6 Interpolation und numerische Approximation

6.1 Polynominterpolation

Das (allgemeine) Interpolationsproblem:

Zu gegebener Funktion $f : [a, b] \to \mathbb{C}$ und gegebenen Stützstellen (Knoten) $a \leq x_0 < x_1 < x_2 < \cdots < x_n \leq b$ soll eine „einfache“ Funktion $p : [a, b] \to \mathbb{C}$ konstruiert werden, die die Interpolationsbedingungen $p(x_i) = f(x_i)\,(i = 0, 1, \ldots, n)$ erfüllt.

Wozu?

- f ist nur an diskreten Punkten bekannt (Messwerte), aber eine geschlossene Formel für f ist auf ganz $[a, b]$ erwünscht (z.B. um f an Zwischenstellen $x \in [a, b] \setminus \{x_0, x_1, \ldots, x_n\}$ auszuwerten),
- f ist „kompliziert“ und soll durch eine „einfache“ Funktion angenähert werden (z.B. um die Ableitung $f'(x)$, $x \in [a, b]$, oder das Integral $\int_a^b f(x)dx$ näherungsweise zu bestimmen).

Das polynomiale Interpolationsproblem:

Zu gegebenen (paarweise verschiedenen) Knoten

$$a \leq x_0 < x_1 < x_2 < \cdots < x_n \leq b$$

und gegebenen Funktionswerten $\{f_i\}_{i=0}^n \in \mathbb{C}$ soll ein Interpolationspolynom

$$p(x) = c_n x^n + c_{n-1} x^{n-1} + \cdots + c_1 x + c_0 \in \mathscr{P}_n$$

(mit komplexen Koeffizienten $c_0, c_1, \ldots, c_n$, d.h. $n+1$ Freiheitsgrade) vom Grad n konstruiert werden, das die $n+1$ Interpolationsbedingungen

$$p(x_i) = f_i, \qquad i = 0, 1, \ldots, n,$$

erfüllt.

Satz 6.1. *Die polynomiale Interpolationsaufgabe ist eindeutig lösbar. Mit den Lagrange-Grundpolynomen [JOSEPH LOUIS LAGRANGE (1736–1813)]*

$$\ell_i(x) := \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j} \in \mathscr{P}_n$$

(beachte $\ell_i(x_i) = 1$ und $\ell_i(x_j) = 0$ für $j \neq i$) lässt sich das Interpolationspolynom in der Lagrange-Form

$$p(x) = \sum_{i=0}^{n} f_i \ell_i(x)$$

darstellen.

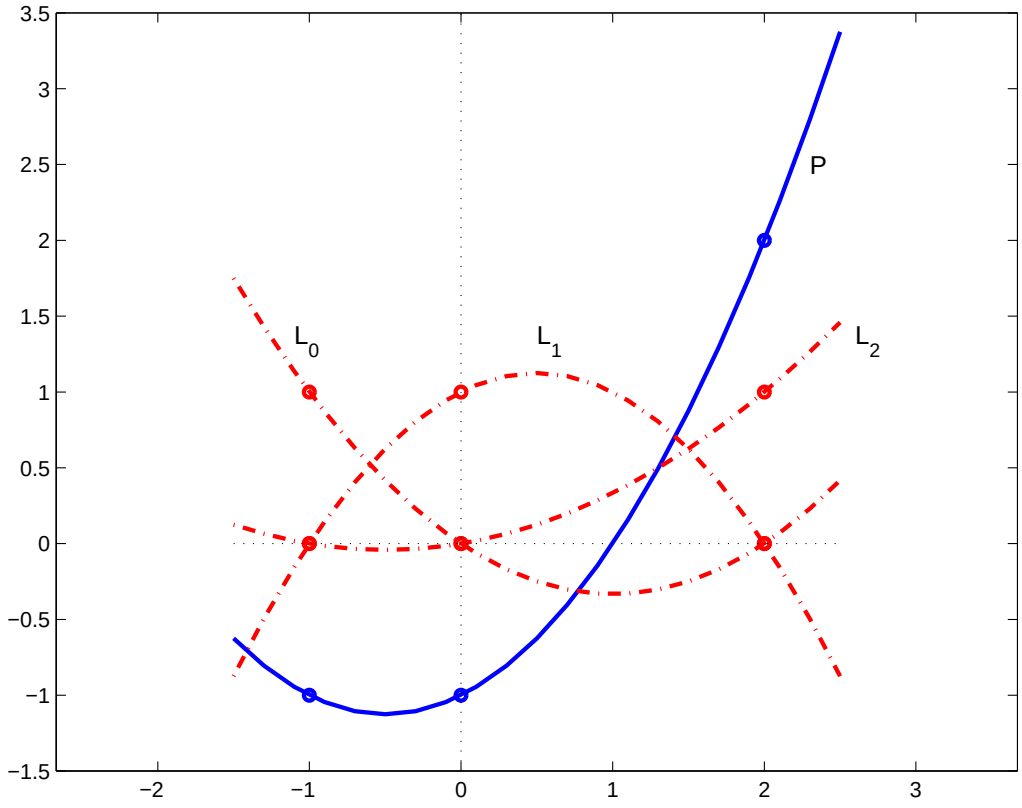
Beispiel 1. Daten: $(x_0, f_0) = (-1, -1)$, $(x_1, f_1) = (0, -1)$, $(x_2, f_2) = (2, 2)$.

Lagrange-Grundpolynome:

$$\ell_0(x) = x(x-2)/3,$$
$$\ell_1(x) = (x+1)(x-2)/(-2),$$
$$\ell_2(x) = (x+1)x/6.$$

Interpolationspolynom:

$$p(x) = -\ell_0(x) - \ell_1(x) + 2\ell_2(x)$$
$$= x^2/2 + x/2 - 1.$$



Die Auswertung der Lagrange-Formel ist aufwendig, wenn ein neues Datenpaar hinzukommt. Eine rekursive Berechnung ist ökonomischer:

Lemma 6.2. *Für eine beliebige Indexmenge* $0 \leq i_0 < i_1 < \cdots < i_k \leq n$ *bezeichne* $p_{i_0,i_1,\ldots,i_k}$ *das (nach Satz 6.1 eindeutig bestimmte) Polynom vom Grad* k*, das die Bedingungen*

$$p_{i_0,i_1,\ldots,i_k}(x_{i_j}) = f_{i_j} \qquad (j = 0, 1, \ldots, k)$$

erfüllt. Dann gilt die Rekursionsformel

$$\begin{aligned} p_i(x) &= f_i, \\ p_{i_0,i_1,\ldots,i_k}(x) &= \frac{(x - x_{i_0})p_{i_1,i_2,\ldots,i_k}(x) - (x - x_{i_k})p_{i_0,i_1,\ldots,i_{k-1}}(x)}{x_{i_k} - x_{i_0}}. \end{aligned}$$

Rechenschema (Algorithmus von Neville-Aitken, [CHARLES WILLIAM NEVILLE ($*$ 1941)]; [ALEXANDER CRAIG AITKEN (1895–1967)]):

x_i	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$
x_0	$p_0(x)=f_0$				
		$p_{0,1}(x)$			
x_1	$p_1(x)=f_1$		$p_{0,1,2}(x)$		
		$p_{1,2}(x)$		$p_{0,1,2,3}(x)$	
x_2	$p_2(x)=f_2$		$p_{1,2,3}(x)$		$p_{0,1,2,3,4}(x)$
		$p_{2,3}(x)$		$p_{1,2,3,4}(x)$	
x_3	$p_3(x)=f_3$		$p_{2,3,4}(x)$		
		$p_{3,4}(x)$			
x_4	$p_4(x)=f_4$				

(Berechnungsreihenfolge : $p_0 \to p_1 \to p_{0,1} \to p_2 \to p_{1,2} \to p_{0,1,2} \to \cdots$)

Beispiel 2 (vgl. Beispiel 1).

x_i	$k=0$	$k=1$	$k=2$
-1	-1		
		$\frac{(x-(-1))(-1)-(x-0)(-1)}{0-(-1)} = -1$	
0	-1		$\frac{(x-(-1))(3x/2-1)-(x-2)(-1)}{2-(-1)}$
		$\frac{(x-0)2-(x-2)(-1)}{2-0} = \frac{3}{2}x - 1$	$= \frac{1}{2}x^2 + \frac{1}{2}x - 1$
2	2		

Aufwand des Neville-Aitken Schemas (für Auswertung des Interpolationspolynoms vom Grad n an einer Stelle x):
$\frac{5}{2}n^2 + \frac{7}{2}n + 1$ Gleitpunktoperationen (falls die Differenzen $x - x_i$ $(0 \leq i \leq n)$ vorab bestimmt werden).

Tableau der dividierten Differenzen von f (vgl. § 4.4):

x_i	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$
x_0	f_0				
		$f_{0,1}$			
x_1	f_1		$f_{0,1,2}$		
		$f_{1,2}$		$f_{0,1,2,3}$	
x_2	f_2		$f_{1,2,3}$		$f_{0,1,2,3,4}$
		$f_{2,3}$		$f_{1,2,3,4}$	
x_3	f_3		$f_{2,3,4}$		
		$f_{3,4}$			
x_4	f_4				

mit

$$f_{i_0,i_1,\ldots,i_k} := \frac{f_{i_1,i_2,\ldots,i_k} - f_{i_0,i_1,\ldots,i_{k-1}}}{x_{i_k} - x_{i_0}} \quad (k \geq 1).$$

Satz 6.3. *(vgl. Satz 4.7 in § 4.4) Mit Hilfe der dividierten Differenzen lässt sich das (nach Satz 6.1 eindeutig bestimmte) Interpolationspolynom p in Newton-Form*

$$\begin{aligned} p(x) &= f_0 + f_{0,1}(x - x_0) + f_{0,1,2}(x - x_0)(x - x_1) + \cdots \\ &\quad \cdots + f_{0,1,\ldots,n}(x - x_0)(x - x_1) \cdots (x - x_{n-1}) \end{aligned}$$

darstellen.

Rechenaufwand:

- Zur Bestimmung der Differenzentafel: $\frac{3}{2}(n^2 + n)$ Gleitpunktoperationen.
- Zur Auswertung des Newtonschen Interpolationspolynoms mit dem Horner-Schema ([WILLIAM GEORGE HORNER (1786–1837)]): $3n$ Gleitpunktoperationen (pro Auswertungspunkt).

Beispiel 3 (vgl. Beispiele 1 und 2).
Dividierte Differenzen:

x_i	$k=0$	$k=1$	$k=2$
-1	$\boxed{-1}$		
		$f_{0,1} = \frac{(-1)-(-1)}{0-(-1)} = \boxed{0}$	
0	-1		$f_{0,1,2} = \frac{3/2-0}{2-(-1)} = \boxed{\frac{1}{2}}$
		$f_{1,2} = \frac{2-(-1)}{2-0} = \frac{3}{2}$	
2	2		

Das bedeutet:

$$p(x) = \boxed{(-1)} + \boxed{0}(x-(-1)) + \boxed{\frac{1}{2}}(x-(-1))(x-0) = \frac{1}{2}x^2 + \frac{1}{2}x - 1.$$

Baryzentrische Interpolationsformeln. Das zu den (paarweise verschiedenen) Interpolationsknoten $\{x_0, x_1, \ldots, x_n\}$ gehörende Knotenpolynom sei definiert durch

$$\omega_{n+1}(x) := (x - x_0)(x - x_1) \cdots (x - x_n) \in \mathscr{P}_{n+1}.$$

Definiert man die baryzentrischen Gewichte $\{w_j\}_{j=0}^n$ durch

$$w_j := \frac{1}{\prod_{\substack{k=0 \\ k \neq j}}^n (x_j - x_k)} = \frac{1}{\omega'_{n+1}(x_j)}, \qquad j = 0, \ldots, n, \tag{6.1}$$

so gilt für die Lagrange Grundpolynome

$$\ell_j(x) = \omega_{n+1}(x) \frac{w_j}{x - x_j}, \qquad j = 0, \ldots, n,$$

und hiermit lässt sich das Interpolationspolynom darstellen durch die ...

erste baryzentrische Formel

$$p(x) = \omega_{n+1}(x) \sum_{j=0}^{n} f_j \frac{w_j}{x - x_j}.$$

Da die konstante Funktion $f \equiv 1$ exakt interpoliert wird gilt

$$1 = \omega_{n+1}(x) \sum_{j=0}^{n} \frac{w_j}{x - x_j},$$

und somit nach Quotientenbildung und Kürzen die zweite baryzentrische Formel

$$p(x) = \frac{\displaystyle\sum_{j=0}^{n} f_j \frac{w_j}{x - x_j}}{\displaystyle\sum_{j=0}^{n} \frac{w_j}{x - x_j}}.$$

Aufdatierung. Bei Hinzunahme von x_{n+1}

$$w_j^{\text{neu}} := \frac{w_j^{\text{alt}}}{x_j - x_{n+1}}, \qquad j = 0, \ldots, n, \quad (2n+2 \text{ Flops}).$$

w_{n+1} aus (6.1), $n+1$ weitere Flops, falls $x_j - x_{n+1}$ gemerkt werden.

Aufwand.

- Berechnung von $\{w_j\}_{j=0}^n$ erfordert $\sum_{j=1}^n 3j = \frac{3}{2}n(n+1)$ Flops.
- Bei gegebenen Gewichten $\{w_j\}_{j=0}^n$ jede Auswertung von p in weiteren $5n + 4 = O(n)$ Flops.

Weitere Vorteile.

- w_j hängen nicht von den Daten f_j ab, d.h. bei gegebenen Gewichten können beliebige Funktionen f in $O(n)$ Flops interpoliert werden.
- w_j unabhängig von Knotennummerierung (vgl. dividierte Differenzen).

Beispiel. Interpolation an äquidistanten Knoten in $[a, b]$ führt auf

$$w_j = (-1)^j \binom{n}{j} \qquad j = 0, 1, \ldots, n$$

(modulo des gemeinsamen Faktors $\frac{(-1)^n}{n!} \left(\frac{n}{b-a}\right)^n$).

