

Vorlesung 401-1652-10L : Numerische Mathematik I

Frühlingssemester 2014 : Prof. Ralf Hiptmair (SAM, D-MATH)

[Vorlesungsniederschrift, Rohformat, URL: <http://www.math.ethz.ch/education/bachelor/lectures/fs2014/math/nm>]

Kapitel 3 : Polynominterpolation

3. A. Abstrakte Interpolation in 1D

Gegeben : Knoten (Stützstellen) $N = \{x_0, \dots, x_n\}$, $n \in N$, $x_0 < x_1 < \dots < x_n$

Daten (Stützwerte) $\mathbf{y} = (y_0, \dots, y_n)^T \in \mathbb{R}^{n+1}$

Funktionsraum $V \subset C^0([x_0, x_n])$, endlichdimensional, $\dim V < \infty$

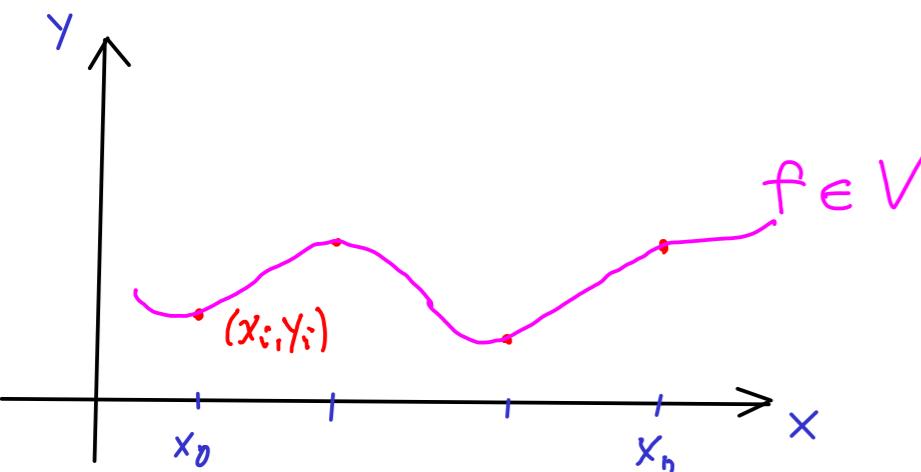
Bsp : $V = P_n := \{x \rightarrow \sum_{j=0}^n c_j x^j, c_j \in \mathbb{R}\}$ (Polynome vom Grad $\leq n$), $\dim P_n = n+1$

$V = \{x \rightarrow \sum_{j=0}^n c_j e^{\lambda_j x}, c_j \in \mathbb{R}, \lambda_j \in \mathbb{R} \text{ gegeben, paarweise verschieden}\}$

$V = \{x \rightarrow \sum_{j=0}^n c_j e^{2\pi i j x}, c_j \in \mathbb{C}\}$ (trigonometrische Polynome)

Gesucht ("Interpolationsaufgabe") : $f \in V : f(x_i) = y_i, i=0, \dots, n$ (IPB)

↳ Interpolant



Anwendung: Kennlinien aus Messdaten

What is a function $f: I \subset \mathbb{R} \mapsto \mathbb{R}$ "from the perspective of a computer"?

```

1 class Function {
2   private:
3     // various internal data describing f
4   public:
5     // Constructor: accepts information necessary for specifying the
       // function
6     Function(/* ... */);
7     // Evaluation operator
8     double operator () (double t) const;
9 };

```

Allgemeines Vorgehen zur Lösung der Interpolationsaufgabe:

① Wahl einer Basis $\mathcal{B} = \{b_0, \dots, b_n\} \subset V$ ($\dim V = n+1$) von V :

Ansatz: $f = \sum_{j=0}^n c_j b_j$ mit unbekannten Koeffizienten $c_j \in \mathbb{R}$ (3.A.1)

② (IPB) \Rightarrow LGS:
$$\begin{pmatrix} b_0(x_0) & \cdots & b_n(x_0) \\ b_0(x_1) & \cdots & b_n(x_1) \\ \vdots & & \vdots \\ b_0(x_n) & \cdots & b_n(x_n) \end{pmatrix} \begin{pmatrix} c_0 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$
 (3.A.2)
 ↑
 Interpolationsmatrix / Punktauswertungsmatrix

③ Auswertung: $f(t) = \sum_{j=0}^n c_j b_j(t)$ für (viele) $t \in [x_0, x_n]$

- * Kriterien:
- Gute konditionierte Interpolationsmatrix
 - (3.A.2) einfach zu lösen
 - $b_j(t)$ einfach auszuwerten

