, \tilde{x}_k] = \frac{\tilde{f}_k - f_k}{\tilde{x}_k - x_k} = f_k'$$



Das zugehörige Interpol.-polynom liegt in Π_{2n+1}.
Für $\tilde{x}_k \to x_k$ erfüllt das "Grenzpolynom" dann die Hermite-Interpol.bedingungen.

Newton-Schema:

$$\begin{array}{lllll}
f[x_0] & & & & \\
f[x_0] & f_0' & & & \\
f[x_1] & f[x_0,x_1] & f[x_0,x_0,x_1] & & \\
f[x_1] & f_1' & f[x_0,x_1,x_1] & \ddots & \\
\vdots & \vdots & \vdots & & \\
f[x_n] & f[x_{n-1},x_n] & f[x_{n-1},x_{n-1},x_n] & & \\
f[x_n] & f_n' & f[x_{n-1},x_n,x_n] & \cdots & f[x_0,x_0,\ldots x_n x_n]
\end{array}$$

$$P(x) = f_0 + f_0'(x-x_0) + f[x_0,x_0,x_1](x-x_0)^2 + \ldots + f[x_0,x_0,\ldots,x_n x_n](x-x_0)^2 \ldots (x-x_{n-1})^2 (x-x_n)$$

Beispiel (3.21)

x_i	0	1	2
f_i	0	1	0
f_i'	1	0	-1

Dividierte Differenzen:

0	0					
0	0	1				
1	1	1	0			
1	1	0	-1	-1		
2	0	-1	-1	0	½	
2	0	-1	0	1	½	0

$$P(x) = 0 + 1\cdot x + 0\cdot x^2 - 1\cdot x^2(x-1) + \tfrac{1}{2} x^2(x-1)^2 + 0\cdot x^2(x-1)^2(x-2)$$

$$= \underline{x - x^2(x-1) + \tfrac{1}{2} x^2 (x-1)^2}$$