Satz 6.4 (Fehler der Polynominterpolation)**.** *Die Funktion* $f \in C^{n+1}[a,b]$ *werde durch das Polynom* $p \in \mathscr{P}_n$ *interpoliert an den paarweise verschiedenen Knoten* $\{x_0, x_1, \ldots, x_n\} \subset [a,b]$*. Deren Knotenpolynom sei bezeichnet mit*

$$\omega_{n+1}(x) = (x - x_0)(x - x_1) \cdots (x - x_n) \in \mathscr{P}_{n+1}.$$

Dann gibt es zu jedem $x \in [a,b]$ *ein* $\xi = \xi(x) \in (a,b)$ *mit*

$$f(x) - p(x) = \frac{\omega_{n+1}(x)}{(n+1)!}\, f^{(n+1)}(\xi).$$

Mit $M_{n+1} := \max_{a \le t \le b} |f^{(n+1)}(t)|$ *gilt somit für alle* $x \in [a,b]$ *die Fehlerabschätzung*

$$|f(x) - p(x)| \le \frac{M_{n+1}}{(n+1)!} \max_{a \le t \le b} |\omega_{n+1}(t)|. \tag{6.2}$$

Korollar 6.5. *Die Funktion* $f \in C^\infty[a,b]$ *mit*

$$|f^{(n)}(x)| \le M \qquad \forall x \in [a,b],\ \forall n \in \mathbb{N}, \tag{6.3}$$

werde für jedes $n \in \mathbb{N}$ *durch das Polynom* $p_n \in \mathscr{P}_n$ *an der beliebigen Knotenfolge* $\{x_j^{(n)}\}_{j=0}^n \subset [a,b]$ *interpoliert. Dann gilt*

$$\max_{x\in[a,b]} |f(x) - p_n(x)| \to 0 \quad \textit{für} \quad n \to \infty.$$

Die sehr starke Forderung (6.3) ist erfüllt z.B. für e^x, $\sin x$, $\cos x$ und (natürlich) für Polynome. Bereits für die rationale Funktion $f(x) = 1/x$ mit $f^{(n)}(x) = \pm n!/x^{n+1}$ gilt (6.3) etwa auf dem Intervall $[1,2]$ schon nicht mehr.

„Optimale“ Knoten: Idee (motiviert durch Fehlerabschätzung):
Wähle Knoten $a \leq x_0 < x_1 < \cdots < x_n \leq b$ so, dass

$$\max_{a \leq t \leq b} |\omega_{n+1}(t)| = \max_{a \leq t \leq b} \prod_{i=0}^{n} |t - x_i|$$

so klein wie möglich wird.

Lösung: **Tschebyscheff-Knoten** [PAFNUTIĬ L'VOVICH TSCHEBYSCHEFF (1821–1894)]

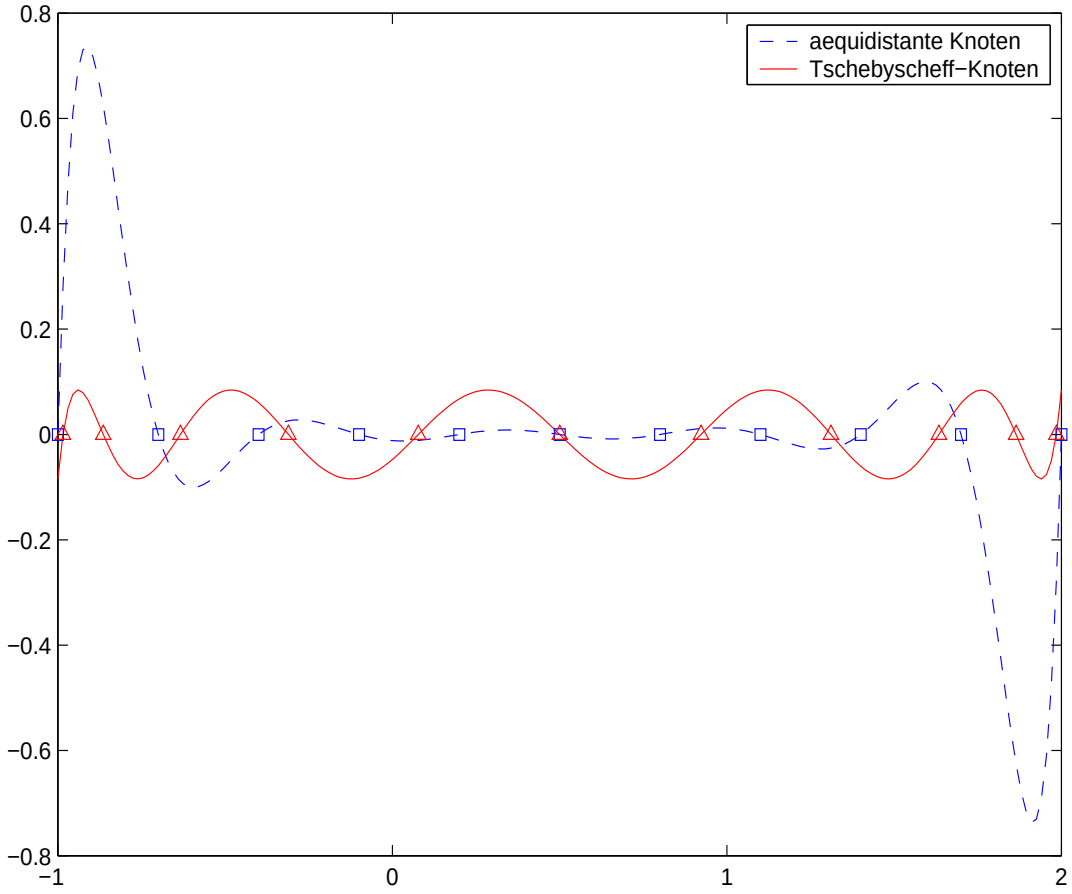
$$x_i^{(\mathrm{T})} = \frac{b-a}{2} \cos\left(\frac{2(n-i)+1}{2n+2}\,\pi\right) + \frac{a+b}{2}, \qquad i = 0, 1, \ldots, n,$$

mit

$$\max_{a \leq t \leq b} \prod_{i=0}^{n} |t - x_i^{(\mathrm{T})}| = 2\left(\frac{b-a}{4}\right)^n < \max_{a \leq t \leq b} \prod_{i=0}^{n} |t - x_i|$$

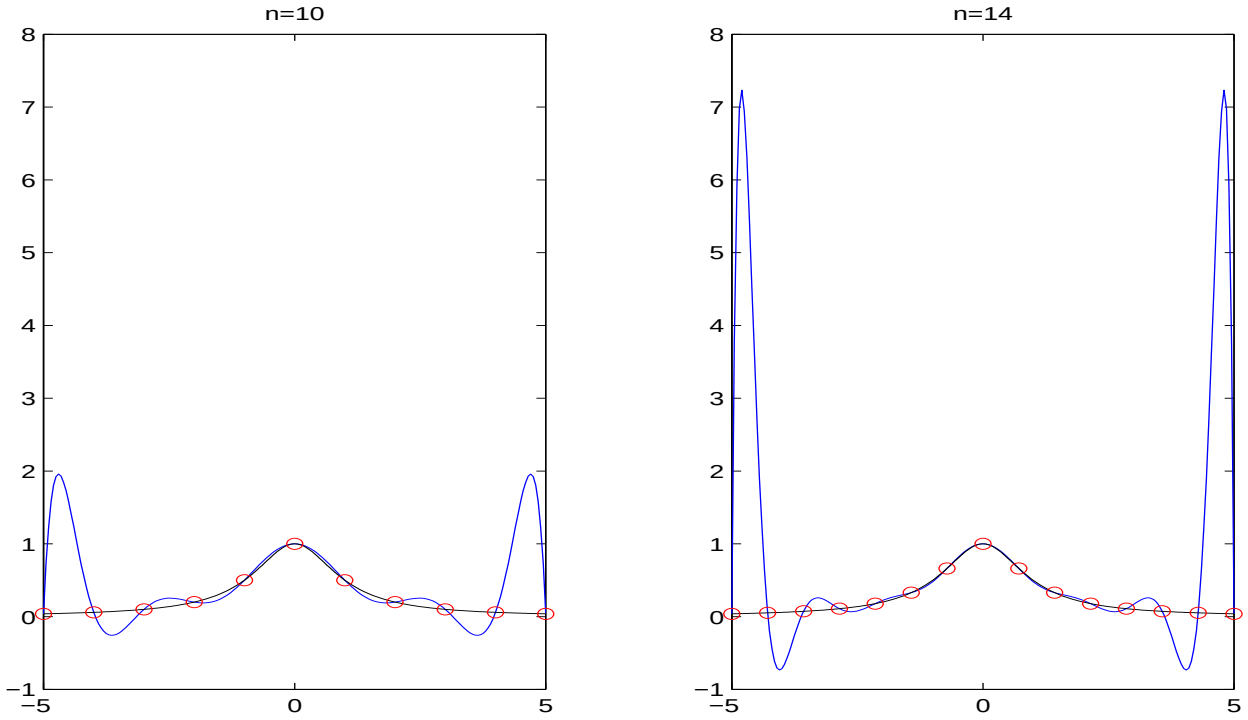
für jede andere Wahl $x_0, \ldots, x_n$ der Knoten.

Knotenpolynome mit äquidistanten und Tschebyscheff-Knoten:



Beispiel 4.(Runge[a]-Phänomen[b]) Interpoliere an $n+1$ äquidistanten Stützstellen

$$f(x) = \frac{1}{1+x^2} \qquad (-5 \le x \le 5) \qquad \text{(Runge-Funktion)}$$

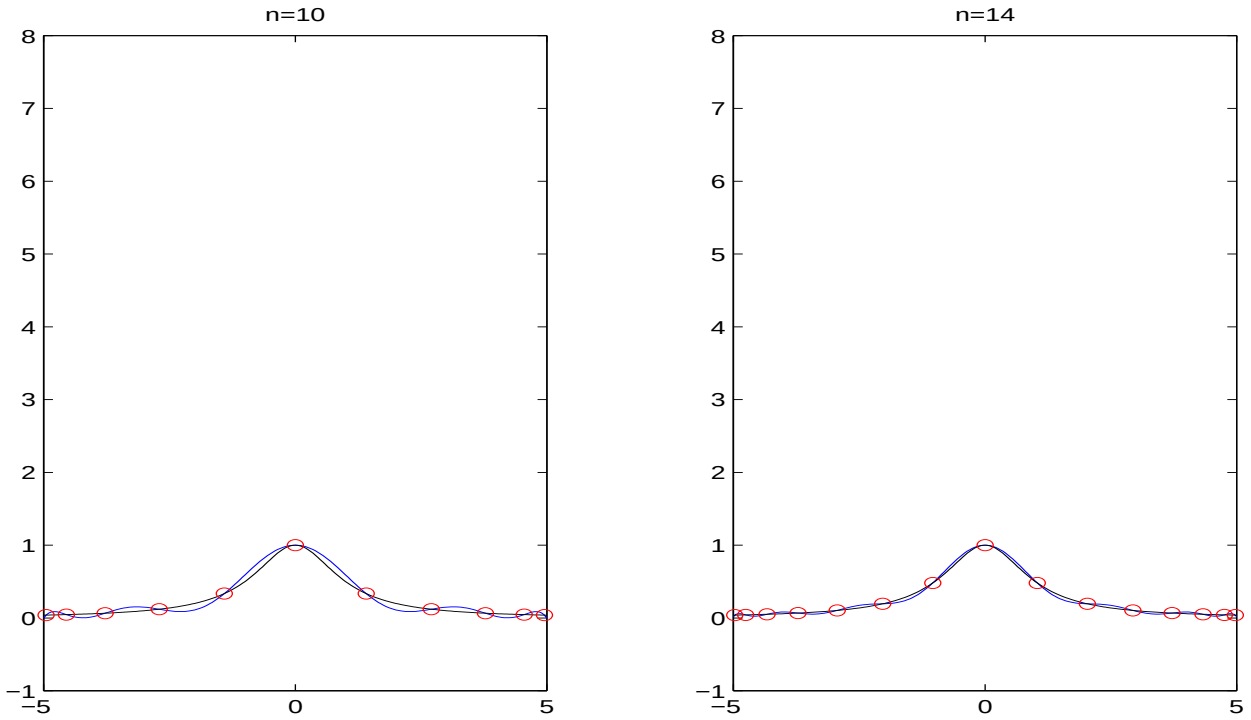


[a][CARL DAVID TOLMÉ RUNGE (1856–1927)].

[b]C. Runge. *Über emprirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten*. Zeitschrift für Mathematik und Physik 46 (1901) pp. 224–243

Beispiel 5. Interpoliere an $n+1$ Tschebyscheff-Knoten

$$f(x) = \frac{1}{1+x^2} \qquad (-5 \le x \le 5).$$

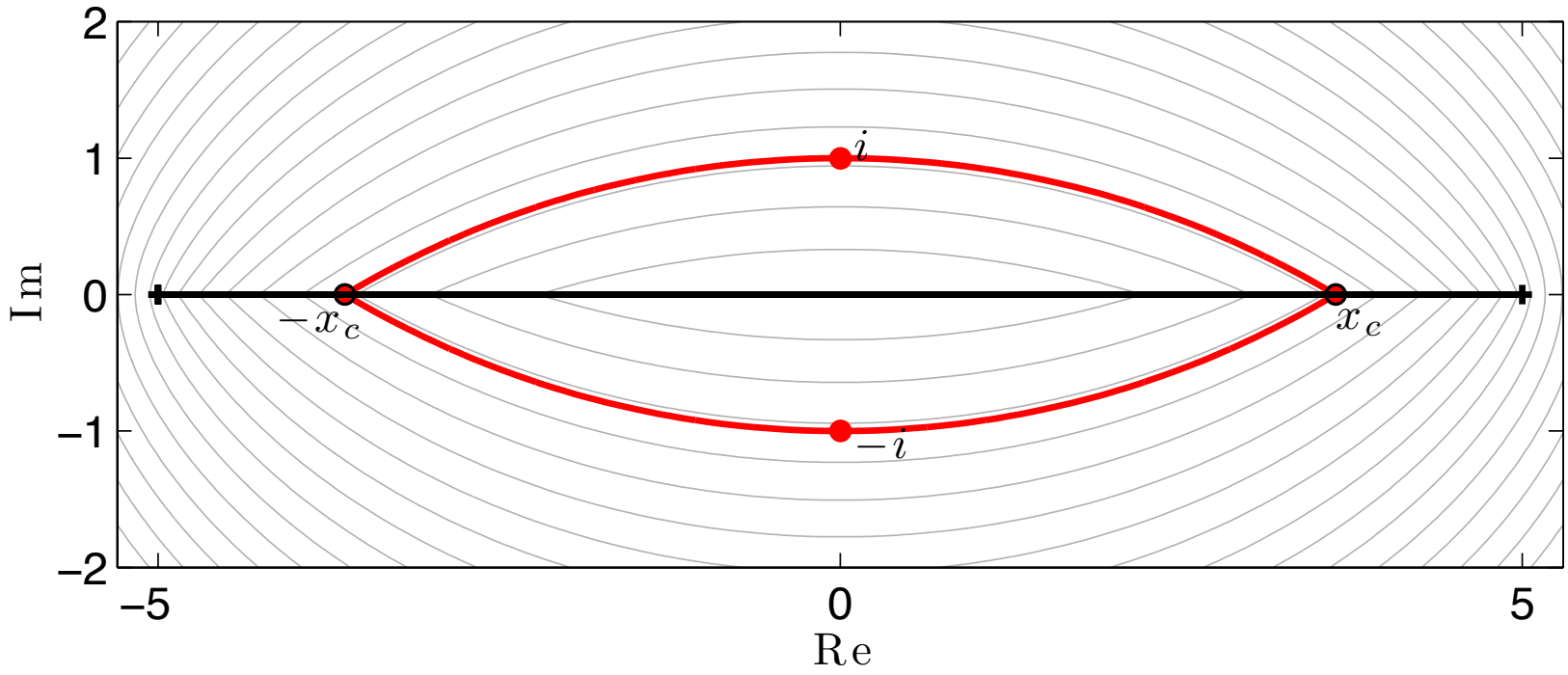


Fazit.