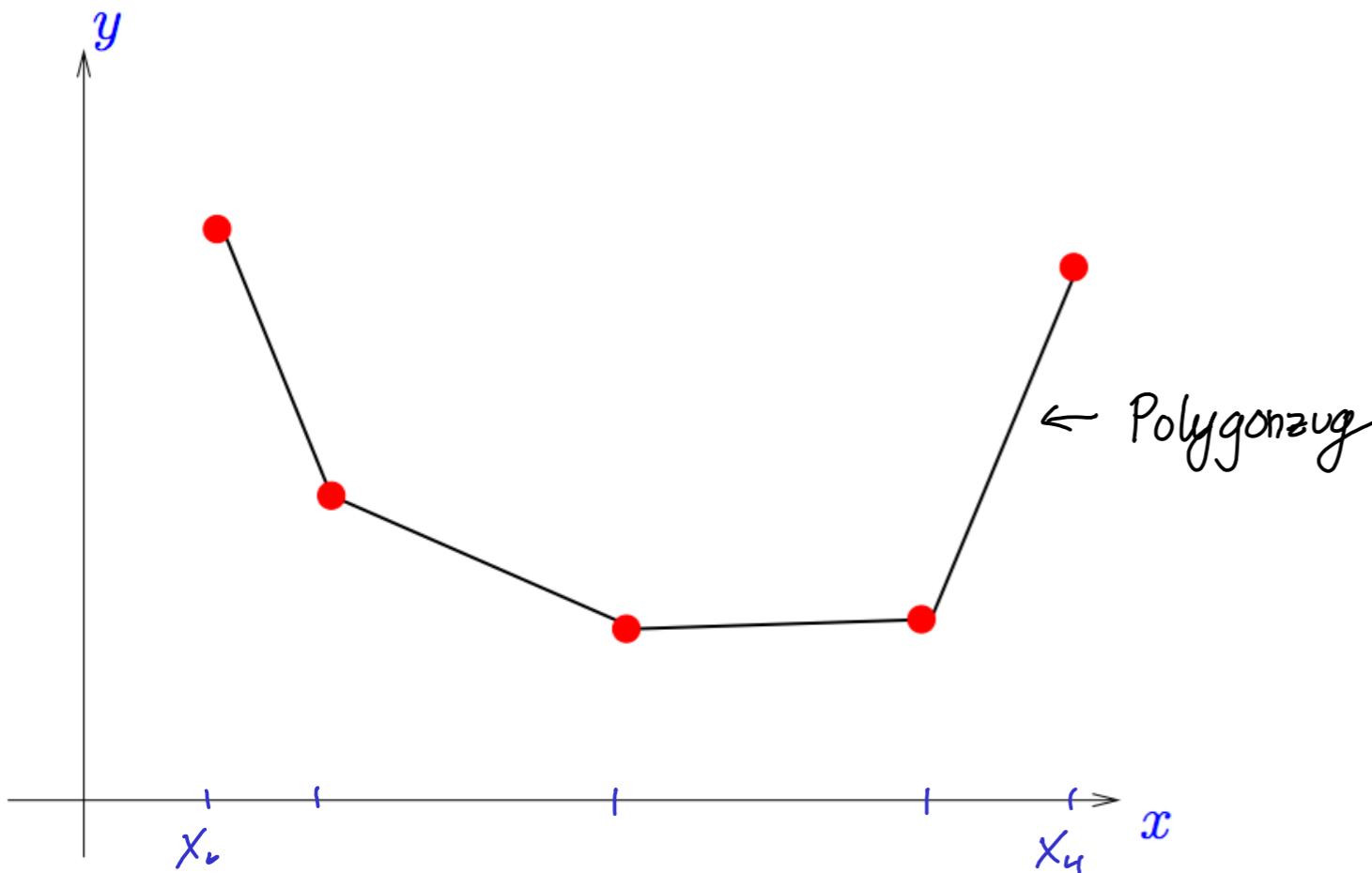
Bem: Interpolant f ist unabhängig von der Wahl von \mathcal{B} (bei exakter Arithmetik)

```

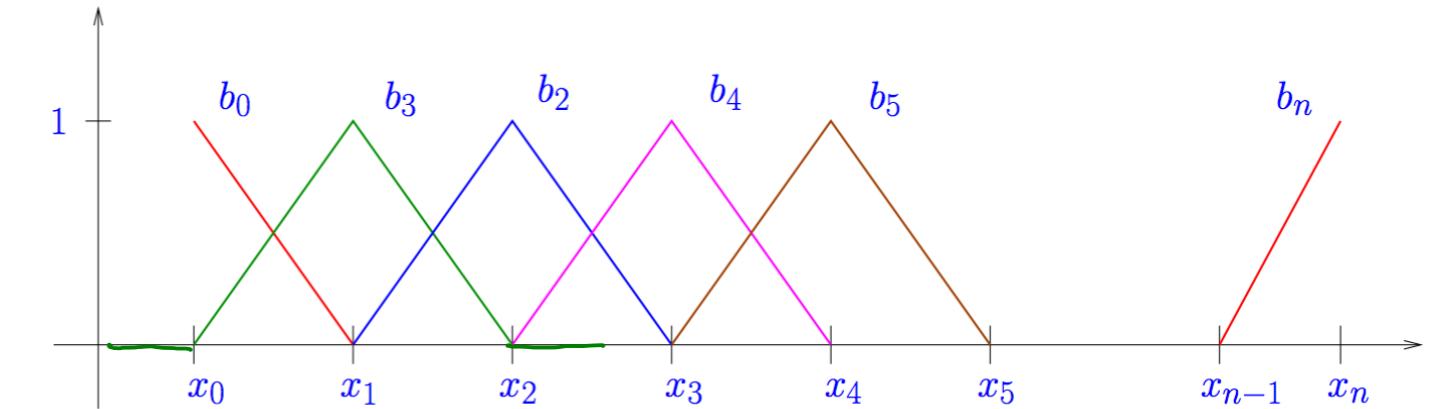
1 class Interpolant {
2   private:
3     // various internal data describing  $f$ 
4   public:
5     // Constructor: computation of coefficients  $c_j$  of representation (3.A.1)
6     Interpolant(const vector<double> &x, const vector<double> &y);
7     // Evaluation operator for interpolant  $f$ 
8     double operator () (double x) const;
9 }