- Durch eine geeignete Knotenwahl (Tschebyscheff-Knoten) lässt sich auch die Runge-Funktion durch Interpolationspolynome beliebig genau annähern.
- Prinzipiell ist eine Approximation durch Interpolationspolynome aber nur dann ratsam, wenn man mit wenigen Knoten (d.h. mit Polynomen niedrigen Grades) ausreichend gute Ergebnisse erzielen kann. Das ist i.A. nur bei extrem glatten Funktionen (wie etwa bei der Exponentialfunktion) gewährleistet. (Die Runge-Funktion ist zwar in ganz $\mathbb{R}$ beliebig oft differenzierbar, besitzt aber Pole in $\pm\sqrt{-1}$. Wie gut eine Funktion durch reelle Interpolationspolynome genähert werden kann, hängt auch von der Lage ihrer komplexen Singularitäten ab!)
- Polynome hohen Grades neigen zu Oszillationen und sind daher zur Approximation oft unbrauchbar.

Für äquidistante Knoten in $[-5,5]$ gilt $\lim_{n\to\infty} |\omega_{n+1}(z)|^{\frac{1}{n+1}} = G(z)$,

$$G(z) = \exp\left\{\frac{1}{10}\operatorname{Re}\left[(z+5)\log(z+5) - (z-5)\log(z-5)\right] - 1\right\}.$$



Höhenlinien von $G(z)$, rot gekennzeichnet ist das Niveau von $G(\pm i)$, welches in $\pm x_c \approx \pm 3.6333843024$ die reelle Achse schneidet.

Satz 6.6 (Runge, 1901)**.** *Besitzt die Funktion f keine Singularität im Gebiet*

$$D_\rho := \{z \in \mathbb{C} : G(z) \leq G(\rho)\}, \qquad \rho > 0,$$

so gilt

$$p_n(x) \to f(x) \quad \textit{für} \quad n \to \infty \textit{ gleichmäßig für } x \in [-\rho, \rho].$$

Eine Anwendung: Numerische Differentiation.

Naheliegende Idee, um die n-te Ableitung einer komplizierten Funktion f anzunähern:

(1) Bestimme ein Interpolationspolynom p vom Grad n für f.
(2) Differenziere p n-mal: $p^{(n)}(x) = n!\, f_{0,1,\ldots,n}$.

Beispiele:

(a) Knoten: x_0 und $x_1 = x_0 + h$, d.h.

$$f'(x_0) \approx p'(x_0) = 1!\, f_{0,1} = \frac{f(x_0 + h) - f(x_0)}{h}.$$

(b) Knoten: $x_0 = x_1 - h$, x_1 und $x_2 = x_1 + h$, d.h.

$$f''(x_1) \approx p''(x_1) = 2!\, f_{0,1,2} = \frac{f(x_1 + h) - 2f(x_1) + f(x_1 - h)}{h^2}.$$

Problematik: Numerische Auslöschung.

Für $f(x) = \sinh(x) = \frac{1}{2}(e^x - e^{-x})$ approximiere

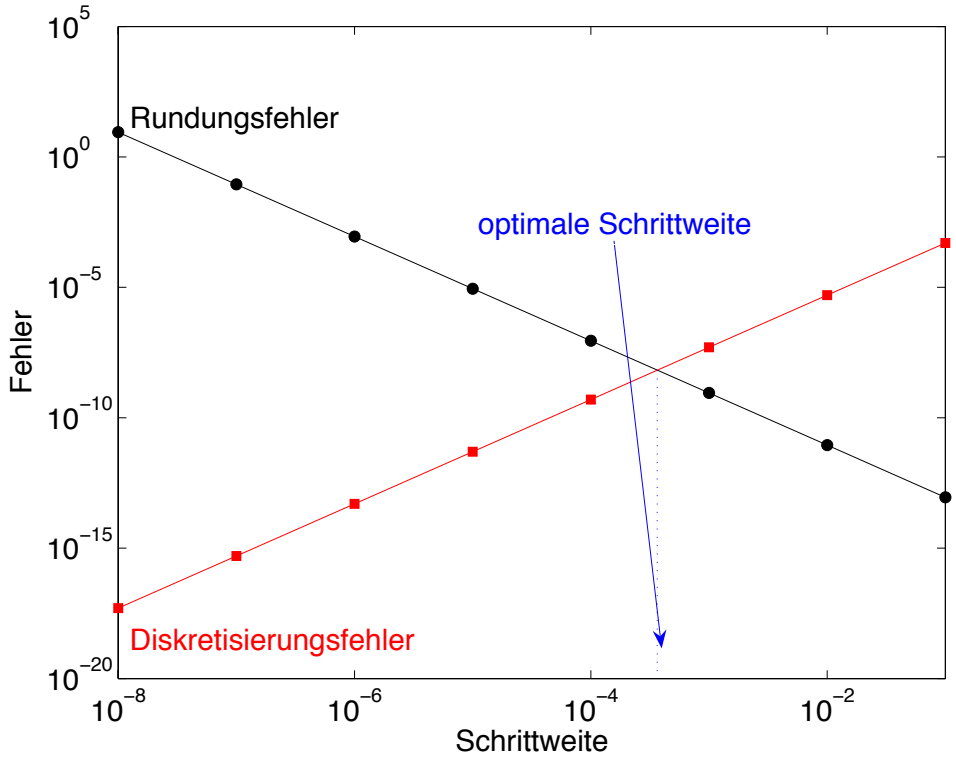
$$0.636653582\ldots = f(0.6) = f''(0.6) \approx \frac{f(0.6-h) - 2f(0.6) + f(0.6+h)}{h^2}$$

für $h = 10^{-e}$, $e = 1, 2, \ldots$, im IEEE-double-Format
(Maschinengenauigkeit: eps $= 2^{-52} \approx 2.2 \cdot 10^{-16}$).

e	$f''(0.6) \approx$	e	$f''(0.6) \approx$
1	0.63718430367986	5	0.63665517302525
2	0.63665888761277	6	0.63682392692499
3	0.63665363525534	7	0.64392935428259
4	0.63665358540632	8	2.22044604925031

$$\text{Diskretisierungsfehler} \quad \sim \quad \frac{1}{12} f^{(4)}(0.6)\, h^2 \quad \approx \quad \frac{1}{20}\, 10^{-2e},$$

$$\text{Rundungsfehler} \quad \approx \quad 4h^{-2}\text{eps} \;=\; 4\ \text{eps}\ 10^{2e}.$$



6.2 Spline-Interpolation

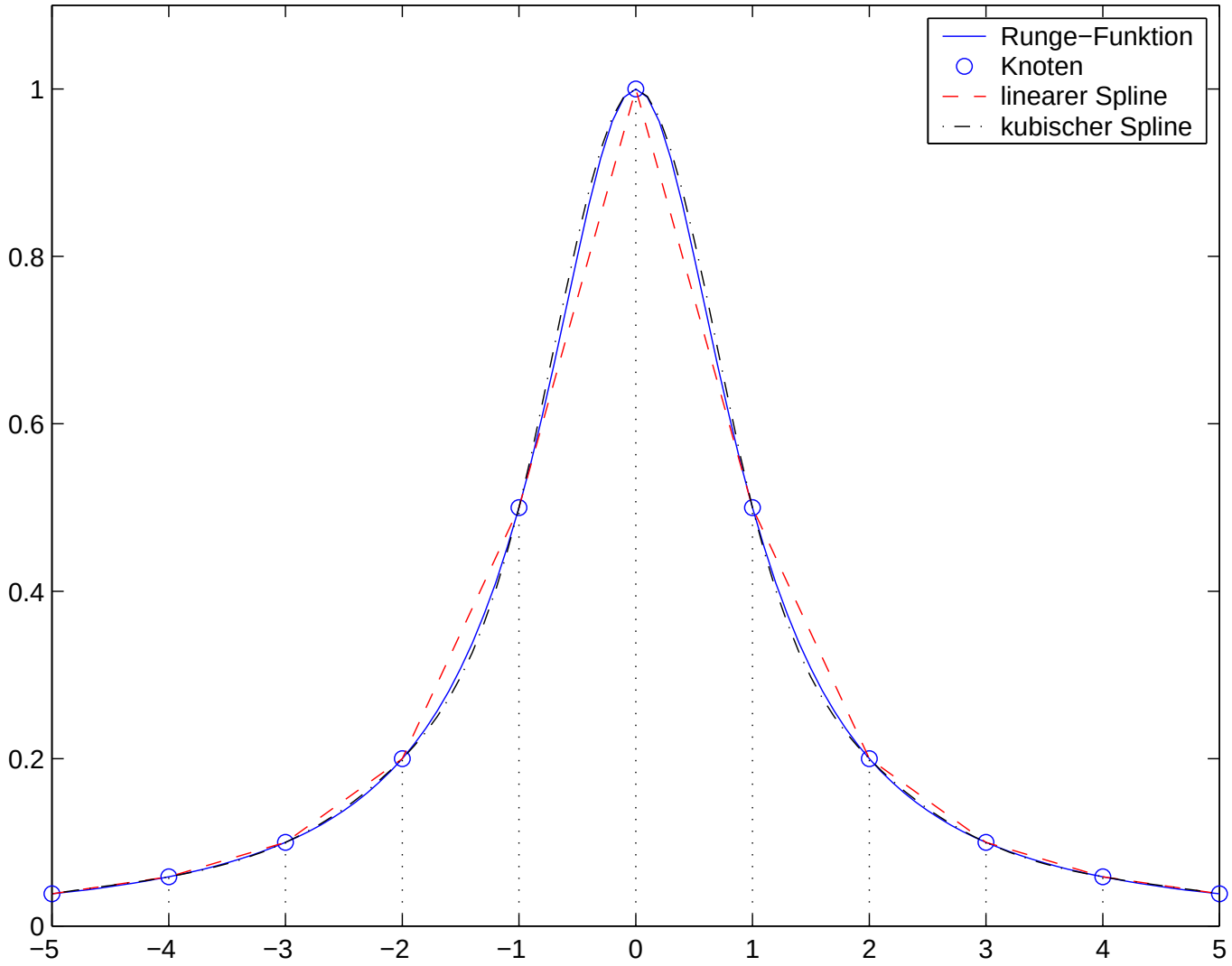
Splines sind „stückweise Polynome“. (Wörtlich: Spezielle biegsame Kurvenlineale, die durch Halterungen gezwungen werden, auf dem Zeichenpapier gegebene Punkte zu verbinden; wurden im Schiffsbau verwendet.)
Idee: Um die Güte der Approximation zu verbesseren, wird hier nicht der Polynomgrad erhöht, sondern die Unterteilung des Intervalls verfeinert.

Seien $n+1$ Knoten in $[a,b]$ gegeben: $a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b$. Mit $\mathscr{T} := [x_0, x_1] \cup [x_1, x_2] \cup \cdots \cup [x_{n-1}, x_n]$ bezeichnen wir die zugehörige Zerlegung des Intervalls $[a,b]$. Ein Spline vom Grad k bez. $\mathscr{T}$ ist eine Funktion $s \in C^{k-1}[a,b]$, die auf jedem Teilintervall von $\mathscr{T}$ mit einem Polynom vom Grad k übereinstimmt:

$$s|_{[x_{i-1},x_i]} \in \mathscr{P}_k \quad \text{für } i = 1, 2, \ldots, n.$$

Satz 6.7. *Die Menge $\mathscr{S}_{\mathscr{T}}^k$ aller Splines vom Grad k bez. $\mathscr{T}$ ist ein $(n+k)$-dimensionaler linearer Raum.*

Im Runge-Beispiel:



6.2.1 Lineare Spline-Interpolation

Einfachster Fall: $k = 1$. Ein Spline s vom Grad 1 (linearer Spline) ist charakterisiert durch die beiden Eigenschaften:
1. Auf jedem Teilintervall $[x_{i-1}, x_i]$ von $\mathscr{T}$ ist s linear:

$$s(x) = \alpha_i + \beta_i x \quad \text{für alle } x \in [x_{i-1}, x_i] \text{ und } i = 1, 2, \ldots, n.$$

2. Auf ganz $[a, b]$ ist s stetig, d.h. für $i = 1, 2, \ldots, n-1$

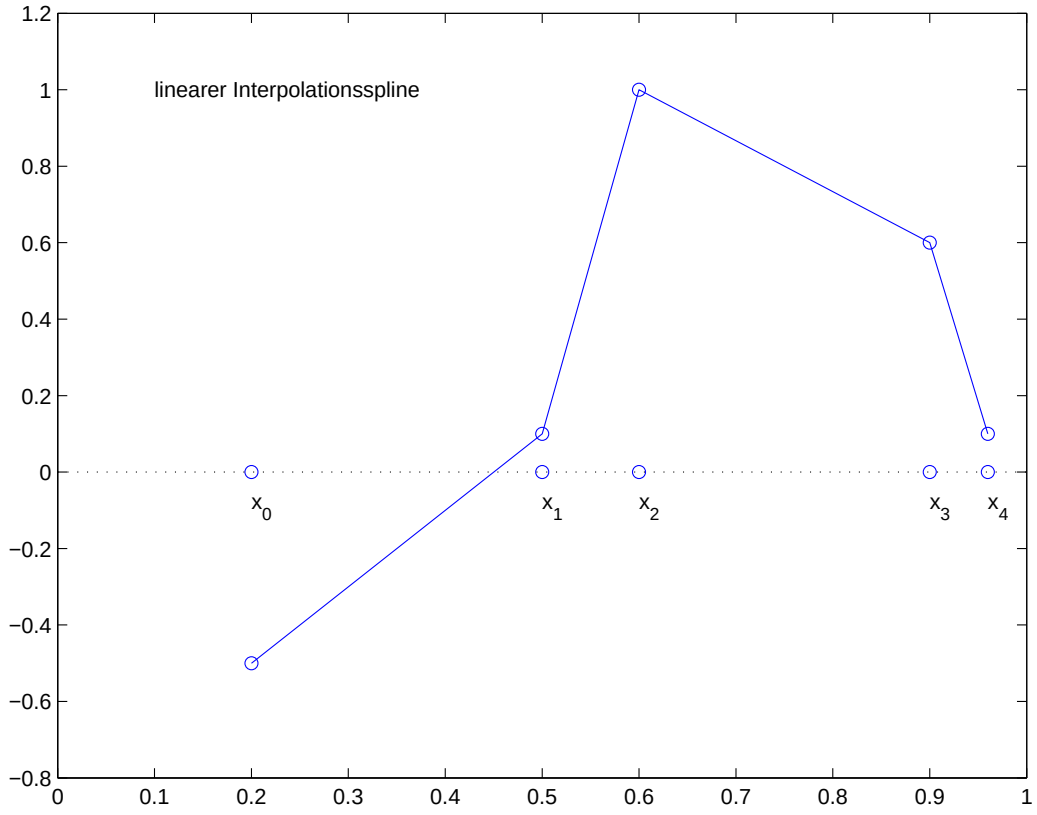
$$\lim_{x \to x_i -} s(x) = \alpha_i + \beta_i x_i = \alpha_{i+1} + \beta_{i+1} x_i = \lim_{x \to x_i +} s(x).$$

Interpolationsaufgabe: Zu vorgebener Zerlegung $\mathscr{T} = [x_0, x_1] \cup [x_1, x_2] \cup \cdots \cup [x_{n-1}, x_n]$ von $[a, b]$ und zu vorgegebenen Werten $f_0, f_1, \ldots, f_n$ bestimme man einen linearen Spline $s \in \mathscr{S}^1_{\mathscr{T}}$ mit

$$s(x_i) = f_i \quad \text{für alle } i = 0, 1, \ldots, n.$$

Offensichtlich: Diese Aufgabe ist eindeutig lösbar:

$$s(x) = f_{i-1} + \frac{f_i - f_{i-1}}{x_i - x_{i-1}} (x - x_{i-1}) \quad \text{für } x \in [x_{i-1}, x_i].$$



Fehler des linearen Interpolationssplines: ($f \in C^2[a,b]$)

Lokal, d.h. für $x \in [x_{i-1}, x_i]$:

$$|f(x) - s(x)| = \frac{1}{2}|f''(\zeta)||(x - x_{i-1})(x - x_i)| \leq \frac{1}{8} M_{2,i}\, h_i^2$$

mit $M_{2,i} = \max_{x_{i-1} \leq \zeta \leq x_i} |f''(\zeta)|$ und $h_i = x_i - x_{i-1}$.

Global, d.h. für $x \in [x_0, x_n]$:

$$|f(x) - s(x)| \leq \frac{1}{8} M_2\, h_{\max}^2$$

mit $M_2 = \max_{1 \leq i \leq n} M_{2,i} = \max_{x_0 \leq \zeta \leq x_n} |f''(\zeta)|$ und $h_{\max} = \max_{1 \leq i \leq n} h_i$.

Adaptive Knotenwahl. Stategie: Fehler etwa gleich auf jedem Teilintervall. D.h.: Wähle h_i invers proportional zu $\sqrt{M_{2,i}}$ (viele Knoten dort, wo die Krümmung von f groß ist).

Zur Implementierung.
Gegeben: $x_0, x_1, \ldots, x_n$ und $f_0, f_1, \ldots, f_n$.

Gesucht: Wert $s(x)$ des linearen Interpolationssplines an der Stelle x.

Bestimme $g_{i-1} = (f_i - f_{i-1})/(x_i - x_{i-1})$ für $i = 1, 2, \ldots, n$.

Falls $x \in [x_{i-1}, x_i]$, dann $s(x) = f_{i-1} + g_{i-1}\,(x - x_{i-1})$.

Problem: Gegeben x, in welchem Teilintervall $[x_{i-1}, x_i]$ liegt x?

Einfach, falls $h_i = h$ (äquidistante Knoten):

$$i = \left\lceil \frac{x - x_0}{h} \right\rceil := \min \left\{ k \in \mathbb{N} \,:\, k \geq \frac{x - x_0}{h} \right\}.$$

Schwieriger bei beliebigen Knoten:
Binäres Suchen ergibt Komplexität von $\approx \log_2 n$.

6.2.2 Kubische Spline-Interpolation

Gesucht ist ein interpolierender kubischer Spline $s \in \mathscr{S}_{\mathscr{T}}^3$.
Charakteristische Eigenschaften:

(1) Auf jedem Teilintervall $[x_{i-1}, x_i]$ von $\mathscr{T}$ ist s kubisch:

$$s(x) = p_i(x) = \alpha_i + \beta_i(x - x_{i-1}) + \gamma_i(x - x_{i-1})^2 + \delta_i(x - x_{i-1})^3.$$

(2) Auf ganz $[a, b]$ ist s zweimal stetig differenzierbar, d.h.:

$$p_i(x_i) = p_{i+1}(x_i), \quad p_i'(x_i) = p_{i+1}'(x_i), \quad p_i''(x_i) = p_{i+1}''(x_i)$$

für $i = 1, 2, \ldots, n-1$.

(3) Interpolationsbedingungen:

$$s(x_i) = f_i, \qquad i = 0, 1, \ldots, n.$$

Fazit: $3(n-1) + (n+1) = 4n - 2$ Bedingungen, aber $4n$ Freiheitsgrade.

Drei Möglichkeiten für die erforderlichen zwei Zusatzbedingungen.

Natürlicher Spline:

$$s''(x_0) = s''(x_n) = 0 \tag{N}$$

Hermitescher oder **vollständiger Spline** [CHARLES HERMITE (1822–1901)]:

$$s'(x_0) = f'_0 \quad \text{und} \quad s'(x_n) = f'_n \quad \text{mit} \quad f'_0, f'_n \in \mathbb{R}. \tag{H}$$

Periodischer Spline: Falls $s(x_0) = s(x_n)$,

$$s'(x_0) = s'(x_n) \quad \text{und} \quad s''(x_0) = s''(x_n). \tag{P}$$

Berechnung des kubischen Interpolationssplines.

Auf jedem Teilintervall $[x_{i-1}, x_i]$ hat der kubische Spline die Form

$$s(x) = p_i(x) = \alpha_i + \beta_i(x - x_{i-1}) + \gamma_i(x - x_{i-1})^2 + \delta_i(x - x_{i-1})^3.$$

Die Koeffizienten lassen sich durch die Momente $\mu_i := s''(x_i)$ und die Funktionswerte f_i $(i = 0, 1, \ldots, n)$ darstellen:

$$\alpha_i = f_{i-1}, \qquad \beta_i = \frac{f_i - f_{i-1}}{h_i} - \frac{h_i}{6}(\mu_i + 2\mu_{i-1}),$$

$$\gamma_i = \frac{1}{2}\mu_{i-1}, \qquad \delta_i = \frac{\mu_i - \mu_{i-1}}{6h_i},$$

wobei $h_i := x_i - x_{i-1}$.

M. a. W.: Ein kubischer Spline ist durch die Funktionswerte f_i und die Momente μ_i $(i = 0, 1, \ldots, n)$ eindeutig bestimmt.

Die $(n+1)$ Momente μ_i erfüllen die $(n-1)$ linearen Gleichungen

$$\frac{h_i}{6}\mu_{i-1} + \frac{h_i + h_{i+1}}{3}\mu_i + \frac{h_{i+1}}{6}\mu_{i+1} = \frac{f_{i+1} - f_i}{h_{i+1}} - \frac{f_i - f_{i-1}}{h_i}$$

$(i = 1, 2, \ldots, n-1)$ und zwei Zusatzgleichungen:

(N)
$$\mu_0 = 0,$$
$$\mu_n = 0,$$

(H)
$$\frac{h_1}{3}\mu_0 + \frac{h_1}{6}\mu_1 = \frac{f_1 - f_0}{h_1} - f_0',$$
$$\frac{h_n}{6}\mu_{n-1} + \frac{h_n}{3}\mu_n = f_n' - \frac{f_n - f_{n-1}}{h_n},$$

(P)
$$\mu_0 = \mu_n,$$
$$\frac{h_1}{6}\mu_1 + \frac{h_1}{6}\mu_{n-1} + \frac{h_1 + h_n}{3}\mu_n = \frac{f_1 - f_n}{h_1} - \frac{f_n - f_{n-1}}{h_n}.$$

Im Folgenden werden nur vollständige kubische Splines (Bedingung (H)) betrachtet, analoge Aussagen gelten unter den Bedingungen (N) bzw. (P).

Die Momente des vollständigen kubischen Splines erfüllen das LGS (s.o.)

$$\begin{bmatrix} \frac{h_1}{3} & \frac{h_1}{6} & & & \\ \frac{h_1}{6} & \frac{h_1+h_2}{3} & \frac{h_2}{6} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{h_{n-1}}{6} & \frac{h_{n-1}+h_n}{3} & \frac{h_n}{6} \\ & & & \frac{h_n}{6} & \frac{h_n}{3} \end{bmatrix} \begin{bmatrix} \mu_0 \\ \mu_1 \\ \vdots \\ \mu_{n-1} \\ \mu_n \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix} \tag{6.4}$$

mit $d_0 = \dfrac{f_1 - f_0}{h_1} - f_0', \qquad d_j = \dfrac{f_{j+1} - f_j}{h_{j+1}} - \dfrac{f_j - f_{j-1}}{h_j} \quad (1 \le j \le n),$

und $d_n = f_n' - \dfrac{f_n - f_{n-1}}{h_n}.$

Satz 6.8. *Für jede Wahl der Knoten* $a = x_0 < x_1 < \cdots < x_n = b$ *ist das Gleichungssystem* (6.4) *eindeutig lösbar. D.h.: Zu jeder Knotenwahl gibt es genau einen vollständigen kubischen Interpolationsspline für* f*.*

Satz 6.9 (**Fehler bei kubischer Spline-Interpolation).** *Ist* $f \in C^4[a,b]$ *und* $s \in \mathscr{S}_{\mathscr{T}}^3$ *der vollständige kubische Interpolationsspline für* f*, dann gelten*

$$\begin{aligned}
\max_{x\in[a,b]} |f(x) - s(x)| &\leq \frac{5}{384} M_4 \, h_{\max}^4, \\
\max_{x\in[a,b]} |f'(x) - s'(x)| &\leq \frac{1}{24} M_4 \, h_{\max}^3, \\
\max_{x\in[a,b]} |f''(x) - s''(x)| &\leq \frac{3}{8} M_4 \, h_{\max}^2
\end{aligned}$$

mit $M_4 := \max\limits_{a\leq x\leq b} |f^{(4)}(x)|$ *und* $h_{\max} := \max\limits_{1\leq i\leq n} h_i = \max\limits_{1\leq i\leq n} (x_i - x_{i-1})$.

Wir definieren allgemein

$$\mathscr{H}^k = \mathscr{H}^k(a,b) := \big\{ f : [a,b] \to \mathbb{R} : f, f', \ldots, f^{(k-1)} \text{ absolutstetig }, \\ f^{(k)} \text{ ex. f.ü } , f^{(k)} \in L^2(a,b) \big\}$$

und setzen für $f \in \mathscr{H}^2$,

$$|f|_2 := \left(\int_a^b |f''(x)|^2 \, dx \right)^{1/2}.$$

Lemma 6.10. *Für $f \in \mathscr{H}^2$ und $s \in \mathscr{S}^3_{\mathscr{T}}$ gilt*

$$|f-s|_2^2 = |f|_2^2 - |s|_2^2 \\ - 2 \left\{ [f'(x) - s'(x)] s''(x) \big|_a^b - \sum_{i=1}^{n} [f(x) - s(x)] s'''(x) \big|_{x_{i-1}+}^{x_i-} \right\}.$$

Satz 6.11 (Minimierungseigenschaft kubischer Splines)**.** *Ist $f \in \mathscr{H}^2$ und $s \in \mathscr{S}^3_{\mathscr{T}}$ ein zugehöriger kubischer Interpolationsspline, der eine der drei Zusatzbedingungungen (N), (H) oder (P) erfüllt, dann folgt*

$$|s|_2^2 \leq |f|_2^2 \left(= \int_a^b f''(x)^2 dx \right).$$

Interpretation von Satz 6.11. Unter allen Funktionen $f \in \mathscr{H}^2$ mit

$$f(x_i) = f_i, \qquad i = 0, 1, \ldots, n,$$

minimiert der interpolierende kubische Spline mit einer der Zusatzbedingungen (H), (N) oder (P) näherungsweise die Biegeenergie

$$E_B(f) := \int_a^b \frac{f''(x)^2}{[1 + f'(x)^2]^{3/2}} dx \approx \int_a^b f''(x)^2 \, dx.$$

6.3 Bestapproximation in Innenprodukträumen

Sei $\mathcal{V}$ ein Vektorraum über $\mathbb{R}$ oder $\mathbb{C}$ mit Innenprodukt $(\cdot,\cdot)$. Dann wird durch $\|\boldsymbol{v}\| := (\boldsymbol{v},\boldsymbol{v})^{1/2}$ ($\boldsymbol{v} \in \mathcal{V}$) eine Norm auf $\mathcal{V}$ definiert. Ist $\mathcal{V}$ bez. dieser Norm vollständig, so heisst $(\mathcal{V},(\cdot,\cdot))$ ein Hilbert-Raum.

Beispiele:

1.) $\mathbb{R}^n$ ($\mathbb{C}^n$) mit Innenprodukt $(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{y}^\top \boldsymbol{x}$ ($(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{y}^H \boldsymbol{x}$) ist ein Hilbert-Raum. (Die vom Innenprodukt induzierte Norm ist die Euklid-Norm.)

2.) $\ell^2 := \left\{\boldsymbol{x} = \{x_j\}_{j\in\mathbb{N}} \subset \mathbb{C} \,:\, \sum_{j=0}^{\infty} |x_i|^2 < \infty\right\}$ mit dem Innenprodukt $(\boldsymbol{x},\boldsymbol{y}) = \sum_{j=1}^{\infty} x_j \overline{y}_j$ ist ein Hilbert-Raum.

3.) $\mathbb{C}^\infty := \left\{\boldsymbol{x} = (x_j)_{j\in\mathbb{N}} \in \ell^2 \,:\, x_j = 0 \text{ bis auf endlich viele } j\right\}$ mit dem Innenprodukt $(\boldsymbol{x},\boldsymbol{y}) = \sum_{j=1}^{\infty} x_j \overline{y}_j$ ist kein Hilbert-Raum.

4.) $\mathbb{C}^{n\times n}$ mit dem Innenprodukt $(A,B) = \operatorname{tr}(B^H A)$ ist ein Hilbert-Raum. (Die vom Innenprodukt induzierte Norm ist die Frobenius-Norm.)

5.) $L^2(a,b) = \{f : [a,b] \to \mathbb{C} \,:\, \int_a^b |f(x)|^2 dx < \infty\}$ mit dem Innenprodukt $(f,g) = \int_a^b f(x)\overline{g(x)}dx$ ist ein Hilbert-Raum.

Approximationsaufgabe: Sei $\mathscr{U}$ ein endlich-dimensionaler Teilraum des Innenproduktraums $\mathscr{V}$ und $\boldsymbol{v} \in \mathscr{V}$. Bestimme $\boldsymbol{u}^* = \boldsymbol{u}^*(\boldsymbol{v}) \in \mathscr{U}$ mit

$$\|\boldsymbol{u}^* - \boldsymbol{v}\| < \|\boldsymbol{u} - \boldsymbol{v}\| \qquad \text{für alle } \boldsymbol{u} \in \mathscr{U},\ \boldsymbol{u} \neq \boldsymbol{u}^*.$$

$\boldsymbol{u}^*$ heißt die Bestapproximation an $\boldsymbol{v}$ aus $\mathscr{U}$.