```

Bsp : Stückweise lineare Interpolation $V := \{ g \in C^0([x_0, x_n]) : g|_{[x_j, x_{j+1}]} \text{ lineare Fkt.}, j=0, \dots, n-1 \}$



Basis : ("Dachfunktionen")



$$b_0(x) = \begin{cases} 1 - \frac{x-x_0}{x_1-x_0} & \text{für } x_0 \leq x < x_1, \\ 0 & \text{für } x \geq x_1. \end{cases}$$

$$b_j(x) = \begin{cases} 1 - \frac{x_j-x}{x_{j-1}-x_{j-1}} & \text{für } x_{j-1} \leq x < x_j, \\ 1 - \frac{x-x_j}{x_{j+1}-x_j} & \text{für } x_j \leq x < x_{j+1}, \\ 0 & \text{sonst in } [x_0, x_n]. \end{cases} \quad j = 1, \dots, n-1,$$

$$b_n(x) = \begin{cases} 1 - \frac{x_n-x}{x_{n-1}-x_{n-1}} & \text{für } x_{n-1} \leq x < x_n, \\ 0 & \text{für } x < x_{n-1}. \end{cases}$$

Kardinalbasis : $b_j(x_i) = \delta_{ij} = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$

↳ Interpolationsmatrix = Einheitsmatrix

Interpolation als lineare Abbildung

Punktauswertungsabbildung : $S_{V,N} : \begin{cases} V & \rightarrow \mathbb{R}^{n+1} \\ g & \mapsto (g(x_i))_{i=0}^n \end{cases}$ \Rightarrow Interpolationsoperator : (S invertierbar)
 $I_{V,N} := S^{-1} : \mathbb{R}^{n+1} \xrightarrow{\uparrow \text{Datenspace}} V$

Satz 3.A.3 (Invertierbarkeit der Punktauswertungsabbildung)

Gegeben sei eine endliche (angeordnete) Knotenmenge $\mathcal{N} = (x_0, \dots, x_n)$, $n \in \mathbb{N}$, und ein Funktionenraum $V \subset C^0([x_0, x_n])$.

Die Punktauswertungsabbildung $S_{\mathcal{N},V} : V \rightarrow \mathbb{R}^{n+1}$ ist genau dann invertierbar, wenn gilt:

- (i) $\dim V = n + 1$,
- (ii) Es gibt $b_0, \dots, b_n \in V$ mit $b_j(x_i) = \delta_{ij}$ (**Kardinalbasis**).

Beweis : $S b_j = e_j$ (j. Einheitsvektor) \Rightarrow Rang (S) = n+1

Korollar 3.A.4 (Existenz und Eindeutigkeit des Interpolanten)

Ist die Punktauswertungsabbildung $S_{\mathcal{N},V} : V \rightarrow \mathbb{R}^{n+1}$ invertierbar, so ist die Interpolationsaufgabe für alle Daten $(y_i)_{i=0}^n \in \mathbb{R}^{n+1}$ eindeutig lösbar.

▷ $S^{-1} =: I$ last Interpolationsaufgabe: $f = I y$

Satz 3.A.5 (Punktauswertungsmatrix)

Gegeben sei eine endliche (angeordnete) Knotenmenge $\mathcal{N} = (x_0, \dots, x_n)$, $n \in \mathbb{N}$, und ein Funktionenraum $V \subset C^0([x_0, x_n])$, aufgespannt von den Funktionen $g_0, \dots, g_n \in V$.

Dann ist die Punktauswertungsabbildung $S_{\mathcal{N},V} : V \rightarrow \mathbb{R}^{n+1}$ genau dann invertierbar, wenn die Matrix

$$(g_j(x_i))_{i,j=0}^{n+1} \in \mathbb{R}^{(n+1) \times (n+1)} = \begin{pmatrix} g_0(x_0) & \cdots & g_n(x_0) \\ \vdots & & \vdots \\ g_0(x_n) & \cdots & g_n(x_n) \end{pmatrix}$$

regulär ist.

▷ Wie oben :

$$f = \sum_{j=0}^n c_j g_j$$

$$\subseteq = ((g_j(x_i))_{i,j=0}^n)^{-1} \times$$

f löst Interpolationsaufgabe

Approximation durch Interpolation

Gegeben : V, \mathcal{N} , Funktion $g \in C^0([x_0, x_n]) \rightarrow$ Daten $y_i := f(x_i)$, dann Interpolation

Def. 3.A.6. (Nodaler Projektionsoperator) [$a := x_0, b := x_n$]

Zu einer endlichen Knotenmenge $\mathcal{N} \subset [a, b]$ und $V \subset C^0([a, b])$ für die die Interpoaltionsaufgabe eindeutig lösbar ist, also der Interpolationsoperator $I_{\mathcal{N},V}$ existiert, heisst

$$P_{\mathcal{N},V} := I_{\mathcal{N},V} \circ S_{\mathcal{N}} : C^0([a, b]) \rightarrow V \subset C^0([a, b])$$

der nodale Projektionsoperator. : $P_{\mathcal{N},V}^2 = P_{\mathcal{N},V}$

$$S_{\mathcal{N}} : \begin{cases} C^0([x_0, x_n]) \rightarrow \mathbb{R}^{n+1} \\ g \rightarrow (g(x_i))_{i=0}^n \end{cases}$$

$$[P^2 = (I \circ S) \circ (I \circ S) = I \circ \underbrace{(S \circ I)}_{= \text{Id wegen (IPB)}} \circ S = P]$$

Frage : $\| f - P_{\mathcal{N},V} f \|$, $\|\cdot\|$ Norm auf $C^0([a, b])$: Abhängigkeit von \mathcal{N}, V ?

3.1. Grundlagen: Polynominterpolation

Nun: $V = \mathbb{P}_n$, $\dim V = n+1$
 • Knotenmenge $\mathcal{N} = \{x_0 < x_1 < \dots < x_n\} \subset \mathbb{R}$

Kardinalbasis: $l_j(t) = \prod_{\substack{i=0 \\ i \neq j}}^n \frac{t-x_i}{x_j-x_i} \in \mathbb{P}_n : l_j(x_i) = \delta_{ij}$: Lagrange-Polygone zu \mathcal{N}

▷ [Satz 3.A.3]: \exists , Interpolationspolynoms

Algorithmen: (Berechnung Interpolationspolynom)

• Monomiale Basis: $\mathbb{P}_n = \text{Span } \mathcal{B}$, $\mathcal{B} := \{1, t, t^2, \dots, t^n\}$: $p(t) = \sum_{j=0}^n c_j t^j$, $c_j \in \mathbb{R}$, $t \in \mathbb{R}$

Auswertung: $p(t) = ((c_n x + c_{n-1}) x + c_{n-2}) x \dots + c_1 x + c_0$ Hornerschema

```

1 function y = horner(c,t)
2 % Evaluation of polynomial in monomial basis representation.
3 % Note: c(1) is the leading coefficient, c(end) the constant term
4 % (This follows the convention of MATLAB's polyval() function
5 - y = c(1); for j=2:length(c), y = t*y+c(j); end

```

▷ Führender Koeffizient in $c(1)$
 $c_0 \sim c(n+1)$, $n+1 = \text{length}(c)$

▷ Asymptotischer Rechenaufwand: $O(n)$ [optimal = Grösse der Daten c_j]

Interpolationsmatrix: $(X_i^j)_{i,j=0}^n$: Vandetmonomatrix

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & & & \vdots \\ \vdots & \vdots & & & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{pmatrix}$$

```

1 function p = ipolvander(x,y)
2 % Computation of polynomial interpolating in (x(i),y(i)) in monomial
3 % representation: UNSTABLE due to ill-conditioning of monomial basis
4 % Note: return vector p is a row vector with leading coefficient in
5 % p(1), agrees with MATLAB's convention for polyval().
6 n = length(x); if (length(y) ~= n), error('data size mismatch'); end
7 x = reshape(x,1,n); y = reshape(y,1,n); % data as row vectors
8 % Initialize interpolation matrix, here the Vandermonde matrix
9 v = repmat(x',1,n).^repmat((n-1:-1:0),n,1); → Initialisierung Vandermondematrix (Kosten: O(n²) )
10 p = (v\y')'; → Löse LGS ↴ [siehe Kapitel 2!]

```

▷ Rechenaufwand: $O(n^3)$

```

1 function v = ipolevalmonom(x,y,t)
2 % Evaluation of interpolating polynomial based on monomial representation
3 % WARNING: unstable due to ill conditioning of monomial basis
4 % x is a row vector of nodes, y a row vector of data, and t a row
5 % vector of evaluation points
6 c = ipolvander(x,y); % compute monomial coefficients
7 v = horner(c,t); % Evaluation by means of Horner scheme

```

Kardinalbasis: $\mathcal{B} = \{ l_i(t) := \prod_{\substack{j=0 \\ j \neq i}}^n \frac{t-x_j}{x_i-x_j} \}$ $\triangleright p(t) = \sum_{j=0}^n y_j l_j(t)$

Auswertung: $p(t) = \sum_{j=0}^n y_j \prod_{\substack{i=0 \\ i \neq j}}^n \frac{t-x_i}{x_j-x_i} = \sum_{j=0}^n \frac{y_j}{t-x_j} \omega_{n+1}(t) \lambda_j$ mit $\omega_{n+1}(t) := \prod_{\substack{k=0 \\ k \neq j}}^n (t-x_k)$
 $\lambda_j := \prod_{\substack{i=0 \\ i \neq j}}^n \frac{1}{x_j-x_i}$

$$= \omega_{n+1}(t) \sum_{j=0}^n \frac{\lambda_j}{t-x_j} y_j \quad (*)$$

↑
unabhängig von j

↓
unabhängig von t

Trick: $y_i = 1 \Rightarrow p(t) \equiv 1 \Rightarrow 1 = \omega_{n+1}(t) \sum_{j=0}^n \frac{\lambda_j}{t-x_j} \quad t \notin N$ $\Rightarrow \omega_{n+1}(t) = \frac{1}{\sum_{j=0}^n \frac{\lambda_j}{t-x_j}}$