Erinnerung. Sei $\mathscr{U}$ ein endlich-dimensionaler Teilraum des Innenproduktraums $\mathscr{V}$. Dann ist die Orthogonalprojektion auf $\mathscr{U}$ $P : \mathscr{V} \to \mathscr{U}$ definiert durch

$$P\boldsymbol{v} = \begin{cases} \boldsymbol{v} & \boldsymbol{v} \in \mathscr{U}, \\ \boldsymbol{0} & \boldsymbol{v} \in \mathscr{U}^{\perp}. \end{cases}$$

Ist $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n\}$ eine Orthonormalbasis von $\mathscr{U}$, so gilt

$$P\boldsymbol{v} = (\boldsymbol{v}, \boldsymbol{u}_1)\boldsymbol{u}_1 + (\boldsymbol{v}, \boldsymbol{u}_2)\boldsymbol{u}_2 + \cdots + (\boldsymbol{v}, \boldsymbol{u}_n)\boldsymbol{u}_n \qquad \text{für alle } \boldsymbol{v} \in \mathscr{V}.$$

Satz 6.12. *Sei $\mathscr{U}$ ein endlich-dimensionaler Teilraum des Innenproduktraums $\mathscr{V}$, P die Orthogonalprojektion auf $\mathscr{U}$ und $\boldsymbol{v} \in \mathscr{V}$.*
Dann ist die Bestapproximation $\boldsymbol{u}^*$ aus $\mathscr{U}$ an $\boldsymbol{v}$ gegeben durch $\boldsymbol{u}^* = P\boldsymbol{v}$.
Die Bestapproximation ist eindeutig bestimmt und charakterisiert durch

$$\boldsymbol{u}^* - \boldsymbol{v} \perp \mathscr{U}\,.$$

Ist $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n\}$ eine Orthonormalbasis von $\mathscr{U}$, so gelten

$$\boldsymbol{u}^* = \sum_{j=1}^{n} (\boldsymbol{v}, \boldsymbol{u}_j)\boldsymbol{u}_j \quad \text{und} \quad \|\boldsymbol{u}^*\| = \left(\sum_{j=1}^{n} |(\boldsymbol{v}, \boldsymbol{u}_j)|^2 \right)^{1/2} \leq \|\boldsymbol{v}\|$$

sowie

$$\|\boldsymbol{u}^* - \boldsymbol{v}\|^2 = \|\boldsymbol{v}\|^2 - \|\boldsymbol{u}^*\|^2.$$

Beispiel. Die Bestapproximation an $A \in \mathbb{R}^{n\times n}$ aus dem Unterraum der symmetrischen Matrizen (bez. der Frobenius-Norm) ist

$$A_S := \tfrac{1}{2}(A + A^\top) \quad \text{(der symmetrische Anteil von } A\text{)}.$$

Beispiel. Der Raum $\mathscr{T}_n$ der trigonometrischen Polynome vom Grad n definiert durch

$$\mathscr{T}_n := \operatorname{span}\{e^{ikt} : k = 0, \pm 1, \ldots, \pm n\} \subset L^2(0, 2\pi), \quad \text{(Bezeichnung: } i^2 = -1\text{)}$$

besitzt die Dimension $2n + 1$. Die Funktionen $\{\frac{1}{\sqrt{2\pi}} e^{ikt}\}_{k=-n}^{n}$ bilden eine ON-Basis von $\mathscr{T}_n$. Die Bestapproximation an $f \in L^2(0, 2\pi)$ aus $\mathscr{T}_n$ ist also

$$u_n^*(t) = \sum_{k=-n}^{n} a_k e^{ikt} \quad \text{mit} \quad a_k = \frac{1}{2\pi} \int_0^{2\pi} f(t) e^{-ikt}\, dt.$$

Bemerkung. Im Fall von $a_k = \overline{a}_{-k}$, $k = 0, 1, \ldots, n$, (z.B. wenn f reellwertig ist) folgt mit $\alpha_0 = 2a_0,\ \alpha_k = 2\,\mathrm{Re}(a_k),\ \beta_k = -2\,\mathrm{Im}(a_k)$ $(k = 1, 2, \ldots, n)$.

$$u_n^*(t) = \frac{\alpha_0}{2} + \sum_{k=1}^{n} \big[\alpha_k \cos(kt) + \beta_k \sin(kt)\big].$$

Dies folgt aus

$$\begin{aligned} u_n^*(t) &= \sum_{k=-n}^{n} a_k e^{ikt} = a_0 + \sum_{k=1}^{n} a_k e^{ikt} + \sum_{k=1}^{n} a_{-k} e^{-ikt} \\ &= a_0 + \sum_{k=1}^{n} \big(a_k[\cos(kt) + i\sin(kt)] + \overline{a_k}[\cos(kt) - i\sin(kt)]\big) \\ &= \underbrace{a_0}_{=:\frac{\alpha_0}{2}} + \sum_{k=1}^{n} \big[\underbrace{2\,\mathrm{Re}(a_k)}_{=:\alpha_k} \cos(kt) \underbrace{-2\,\mathrm{Im}(a_k)}_{=:\beta_k} \sin(kt)\big]. \end{aligned}$$

6.4 Trigonometrische Interpolation

Seien

$$f_0, f_1, \ldots, f_{m-1} \in \mathbb{R} \quad \text{und} \quad x_j := 2\pi j/m, \quad j = 0, 1, \ldots, m-1,$$

d.h. $x_0 < x_1 < \cdots < x_{m-1}$ sind äquidistante Knoten aus $[0, 2\pi)$.

Gesucht ist ein reelles trigonometrisches Polynom vom Grad n,

$$t_n(x) = \frac{\alpha_0}{2} + \sum_{k=1}^{n} \big[\alpha_k \cos(kx) + \beta_k \sin(kx)\big],$$

das die m Interpolationsbedingungen

$$t_n(x_j) = f_j \qquad (j = 0, 1, \ldots, m-1) \tag{6.5}$$

erfüllt. Hierbei ist

$$n = \begin{cases} \frac{m}{2} & \text{falls } m \text{ gerade,} \\ \frac{m-1}{2} & \text{falls } m \text{ ungerade.} \end{cases}$$

Transformation auf den (komplexen) Einheitskreis:

$$\phi : [0, 2\pi) \longrightarrow \mathbb{T} := \{z \in \mathbb{C} : |z| = 1\}, \quad x \mapsto z = e^{ix} = \cos x + i \sin x.$$

Die Knoten x_j gehen über in die m-ten Einheitswurzeln:

$$\phi(x_j) = e^{2\pi ij/m} = [e^{2\pi i/m}]^j = \omega_m^j, \qquad j = 0, 1, \ldots, m-1,$$

mit $\omega_m := e^{2\pi i/m} = \cos \frac{2\pi}{m} + i \sin \frac{2\pi}{m}$.

Setzt man $\beta_0 = 0$ und für $k = 0, 1, \ldots, n$

$$\mathbb{C}^2 \ni \begin{bmatrix} a_k \\ a_{-k} \end{bmatrix} := \begin{bmatrix} \frac{1}{2}(\alpha_k - i\beta_k) \\ \frac{1}{2}(\alpha_k + i\beta_k) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & -i \\ 1 & i \end{bmatrix} \begin{bmatrix} \alpha_k \\ \beta_k \end{bmatrix}, \text{ d.h.}$$

$$\mathbb{R}^2 \ni \begin{bmatrix} \alpha_k \\ \beta_k \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix} \begin{bmatrix} a_k \\ a_{-k} \end{bmatrix} = \begin{bmatrix} a_k + a_{-k} \\ i(a_k - a_{-k}) \end{bmatrix} = \begin{bmatrix} 2\ \mathrm{Re}\, a_k \\ -2\,\mathrm{Im}\, a_k \end{bmatrix},$$

so folgt

$$t_n(x) = \sum_{k=-n}^{n} a_k e^{ikx} = \sum_{k=-n}^{n} a_k z^k = z^{-n} \sum_{k=-n}^{n} a_k z^{k+n} = z^{-n} p_{2n}(z)$$

mit $p_{2n}(z) = \sum_{k=-n}^{n} a_k z^{k+n} = \sum_{j=0}^{2n} a_{j-n} z^j \in \mathscr{P}_{2n}$.

Wegen

$$p_{2n}(\omega_m^j) = \omega_m^{jn}\ t_n(x_j)$$

ist die trigonometrische Interpolationsaufgabe hiermit zurückgeführt auf eine (gewohnliche) Interpolationsaufgabe für (algebraische) Polynome.

Satz 6.13. *Zu beliebig vorgegebenen paarweise verschiedenen Knoten* $x_0, x_1, \ldots, x_{2n} \in [0, 2\pi)$ *und zu beliebigen Funktionswerten* $f_0, f_1, \ldots, f_{2n} \in \mathbb{R}$ *gibt es genau ein reelles trigonometrisches Polynom* $t_n \in \mathscr{T}_n$ *mit* $t_n(x_j) = f_j \quad (j = 0, 1, \ldots, 2n)$.

Lemma 6.14. *Für die* m*-ten Einheitswurzeln* ω_m^k *(*$k \in \mathbb{Z}$, $m \in \mathbb{N}$*) gelten:*

a) $$[\omega_m^k]^j = \omega_m^{kj} = [\omega_m^j]^k \quad (j \in \mathbb{Z}),$$

b) $$\omega_{m\ell}^{k\ell} = \omega_m^k \quad (\ell \in \mathbb{Z},\, \ell \neq 0),$$

c) $$\overline{\omega_m^k} = \omega_m^{-k},$$

d) $$\sum_{j=0}^{m-1} \omega_m^{kj} = \begin{cases} m, & \textit{falls} \quad k = 0 \pmod m, \\ 0, & \textit{falls} \quad k \neq 0 \pmod m. \end{cases}$$

Satz 6.15. *Das komplexe (algebraische) Interpolationspolynom*

$$p_{m-1}(z) = \sum_{k=0}^{m-1} c_k z^k \in \mathscr{P}_{m-1}$$

mit $p_{m-1}(\omega_m^j) = f_j \in \mathbb{C} \quad (j = 0, 1, \ldots, m-1)$ *besitzt die Koeffizienten*

$$c_k = \frac{1}{m} \sum_{j=0}^{m-1} f_j \omega_m^{-kj}, \qquad k = 0, 1, \ldots, m-1. \tag{6.6}$$

In Matrix-Vektor-Schreibweise

$$\begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{m-1} \end{bmatrix} = \frac{1}{m} F_m \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_{m-1} \end{bmatrix}$$

mit der Fourier-Matrix

$$F_m := \left[\omega_m^{-kj}\right]_{0\le k,j\le m-1} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega_m^{-1} & \cdots & \omega_m^{-m+1} \\ \vdots & \vdots & & \vdots \\ 1 & \omega_m^{-m+1} & \cdots & \omega_m^{-(m-1)^2} \end{bmatrix}.$$

Bemerkung. Mit den Bezeichnungen aus Satz 6.15 minimiert das „abgeschnittene“ Interpolationspolynom

$$p_{m,d}(z) := c_0 + c_1 z + \cdots + c_d z^d, \qquad 0 \le d \le m-1,$$

unter allen Polynomen $q \in \mathscr{P}_d$ die Fehlerquadratsumme zur Interpolationsvorschrift:

$$\sum_{j=0}^{m-1} |f_j - p_{m,d}(\omega_m^j)|^2 < \sum_{j=0}^{m-1} |f_j - q(\omega_m^j)|^2 \quad \text{für alle } q \in \mathscr{P}_d,\ q \ne p_{m,d}.$$

Satz 6.16. *Für $m = 2n$ oder $m = 2n + 1$ gibt es zu beliebigen $f_0, f_1, \ldots, f_{m-1} \in \mathbb{R}$ ein reelles trigonometrisches Interpolationspolynom*

$$t_n(x) = \frac{\alpha_0}{2} + \sum_{k=1}^{n} \big[\alpha_k \cos(kx) + \beta_k \sin(kx)\big] \in \mathscr{T}_n$$

vom Grad n, das die m Bedingungen

$$t_n(2\pi j/m) = f_j \qquad (j = 0, 1, \ldots, m-1)$$

erfüllt. Seine Koeffizienten sind gegeben durch

$$\alpha_k = \frac{2}{m} \sum_{j=0}^{m-1} f_j \cos \frac{2\pi jk}{m} \quad \textit{bzw.} \quad \beta_k = \frac{2}{m} \sum_{j=0}^{m-1} f_j \sin \frac{2\pi jk}{m}, \quad (k = 0, 1, \ldots, n).$$

Im Fall $m = 2n$ muss $\beta_n = 0$ gesetzt und α_n halbiert werden.

6.5 Schnelle Fourier-Transformation (FFT)

Seien $\{\omega_m^j\}_{j=0}^{m-1}$ die m-ten Einheitswurzeln, ($\omega_m := e^{2\pi i/m}$).

Wir unterscheiden zwei grundlegende Aufgabenstellungen:

Diskrete Fourier-Analyse: Bestimme zu vorgegebenen Funktionswerten $f_0, \ldots, f_{m-1} \in \mathbb{C}$ die Koeffizienten $c_0, \ldots, c_{m-1}$ des Interpolationspolynoms

$$p(z) = \sum_{j=0}^{m-1} c_j z^j \quad \text{mit} \quad p(\omega_m^j) = f_j \qquad (j = 0, \ldots, m-1).$$

Wir wissen: Mit der Fourier-Matrix $F_m := [\omega_m^{-kj}]_{0 \le k,j \le m-1} \in \mathbb{C}^{m \times m}$ gilt

$$\begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{m-1} \end{bmatrix} = \frac{1}{m} F_m \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_{m-1} \end{bmatrix}.$$

Diskrete Fourier-Synthese (inverse Aufgabe): Bestimme zu vorgebenen Koeffizienten $c_0, \ldots, c_{m-1} \in \mathbb{C}$ die Funktionswerte $f_0, \ldots, f_{m-1}$ des Polynoms $p(z) = \sum_{j=0}^{m-1} c_j z^j$ an den m-ten Einheitswurzeln $\omega_m^0, \ldots \omega_m^{m-1}$.
Offensichtlich:

$$\begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_{m-1} \end{bmatrix} = W_m \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{m-1} \end{bmatrix}$$

mit der Matrix $W_m := [\omega_m^{kj}]_{0 \le k,j \le m-1} = F_m^H (= \overline{F_m}) \in \mathbb{C}^{m \times m}$.

Lemma 6.17. *Für die Fourier-Matrix* $F_m = [\omega_m^{-kj}]_{0\le k,j\le m-1} \in \mathbb{C}^{m\times m}$ *gelten:*

(a) $F_m^\top = F_m$ *(aber* $F_m^H \neq F_m$ *für* $m > 2$*!),*
(b) $F_m^H F_m = mI_m$*, d.h. die Spalten von* F_m *sind orthogonal und besitzen alle die Euklid-Norm* $\sqrt{m}$*.*
(c) $F_m^{-1} = \frac{1}{m}F_m^H = \frac{1}{m}\overline{F_m}$*.*

Diskrete Fourier-Transformationen (d.h. diskrete Fourier-Analysen und Synthesen) müssen in der Praxis oft berechnet werden (Signalverarbeitung, Lösung der Poisson-Gleichung etc.).