In (*): $p(t) = \frac{\sum_{j=0}^n w_j(t) y_j}{\sum_{j=0}^n w_j(t)}$, $w_j(t) := \frac{\lambda_j}{t-x_j}$: Baryzentrische* Interpolationsformel

```

1 function p = ipolevalbarycentric(x,y,t)
2 % Evaluation of interpolating polynomial based on barycentric
3 % interpolation formula.
4 % x is a row vector of nodes, y a row vector of data, and t a row
5 % vector of evaluation points
6 n = length(x); % number of interpolation nodes = degree of polynomial -1
7 N = length(t); % Number of evaluation points stored in t
8 % Precompute the weights
9 for k = 1:n
10 lambda(k) = 1 / prod(x(k) - x([1:k-1,k+1:n])); end; →  $\lambda_j$  }
11 for i = 1:N → Schleife über Auswertungspunkte
12 % Compute quotient of weighted sums
13 z = (t(i)-x); j = find(abs(z) <= eps*norm(z)); ← Abfrage  $t \notin N$ , numerisch sauber
14 if (~isempty(j)), p(i) = y(j); % avoid division by zero
15 else
16 mu = lambda./z; p(i) = sum(mu.*y)/sum(mu); ↓ * Baryzentrische Summe = gewichtete Summe
17 end
18 end      ↓ mit  $\sum$  Gewichte = 1
          Berechnung  $w_j$  Rechenaufwand  $O(n)$ 

```

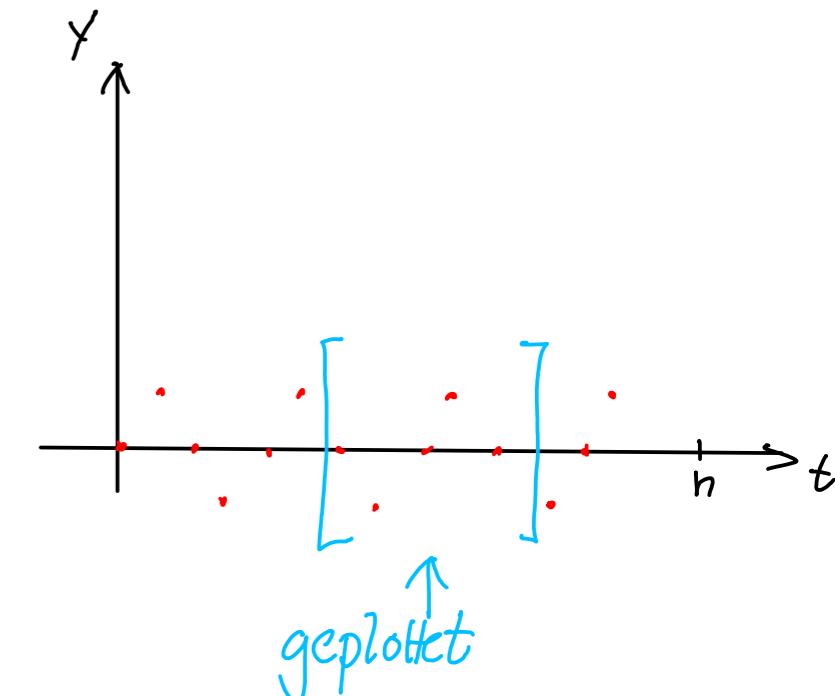
Rechenaufwand : $O(n^2)$ [für eine Polynomauswertung]
 $O(n^2 + Nn)$ [für Auswertung an $N \gg 1$ Punkten]

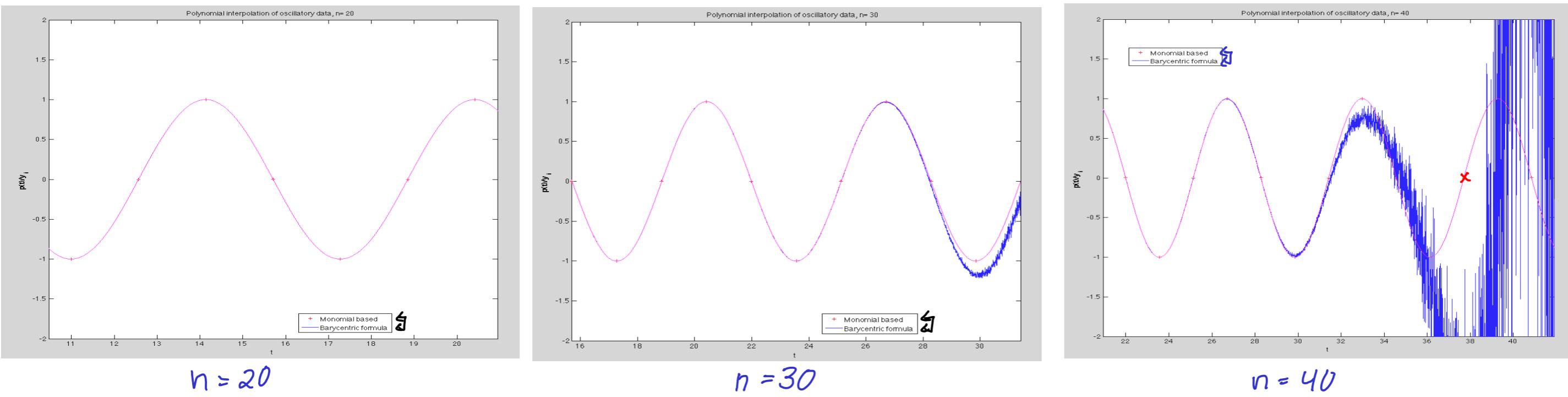
Bsp : Interpolation oszillativer Daten

```

1 function poliposdata(n,N)
2 % MATLAB function for polynomial interpolation of oscillatory data
3 % Argument n specifies the polynomial degree
4 if (nargin < 2), N = 10000; end;
5 t = linspace(0,n*pi/2,N); % For plotting
6
7 % Initialization of oscillatory data
8 x = linspace(0,n*pi/2,n+1); y = sin(x); →  $y_i \in \{-1, 0, 1\}$ 
9
10 % Two different ways to compute polynomial interpolant
11 v_monom = ipolevalmonom(x,y,t);
12 v_bary = ipolevalbarycentric(x,y,t);
13
14 % Graphical output
15 figure('name','Polynomial interpolation');
16 plot(x,y,'r+',t,v_monom,'b-',t,v_bary,'m-');
17 xmax = max(x); xmin = min(x);
18 axis([xmin*2/3+xmax/3,xmin/3+xmax*2/3,-2,2]);
19 set(gca,'fontsize',14);
20 xlabel('{\bf t}');
21 ylabel('{\bf p(t)/y_{i}}');
22 title(['Polynomial interpolation of oscillatory data, n= ', num2str(n)]);

```





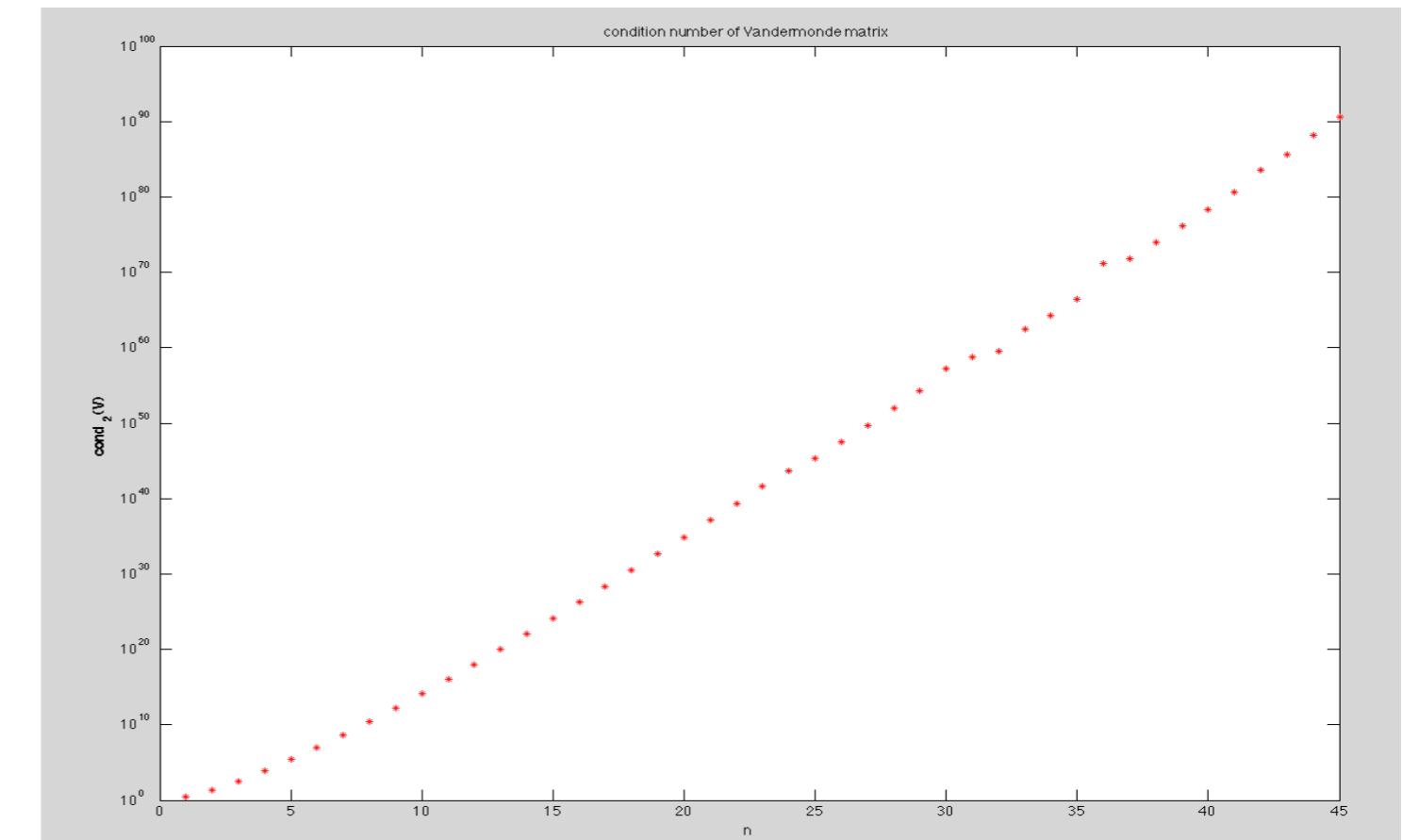
▷ Baryzentrisch (empirisch) stabil, Vandermondealgorithmus instabil

```

1 % Plot condition number of Vandermonde matrix
2 cn = [];
3 for n=1:45
4     x = linspace(0,n*pi/2,n+1);
5     V = repmat(x',1,n+1).^repmat((n:-1:0),n+1,1);
6     cn = [cn; n cond(V)];
7 end
8 figure; semilogy(cn(:,1),cn(:,2),'r*');
9 set(gca,'fontsize',14);
10 xlabel('{\bf n}');
11 ylabel('{\bf cond_2(V)}');
12 title('condition number of Vandermonde matrix');