Die „naive“ Berechnung einer Fourier-Transformation (Matrix-Vektor Produkt mit F_m/m bzw. W_m) erfordert offenbar $O(m^2)$ komplexe Multiplikationen. Bei Anwendung der schnellen Fourier-Transformation (FFT) reduziert sich dieser Aufwand auf $O(m \log m)$[a].

[a] James William Cooley(∗1926) and John Wilder Tukey (1915–2000): An algorithm for the machine calculation of complex Fourier series, *Math. Comp.* **19**, 297–301 (1965).

Diese Verbesserung kann nicht überbewertet werden:

*„It [the FFT] has changed the face of science and engineering so much that it is not an exaggeration to say that **life as we know it would be very different without the FFT**.“*

[Charles Van Loan, Computational Frameworks for the Fast Fourier Transform, SIAM, Philadelphia 1992, p. ix]

Wir setzen (aus schreibtechnischen Gründen) im Folgenden

$$\zeta_m := \overline{\omega_m} = e^{-2\pi i/m} = \cos\left(\frac{2\pi}{m}\right) - i\sin\left(\frac{2\pi}{m}\right),$$

so dass $F_m = [\zeta_m^{kj}]_{0\leq k,j\leq m-1}$. Außerdem sei m gerade.

Die Idee der FFT (für $m = 8$): Mit $\zeta := \zeta_8$ ist

$$F_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \zeta & \zeta^2 & \zeta^3 & \zeta^4 & \zeta^5 & \zeta^6 & \zeta^7 \\ 1 & \zeta^2 & \zeta^4 & \zeta^6 & \zeta^8 & \zeta^{10} & \zeta^{12} & \zeta^{14} \\ 1 & \zeta^3 & \zeta^6 & \zeta^9 & \zeta^{12} & \zeta^{15} & \zeta^{18} & \zeta^{21} \\ 1 & \zeta^4 & \zeta^8 & \zeta^{12} & \zeta^{16} & \zeta^{20} & \zeta^{24} & \zeta^{28} \\ 1 & \zeta^5 & \zeta^{10} & \zeta^{15} & \zeta^{20} & \zeta^{25} & \zeta^{30} & \zeta^{35} \\ 1 & \zeta^6 & \zeta^{12} & \zeta^{18} & \zeta^{24} & \zeta^{30} & \zeta^{36} & \zeta^{42} \\ 1 & \zeta^7 & \zeta^{14} & \zeta^{21} & \zeta^{28} & \zeta^{35} & \zeta^{42} & \zeta^{49} \end{bmatrix}.$$

Wegen $\zeta^8 = 1$, d.h. $\zeta^j = \zeta^k$, wenn $j - k$ (ohne Rest) durch 8 teilbar ist, folgt

$$F_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \zeta & \zeta^2 & \zeta^3 & \zeta^4 & \zeta^5 & \zeta^6 & \zeta^7 \\ 1 & \zeta^2 & \zeta^4 & \zeta^6 & 1 & \zeta^2 & \zeta^4 & \zeta^6 \\ 1 & \zeta^3 & \zeta^6 & \zeta & \zeta^4 & \zeta^7 & \zeta^2 & \zeta^5 \\ 1 & \zeta^4 & 1 & \zeta^4 & 1 & \zeta^4 & 1 & \zeta^4 \\ 1 & \zeta^5 & \zeta^2 & \zeta^7 & \zeta^4 & \zeta & \zeta^6 & \zeta^3 \\ 1 & \zeta^6 & \zeta^4 & \zeta^2 & 1 & \zeta^6 & \zeta^4 & \zeta^2 \\ 1 & \zeta^7 & \zeta^6 & \zeta^5 & \zeta^4 & \zeta^3 & \zeta^2 & \zeta^1 \end{bmatrix}.$$

Jetzt nummerieren wir die Zeilen von F_8 um: zuerst werden die mit geradem (0,2,4,6), danach die mit ungeradem Index (1,3,5,7) gezählt. Die zugehörige Pemutationsmatrix wird mit P bezeichnet.

$$PF_8 = \left[\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \zeta^2 & \zeta^4 & \zeta^6 & 1 & \zeta^2 & \zeta^4 & \zeta^6 \\ 1 & \zeta^4 & 1 & \zeta^4 & 1 & \zeta^4 & 1 & \zeta^4 \\ 1 & \zeta^6 & \zeta^4 & \zeta^2 & 1 & \zeta^6 & \zeta^4 & \zeta^2 \\ \hline 1 & \zeta & \zeta^2 & \zeta^3 & \zeta^4 & \zeta^5 & \zeta^6 & \zeta^7 \\ 1 & \zeta^3 & \zeta^6 & \zeta & \zeta^4 & \zeta^7 & \zeta^2 & \zeta^5 \\ 1 & \zeta^5 & \zeta^2 & \zeta^7 & \zeta^4 & \zeta & \zeta^6 & \zeta^3 \\ 1 & \zeta^7 & \zeta^6 & \zeta^5 & \zeta^4 & \zeta^3 & \zeta^2 & \zeta^1 \end{array}\right] =: \begin{bmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{bmatrix}$$

Wir untersuchen die einzelnen Blöcke: Wegen $\zeta = \zeta_8$ ist $\zeta^2 = \zeta_4$, d.h.

$$B_{1,1} = B_{1,2} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \zeta_4 & \zeta_4^2 & \zeta_4^3 \\ 1 & \zeta_4^2 & \zeta_4^4 & \zeta_4^6 \\ 1 & \zeta_4^3 & \zeta_4^6 & \zeta_4^9 \end{bmatrix} = F_4.$$

Aus den Spalten 0,1,2 bzw. 3 von $B_{2,1}$ „klammern“ wir ζ^0, ζ^1, ζ^2 bzw. ζ^3 „aus“:

$$B_{2,1} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \zeta^2 & \zeta^4 & \zeta^6 \\ 1 & \zeta^4 & 1 & \zeta^4 \\ 1 & \zeta^6 & \zeta^4 & \zeta^2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \zeta & 0 & 0 \\ 0 & 0 & \zeta^2 & 0 \\ 0 & 0 & 0 & \zeta^3 \end{bmatrix} = F_4 D_4.$$

Analog:

$$B_{2,2} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \zeta^2 & \zeta^4 & \zeta^6 \\ 1 & \zeta^4 & 1 & \zeta^4 \\ 1 & \zeta^6 & \zeta^4 & \zeta^2 \end{bmatrix} \begin{bmatrix} \zeta^4 & 0 & 0 & 0 \\ 0 & \zeta^5 & 0 & 0 \\ 0 & 0 & \zeta^6 & 0 \\ 0 & 0 & 0 & \zeta^7 \end{bmatrix} = F_4(\zeta^4 D_4) = -F_4 D_4.$$

Insgesamt erhalten wir

$$PF_8 = \begin{bmatrix} F_4 & F_4 \\ F_4 D_4 & -F_4 D_4 \end{bmatrix} = \begin{bmatrix} F_4 & O \\ O & F_4 \end{bmatrix} \begin{bmatrix} I_4 & I_4 \\ D_4 & -D_4 \end{bmatrix}.$$

Satz 6.18. *Seien m gerade, σ die folgende (even/odd) Permutation*

$$\sigma = [0,\, 2\, \ldots, m-2, 1, 3, \ldots, m-1]$$

und $P = P_\sigma$ die zugehörige Permutationsmatrix.
Dann besitzt die zeilenpermutierte Fourier-Matrix F_m die Zerlegung

$$PF_m = \begin{bmatrix} F_{m/2} & F_{m/2} \\ F_{m/2}D_{m/2} & -F_{m/2}D_{m/2} \end{bmatrix} = \begin{bmatrix} F_{m/2} & O \\ O & F_{m/2} \end{bmatrix} \begin{bmatrix} I_{m/2} & I_{m/2} \\ D_{m/2} & -D_{m/2} \end{bmatrix}.$$

Dabei bezeichnet $D_{m/2}$ die Diagonalmatrix

$$D_{m/2} = \operatorname{diag}\left(\zeta_m^0, \zeta_m^1, \ldots, \zeta_m^{m/2-1}\right) \in \mathbb{C}^{(m/2)\times(m/2)}$$

mit $\zeta_m = \overline{\omega_m} = e^{-2\pi i/m}$.

Berechne jetzt $\boldsymbol{y} = F_m \boldsymbol{x}$ für ein $\boldsymbol{x} \in \mathbb{C}^m$ (m gerade). Gemäß der Zerlegung von F_m aus Satz 6.18 unterteilen wir dies in zwei Schritte:

1. Reduktionsschritt: Berechne

$$\boldsymbol{z} = \begin{bmatrix} I_{m/2} & I_{m/2} \\ D_{m/2} & -D_{m/2} \end{bmatrix} \boldsymbol{x}.$$

Im Fall $m = 8$ ergibt sich:

$$z_0 = x_0 + x_4, \quad z_1 = x_1 + x_5, \quad z_2 = x_2 + x_6, \quad z_3 = x_3 + x_7$$
$$z_4 = (x_0 - x_4), \quad z_5 = (x_1 - x_5)\zeta_m, \quad z_6 = (x_2 - x_6)\zeta_m^2, \quad z_7 = (x_3 - x_7)\zeta_m^3$$

($m/2$ komplexe Multiplikationen und m komplexe Additionen).

2. Teilprobleme: Berechne

$$F_{m/2}\boldsymbol{z}(0 : m/2 - 1) \quad \text{und} \quad F_{m/2}\boldsymbol{z}(m/2 : m - 1)$$

(zwei Fourier-Transformationen der Dimension $m/2$).

Ist $m = 2^p$ eine Zweierpotenz, so ist $m/2$ ebenfalls gerade und die beiden Fourier-Transformationen der Dimension $m/2$ können auf vier Fourier-Transformationen der Dimension $m/4$ reduziert werden.

Der Aufwand zur Reduktion beträgt $2 \cdot m/4 = m/2$ komplexe Multiplikationen (und $2 \cdot m/2 = m$ komplexe Additionen). Dieser Prozess wird solange fortgesetzt bis man eine Multiplikation mit F_m auf m Multiplikationen mit $F_1 = [1]$ reduziert hat (eine Multiplikation mit F_1 erfordert offenbar keinen Aufwand).

Dieses Reduktionsverfahren heißt schnelle Fourier-Transformation (FFT = Fast Fourier Transform).

Satz 6.19. *Zur Durchführung einer schnellen Fourier-Transformation der Ordnung $m = 2^p$ sind*

$$\frac{m}{2}p = \frac{m}{2}\log_2(m) \quad \textit{komplexe Multiplikationen}$$

und $m \log_2(m)$ komplexe Additionen erforderlich.

Die naive Berechnung einer Fourier-Transformation der Länge $m = 2^p$ durch $F_m \boldsymbol{x}$ erfordert also $2^{p+1}/p$-mal mehr Multiplikationen als ihre Berechnung durch FFT. Wenn z.B. für $p = 20$ die FFT-Version eine Sekunde benötigt, so benötigt $F_m \boldsymbol{x}$ etwa 29 Stunden.

Verbleibendes Problem: Bestimmt man $\boldsymbol{y} = F_m \boldsymbol{x}$ durch FFT, so erhält man zunächst eine permutierte Version $\tilde{\boldsymbol{y}} = Q\boldsymbol{y}$ von $\boldsymbol{y}$ mit einer Permutationsmatrix $Q \in \mathbb{R}^{m \times m}$.

Es gilt: Besitzt für $m = 2^p$ der Index $i \in \{0, 1, \ldots, m-1\}$ die Binärdarstellung $i = b_{p-1}2^{p-1} + \cdots + b_2 2^2 + b_1 2 + b_0 =: [b_{p-1} \ \ldots \ b_2 \, b_1 \, b_0]_2$, und ist

$$r(i) := [b_0 \, b_1 \, b_2 \ \ldots \ b_{p-1}]_2 = b_0 2^{p-1} + b_1 2^{p-2} + b_2 2^{p-3} + \cdots + b_{p-1}$$

(bit reversal), dann gelten

$$y_i = \tilde{y}_{r(i)} \quad \text{und} \quad \tilde{y}_i = y_{r(i)}.$$

6.6 Anwendungen der FFT

Schnelle Berechnung einer Faltung: Sei

$$\mathscr{S}_m := \{\boldsymbol{x} = \{\ldots, x_0, x_1, \ldots, x_{m-1}, \ldots\} : x_j \in \mathbb{C}\}$$

der Raum der doppelseitigen m-periodischen Folgen. $\mathscr{S}_m$ ist isomorph zum $\mathbb{C}^m$. Auf $\mathscr{S}_m$ sind zwei Multiplikationen definiert:

Hadamard-Produkt: $[\boldsymbol{x} \odot \boldsymbol{y}]_k = x_k y_k,$

Faltung oder Cauchy-Produkt: $[\boldsymbol{x} * \boldsymbol{y}]_k = \sum_{j=0}^{m-1} x_j y_{k-j}.$

Lemma 6.20 (Faltungssatz). *Für $\boldsymbol{x}, \boldsymbol{y} \in \mathscr{S}_m$ gelten*

$$\begin{aligned} F_m(\boldsymbol{x} * \boldsymbol{y}) &= (F_m \boldsymbol{x}) \odot (F_m \boldsymbol{y}), \\ m\, F_m(\boldsymbol{x} \odot \boldsymbol{y}) &= (F_m \boldsymbol{x}) * (F_m \boldsymbol{y}). \end{aligned}$$

Ist $m = 2^p$, so kann die Faltung wegen

$$\boldsymbol{x} * \boldsymbol{y} = F_m^{-1}\left[(F_m\boldsymbol{x}) \odot (F_m\boldsymbol{y})\right] = \frac{1}{m}\bar{F}_m\left[(F_m\boldsymbol{x}) \odot (F_m\boldsymbol{y})\right]$$

durch drei FFT's, also mit $m(1.5\log_2(m) + 1)$ komplexen Multiplikationen, bestimmt werden (konventionelle Berechnung erfordert m^2 Multiplikationen). Dies wird zur Multiplikation großer ganzer Zahlen und zur Multiplikation von Polynomen eingesetzt.

Eine Matrix $A \in \mathbb{C}^{m\times m}$ heißt zirkulant, wenn sie die Form

$$A = \text{circul}(a_0, \ldots, a_{m-1}) = \begin{bmatrix} a_0 & a_1 & \cdots & a_{m-2} & a_{m-1} \\ a_{m-1} & a_0 & \cdots & a_{m-3} & a_{m-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_2 & a_3 & \cdots & a_0 & a_1 \\ a_1 & a_2 & \cdots & a_{m-1} & a_0 \end{bmatrix}$$

besitzt.