```

- ▷ Vandermondematrix extrem schlecht konditioniert
- ▷ Grosse Fehler in berechneten Monomialkoeffizienten
- ▷ Monomialbasis numenisch schlecht



Polynominterpolation: Finde $p \in P_h$: $p(x_\ell) = y_\ell$ für gegebene $(x_\ell, y_\ell) \in \mathbb{R}^2$, $\ell = 0, \dots, n$

3.3. Das Aitken-Neville-Schema

Gegeben: $(x_\ell, y_\ell) \in \mathbb{R}^2$, $\ell = 0, \dots, n$, Gesucht: $p(t)$, $t \in \mathbb{R}$, mit $p \stackrel{?}{=} \text{Interpolationspolynom}$

Notation: $P_{j,k} \in P_{k-j}$, $0 \leq j \leq k \leq n$: Interpolationspolynom mit $p(x_\ell) = y_\ell$, $j \leq \ell \leq k$

Satz 3.3.A (Aitken-Neville-Rekursionsformel)

Zu einer Knotenmenge $\mathcal{N} = \{x_0, \dots, x_n\}$ und Stützwerten y_i , $i = 1, \dots, n$, sei $p_{j,k} \in P_{k-j}$, $0 \leq j \leq k \leq n$, das Interpolationspolynom durch die Punkte (x_i, y_i) , $j \leq i \leq k$. Dann gilt

$$p_{j,k}(t) = \begin{cases} y_j & , \text{ falls } k = j , \\ \frac{(t - x_j)p_{j+1,k}(t) + (x_k - t)p_{j,k-1}(t)}{x_k - x_j} & , \text{ falls } 0 \leq j < k \leq n . \end{cases}$$

Beachte:

$$P = P_{0,n}$$

Beweis: "Induktion", $k = j \vee$

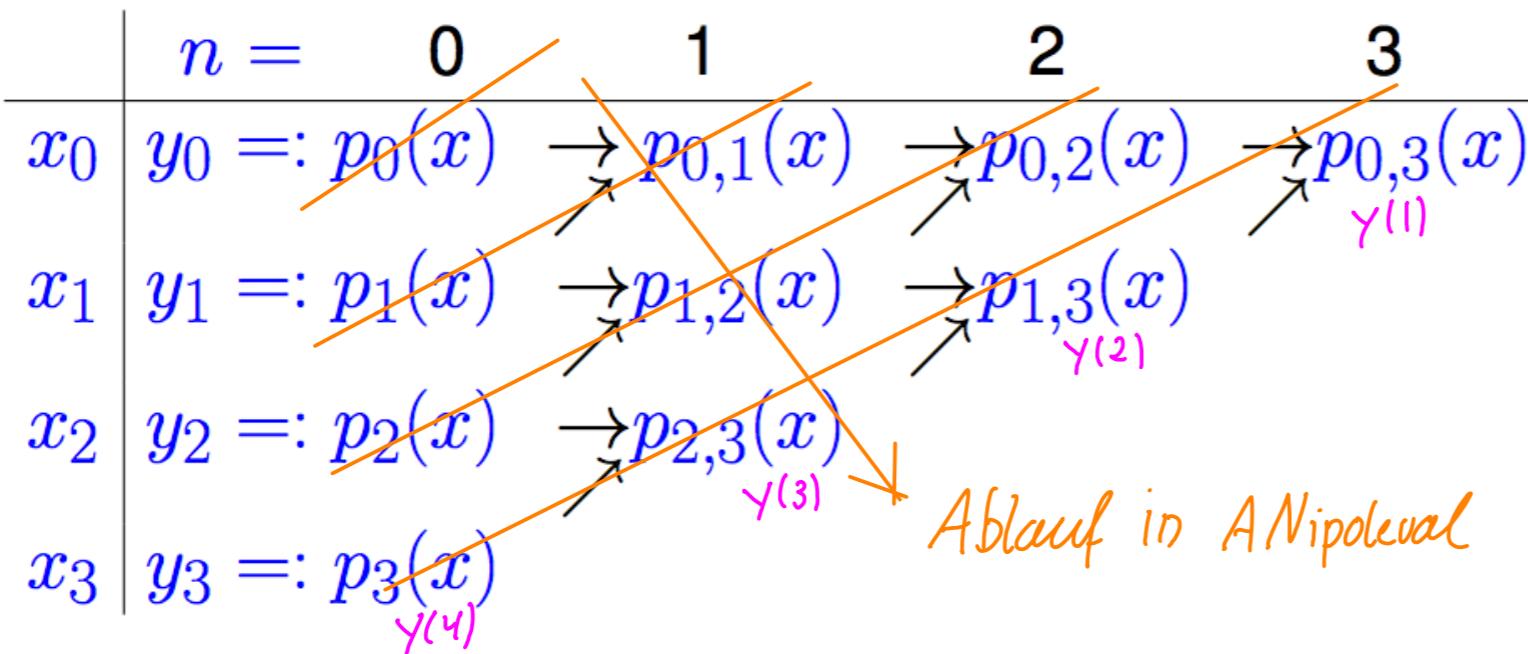
$$P_{j,k}(x_\ell) = \begin{cases} \frac{x_k - x_j}{x_k - x_\ell} \cdot (x_k - x_\ell) p_{j,k-1}(x_\ell) + y_j & , \ell = j \\ \frac{(x_\ell - x_j) p_{j+1,k}(x_\ell) + (x_k - x_\ell) p_{j,k-1}(x_\ell)}{x_k - x_j} & , j < \ell < k \end{cases}$$