Lemma 6.21. *Mit Hilfe der (zirkulanten) Shiftmatrix*

$$S_m := \operatorname{circul}(0, 1, 0 \ldots, 0) \in \mathbb{C}^{m \times m}$$

kann jede zirkulante Matrix $A = \operatorname{circul}(a_0, a_1, \ldots, a_{m-1})$ *in der Form*

$$A = p(S_m) = a_0 I_m + a_1 S_m + a_2 S_m^2 + \cdots a_{m-1} S_m^{m-1},$$

d.h. als Polynom in S_m*, geschrieben werden.*

Die Eigenwerte λ_j *von* A *sind deshalb durch*

$$\lambda_j = p\left(\omega_m^j\right) = p\left(\exp\left(2\pi i j/m\right)\right) \qquad (j = 0, 1, \ldots, m-1)$$

gegeben.

$$\boldsymbol{v}_j = \left[\omega_m^0, \omega_m^j, \omega_m^{2j}, \ldots, \omega_m^{(m-1)j}\right]^\top$$

ist ein zugehöriger Eigenvektor.

Lemma 6.22. *Seien* $\boldsymbol{a} = [a_0, a_1, \ldots, a_{m-1}]^\top \in \mathbb{C}^m$, $A = \operatorname{circul}(\boldsymbol{a})$ *und* $\boldsymbol{x} \in \mathbb{C}^m$. *Dann ist*

$$A^\top \boldsymbol{x} = \boldsymbol{a} * \boldsymbol{x}.$$

Satz 6.23. *Seien* $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{C}^m$ *und* $A = \operatorname{circul}(\boldsymbol{a})$.
Dann gelten:

(a) $\det(A) \neq 0 \Leftrightarrow$ *alle Komponenten von* $F_m \boldsymbol{a}$ *sind von 0 verschieden.*

(b) Das LGS $A^\top \boldsymbol{x} = \boldsymbol{b}$ *ist genau dann lösbar, wenn* $[F_m \boldsymbol{a}]_j = 0$ *stets* $[F_m \boldsymbol{b}]_j = 0$ *impliziert (*$j = 0, 1, \ldots, m-1$*).*

(c) Ist $A^\top \boldsymbol{x} = \boldsymbol{b}$ *lösbar, so gilt für jede Lösung* $\boldsymbol{x}^*$*:*

$$[F_m \boldsymbol{x}^*]_j = [F_m \boldsymbol{b}]_j / [F_m \boldsymbol{a}]_j$$

für alle $j \in \{0, 1, \ldots, m-1\}$ *mit* $[F_m \boldsymbol{a}]_j \neq 0$. *Ist* $[F_m \boldsymbol{a}]_j = 0$ *(und folglich* $[F_m \boldsymbol{b}]_j = 0$*), so kann* $[F_m \boldsymbol{x}^*]_j$ *beliebig gewählt werden.*

Mit Hilfe der FFT kann das Produkt einer zirkulanten Matrix der Dimension m mit einem Vektor also in nur $m(1.5\log_2(m)+1)$ komplexen Multiplikationen berechnet werden (vgl. Lemma 6.22).

Darüberhinaus kann ein m-dimensionales lineares Gleichungssystem $A^\top \boldsymbol{x} = \boldsymbol{b}$ mit einer zirkulanten Koeffizientenmatrix

$$A^\top, \quad A = \operatorname{circul}(\boldsymbol{a}),$$

i.W. durch 3 FFT's (in ebenfalls $m(1.5\log_2(m)+1)$ komplexen Multiplikationen) gelöst werden (vgl. Satz 6.23): Ist A invertierbar, so gilt

$$A^{-\top}\boldsymbol{b} = \frac{1}{m}\overline{F_m}\left((F_m\boldsymbol{b})\,./\,(F_m\boldsymbol{a})\right),$$

wobei ./ komponentenweise Division bezeichnet.

6.7 Mustererkennung und Rekonstruktion von Signalen

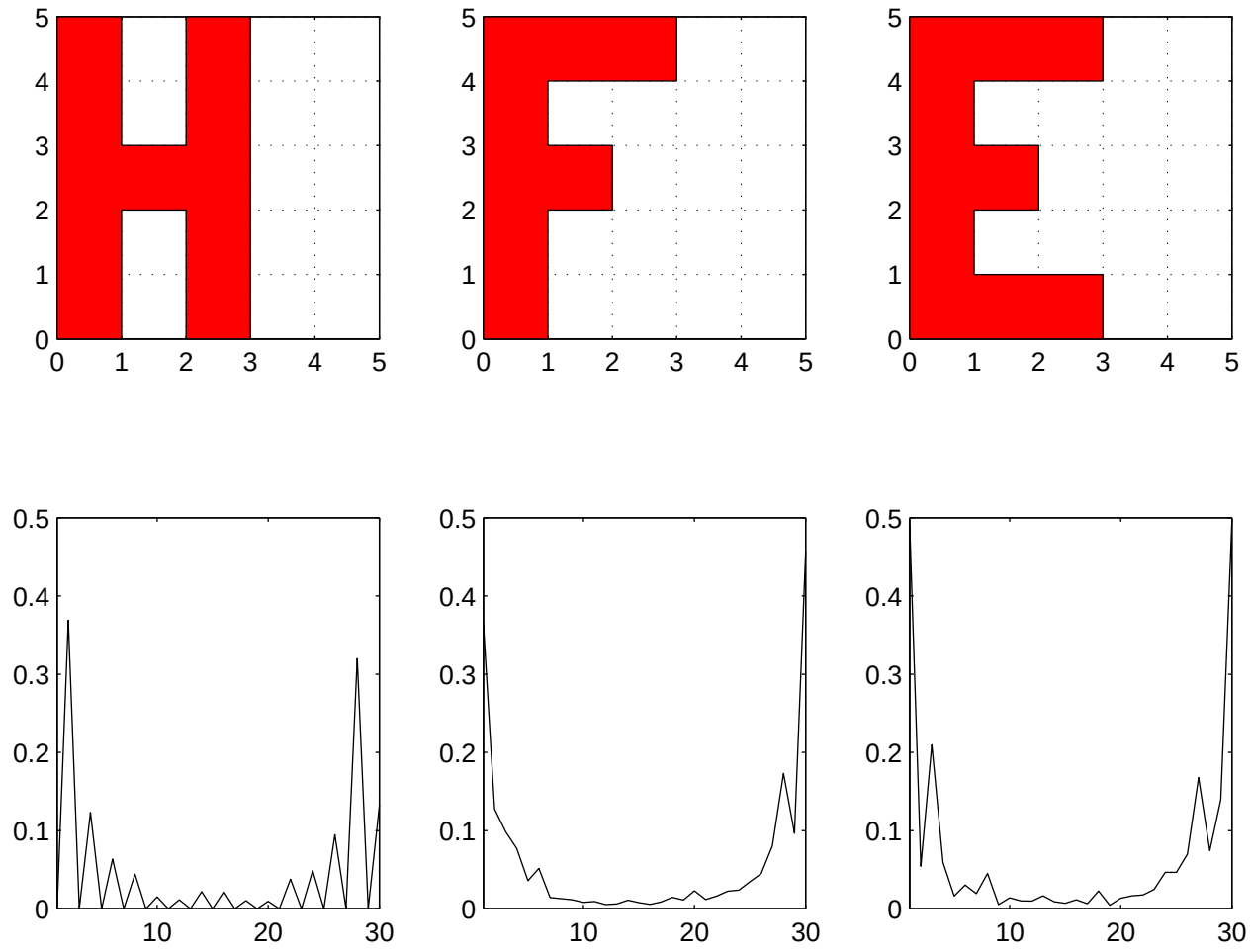
Interpretiere die m Ecken eines Polygons, $(x_0, y_0), \ldots, (x_{m-1}, y_{m-1})$, als komplexe Zahlen: $f_0 = x_0 + iy_0, \ldots, f_{m-1} = x_{m-1} + iy_{m-1}$ ($i^2 = -1$).

Das Ergebnis einer diskreten Fourier-Analyse dieser Zahlen

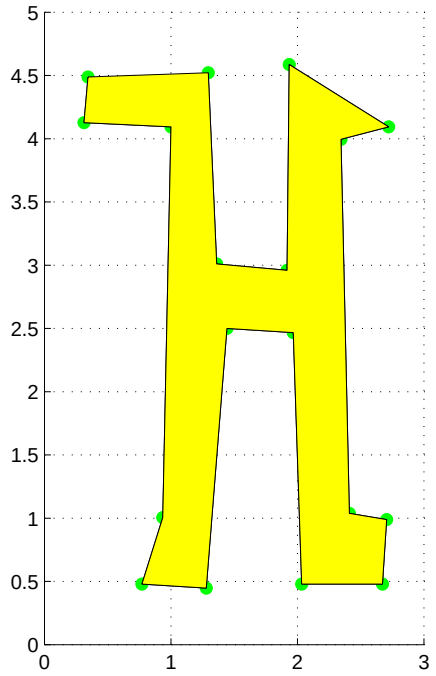
$$[c_0, c_1, \ldots, c_{m-1}]^\top = \frac{1}{m} F_m [f_0, f_1, \ldots, f_{m-1}]^\top$$

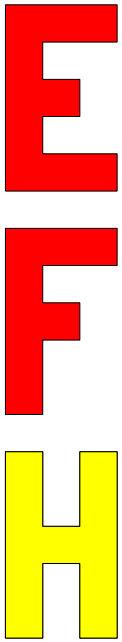
nennt man diskretes komplexes Spektrum des Polygons. Es spiegelt geometrische Eigenschaften des Polygons wider und kann daher zur Klassifikation von Formen (Mustererkennung) verwendet werden.
Lage- und größenunabhängig ist das normierte Amplitudenspektrum

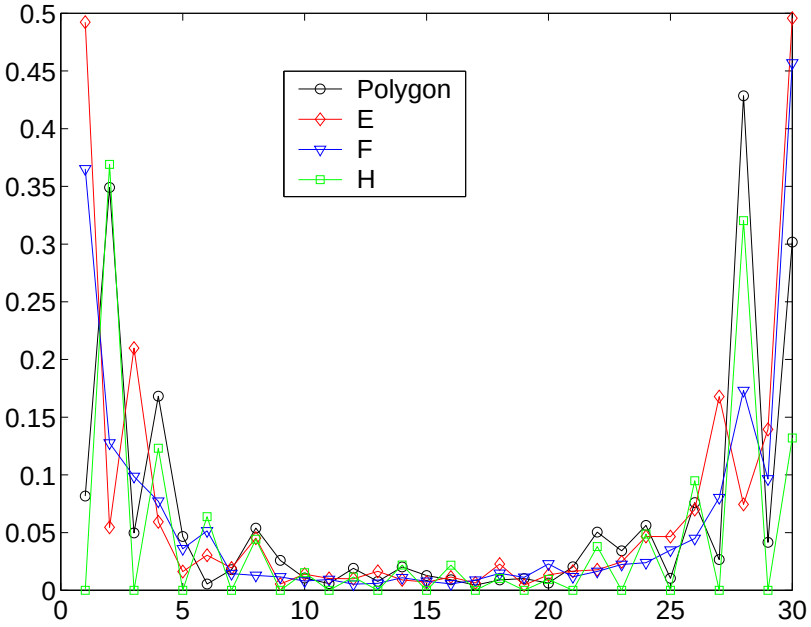
$$a_k = |c_{k+2}/c_1| \qquad (k = 0, \ldots, m-3).$$

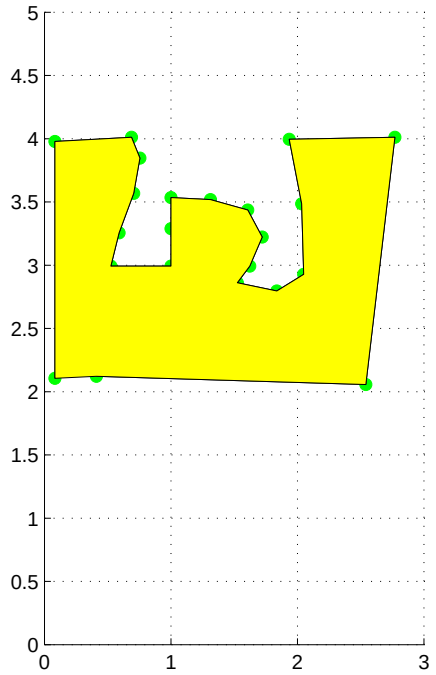


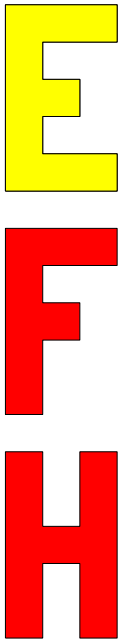
$$a_1 < a_2,\ a_2 > a_3;\ a_1 > a_2,\ a_2 > a_3;\ a_1 > a_2,\ a_2 < a_3;$$

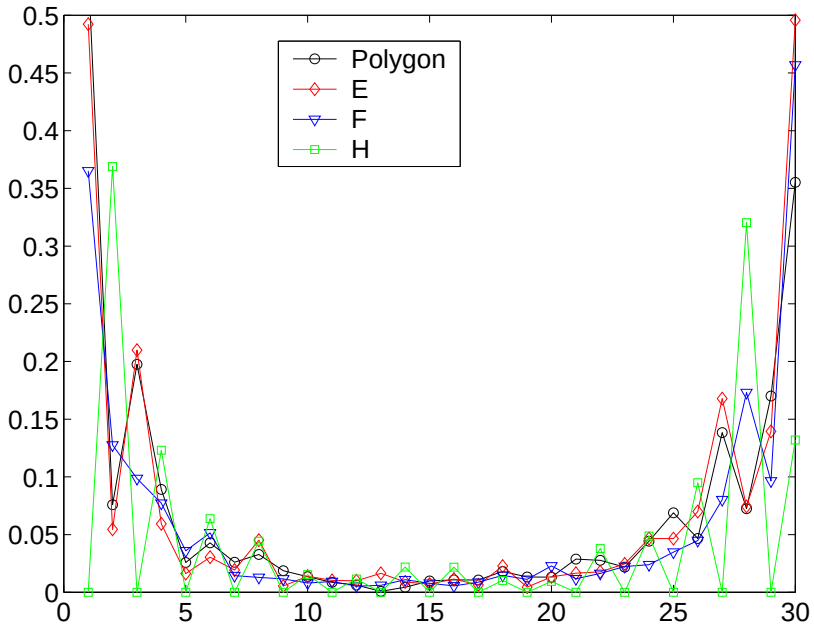

0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
0
1
2
3

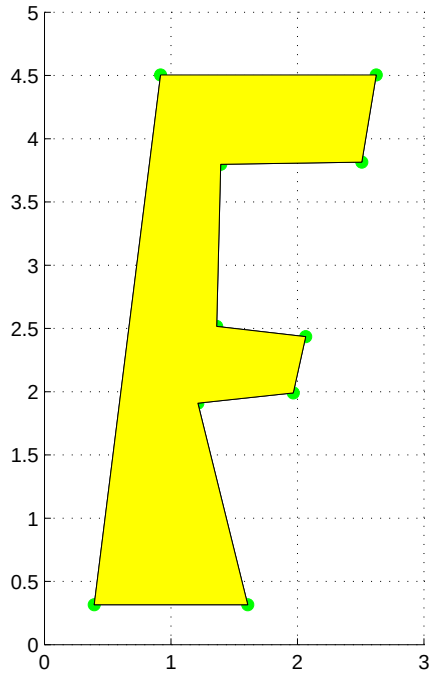


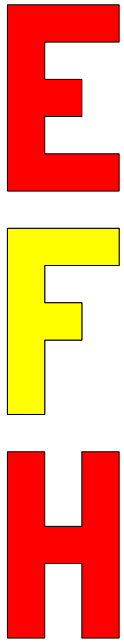

Polygon
E
F
H
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0
5
10
15
20
25
30

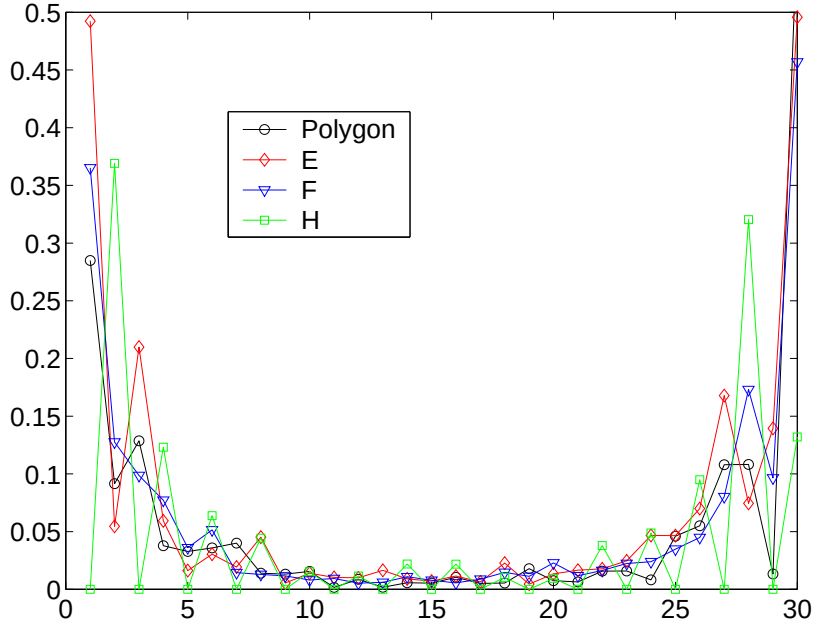

5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
0
1
2
3




Polygon
E
F
H
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
15
20
25
30

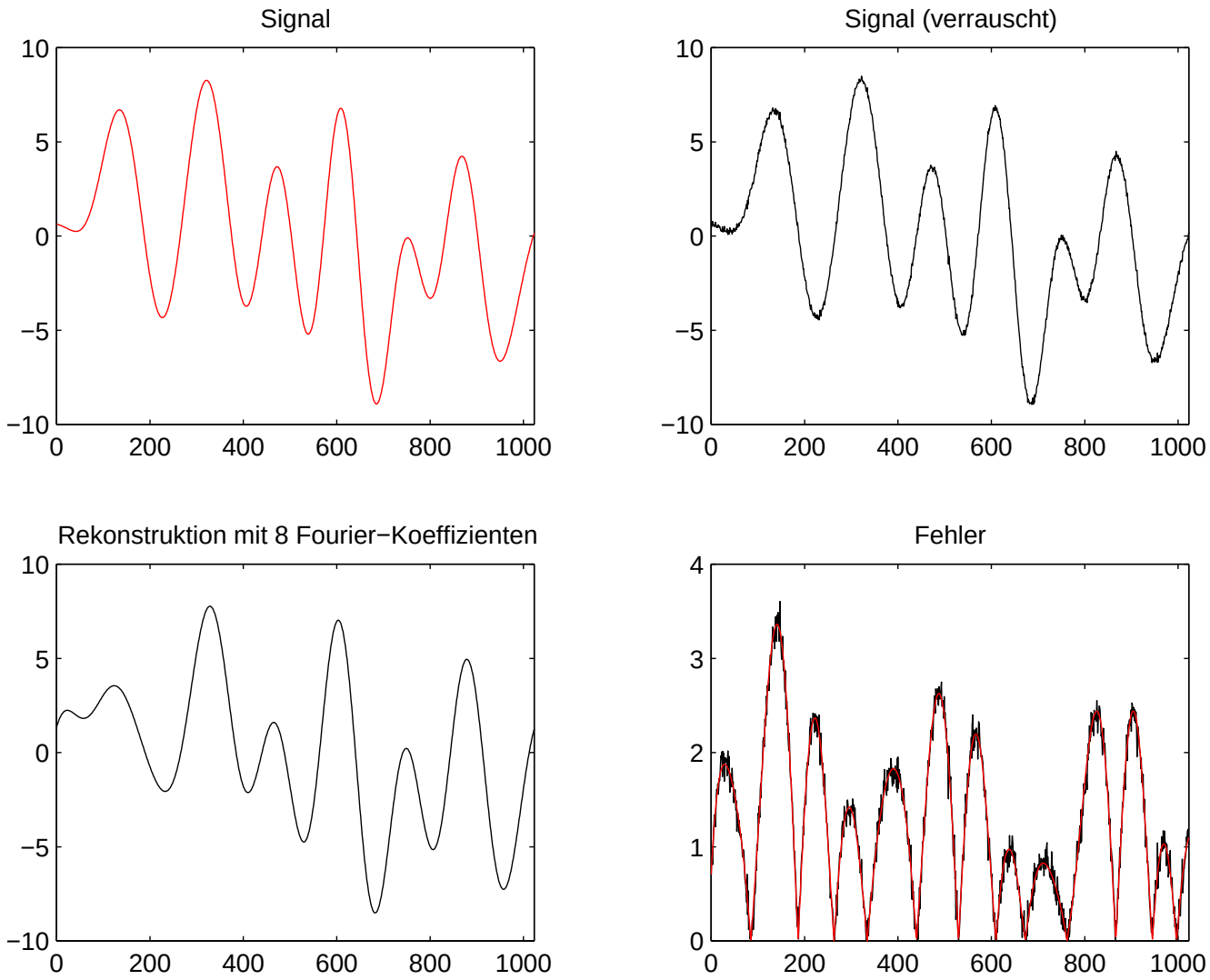

0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
0
1
2
3

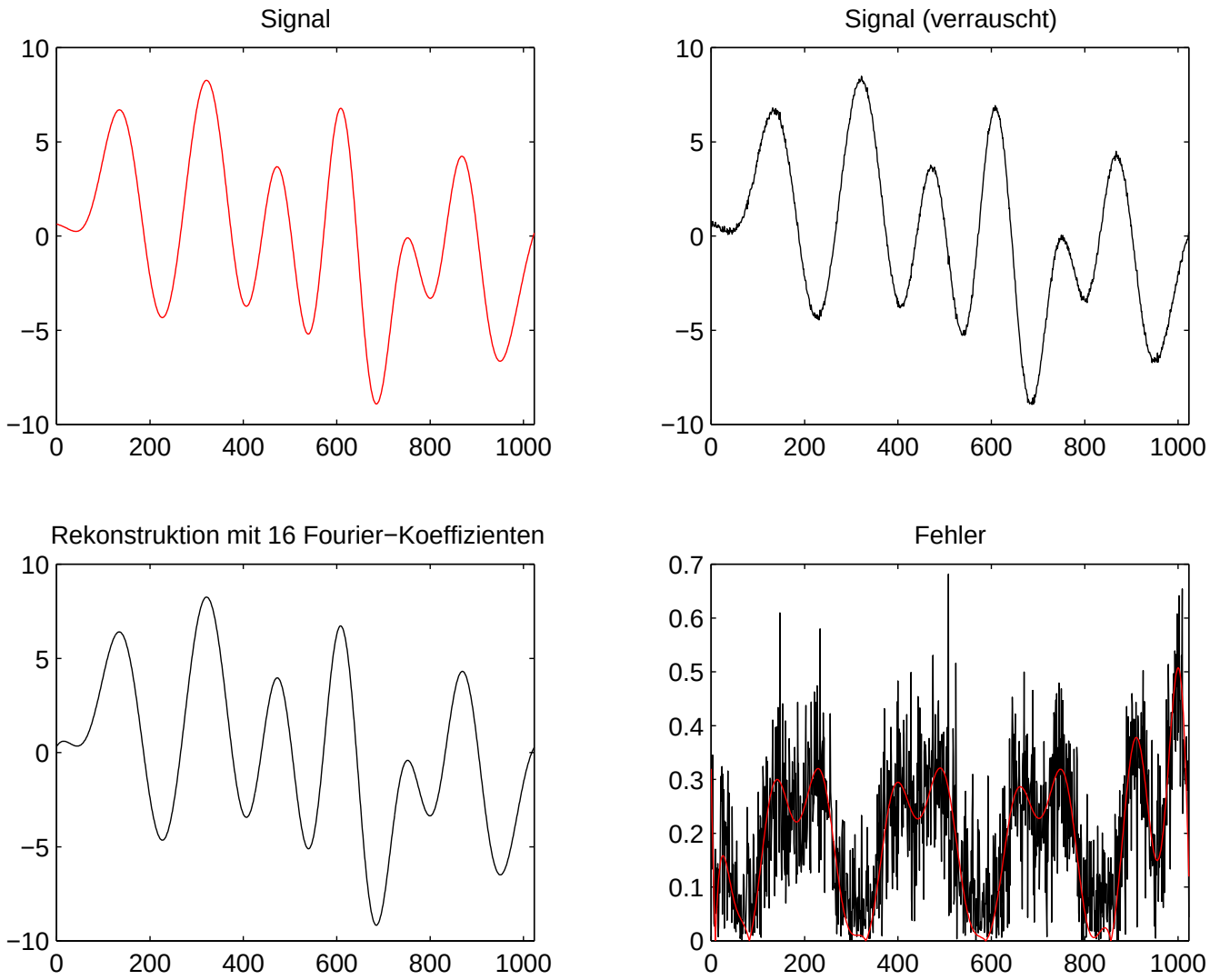


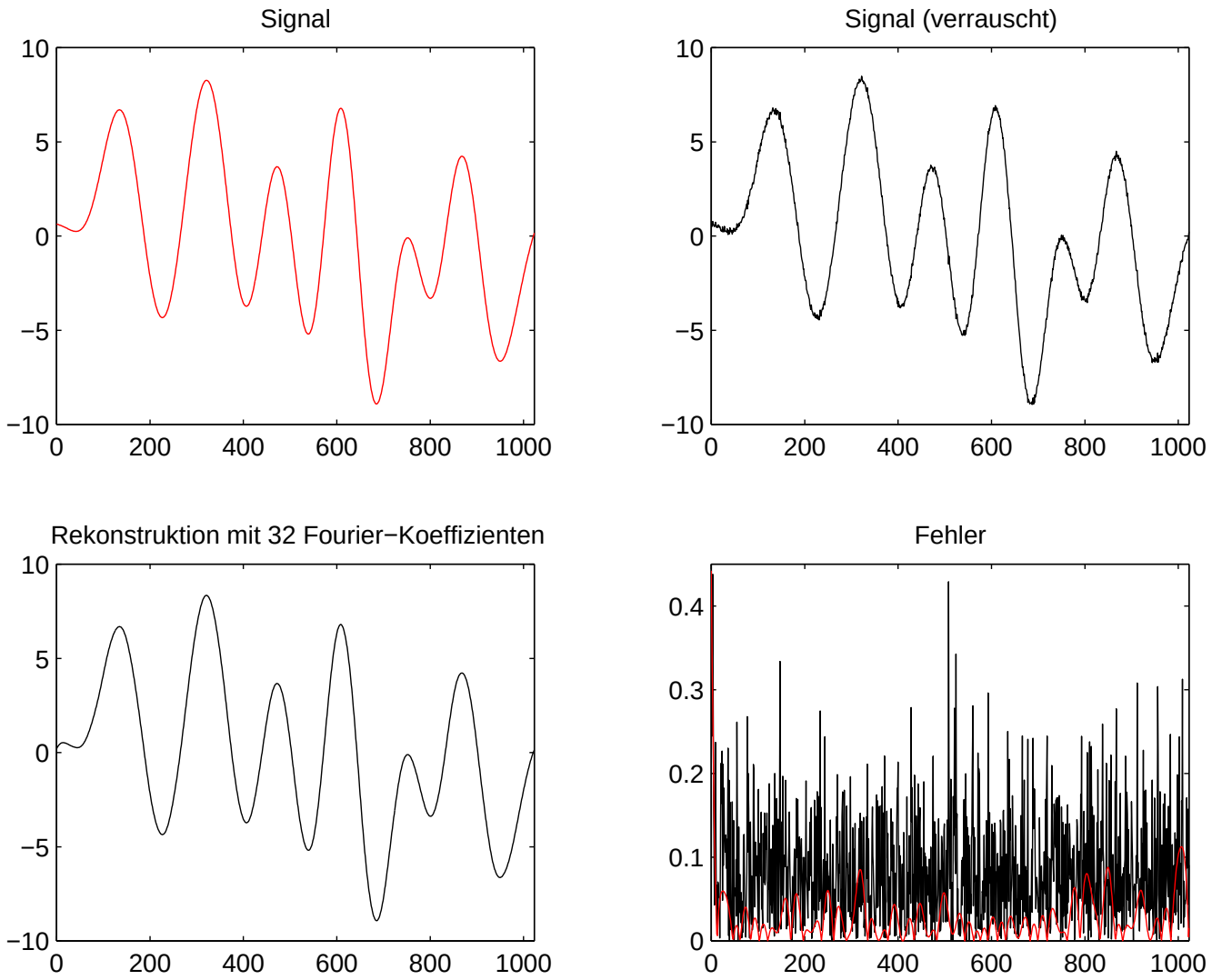

Polygon
E
F
H
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0
5
10
15
20
25
30

```
Gegeben: Signal f (Dimension m = 1024),
       k = 8 (16, 32).
Aufgabe: Unterlege f mit Rauschen, bestimme die
       k groessten Fourier-Koeffizienten und
       rekonstruiere aus diesen das Signal.

   f_rausch = f + .1*randn(size(f));
   c = fft(f_rausch);
   [ignore,j] = sort(abs(c));
   ind = [m-k+1:m];
   c_compr = zeros(size(c));
   c_compr(j(ind)) = c(j(ind));
   recon = ifft(c_compr);
```


Signal
Signal (verrauscht)
Rekonstruktion mit 8 Fourier−Koeffizienten
Fehler
10
5
0
−5
−10
4
3
2
1
0
0
200
400
600
800
1000


Signal
Signal (verrauscht)
Rekonstruktion mit 16 Fourier−Koeffizienten
Fehler
10
5
0
−5
−10
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
200
400
600
800
1000


Signal
Signal (verrauscht)
Rekonstruktion mit 32 Fourier−Koeffizienten
Fehler
10
5
0
−5
−10
0.4
0.3
0.2
0.1
0
0
200
400
600
800
1000