\hookrightarrow erfüllt Interpolationsbedingungen

□



Aitken-Neville-Algorithmus:



→ Formel aus Satz 3.3.A

Satz 3.3.A (Aitken-Neville-Rekursionsformel)

Zu einer Knotenmenge $\mathcal{N} = \{x_0, \dots, x_n\}$ und Stützwerten $y_i, i = 1, \dots, n$, sei $p_{j,k} \in \mathbb{P}_{k-j}$, $0 \leq j \leq k \leq n$, das Interpolationspolynom durch die Punkte $(x_i, y_i), j \leq i \leq k$. Dann gilt

$$p_{j,k}(t) = \begin{cases} y_j & , \text{ falls } k = j , \\ \frac{(t - x_j)p_{j+1,k}(t) + (x_k - t)p_{j,k-1}(t)}{x_k - x_j} & , \text{ falls } 0 \leq j < k \leq n . \end{cases}$$

```

1 function v = ANipoleval(x,y,t)
2 % Aitken-Neville algorithms for evaluation of interpolating polynomial
3 % x passed interpolation nodes, y the data values, and t the evaluation
4 % point
5 for i=1:length(y)
6   for k=i-1:-1:1
7     y(k) = y(k+1)+(y(k+1)-y(k))*(t-x(i))/(x(i)-x(k)); → O(1) Aufwand
8   end
9 end
10 v = y(1);

```

< In situ Implementierung
Asymptotischer Rechenaufwand
 $O(n^2)$

Bem: "Update friendly": Einfache Berücksichtigung zusätzlicher Datenpunkte, Aufwand $O(j)$, falls schon j Punkte bekannte

Gegeben: x_0, x_1, x_2, \dots
 y_0, y_1, y_2, \dots

solange " $p_{0,j}(t)$ schlecht": $j := j+1$

3.4. Newton-Darstellung und dividierte Differenzen

```

1 class PolyEval {
2   private:
3   // evaluation point and various internal data describing the polynomials
4   public:
5     // Idle Constructor
6   PolyEval();
7   // Add another data point and update internal information
8   void addPoint(t, y);
9   // Evaluation of current interpolating polynomial at x
10  double operator () (double x) const;
11 }

```

} häufige Aufrufe in beliebiger Reihenfolge

Newton-Basis von P_h (zu Knotenmenge $\mathcal{N} = \{x_0, x_1, \dots, x_n\}$): $\{N_j\}_{j=0}^n$, $N_0(t) \equiv 1$
 $N_j(t) = \prod_{i=0}^{j-1} (t - x_i)$, $j \geq 1$

↳ Grad j , führender Koeffizient 1
 $N_j(x_l) = 0$ für $l < j$

> LGS für Polynominterpolation in **Newton-Basis**

$a_j \in \mathbb{R}: a_0 N_0(x_j) + a_1 N_1(x_j) + \dots + a_n N_n(x_j) = y_j, \quad j = 0, \dots, n.$ (*)

↔ gestaffeltes Gleichungssystem

$$\begin{pmatrix} 1 & 0 & \dots & 0 \\ 1 & (x_1 - x_0) & \dots & : \\ \vdots & \vdots & \ddots & 0 \\ 1 & (x_n - x_0) & \dots & \prod_{i=0}^{n-1} (x_n - x_i) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}.$$

▷ Lösung mit Vorwärtstechnik,
Aufwand $O(n^2)$

Bem: a_j aus (*) ist der führende Koeffizient von $P_{0,j}$

Newton-Basis-Darstellung :

$$P_{0,n}(t) = \underbrace{a_0 N_0(t) + a_1 N_1(t) + \cdots + a_j N_j(t) + \cdots + a_n N_n(t)}_{= P_{0,j}(t)}$$

Notation :

$$\gamma [x_0, \dots, x_j] := a_j$$

Def. 3.4.A (Dividierte Differenzen)

Zu gegebener (angeordneter) Knotenmenge $\mathcal{N} = (x_0, \dots, x_n)$, $n \in \mathbb{N}$, und Stützwerten y_i , $i = 1, \dots, n$, definiere die dividierte Differenz

$\gamma[x_i, x_{i+1}, \dots, x_j]$:= führender Koeffizient des Interpolationspolynoms $p_{i,j}$.

Bezeichnung klar aus:

Satz 3.4.B (Rekursionsformel für dividierte Differenzen)

$$\begin{aligned} \gamma[x_i] &= y_i, \quad i = 1, \dots, n, \\ \gamma[x_i, \dots, x_j] &= \frac{\gamma[x_{i+1}, \dots, x_j] - \gamma[x_i, \dots, x_{j-1}]}{x_j - x_i}, \quad 0 \leq i < j \leq n. \end{aligned}$$

Beweis:

Satz 3.3.A (Aitken-Neville-Rekursionsformel)

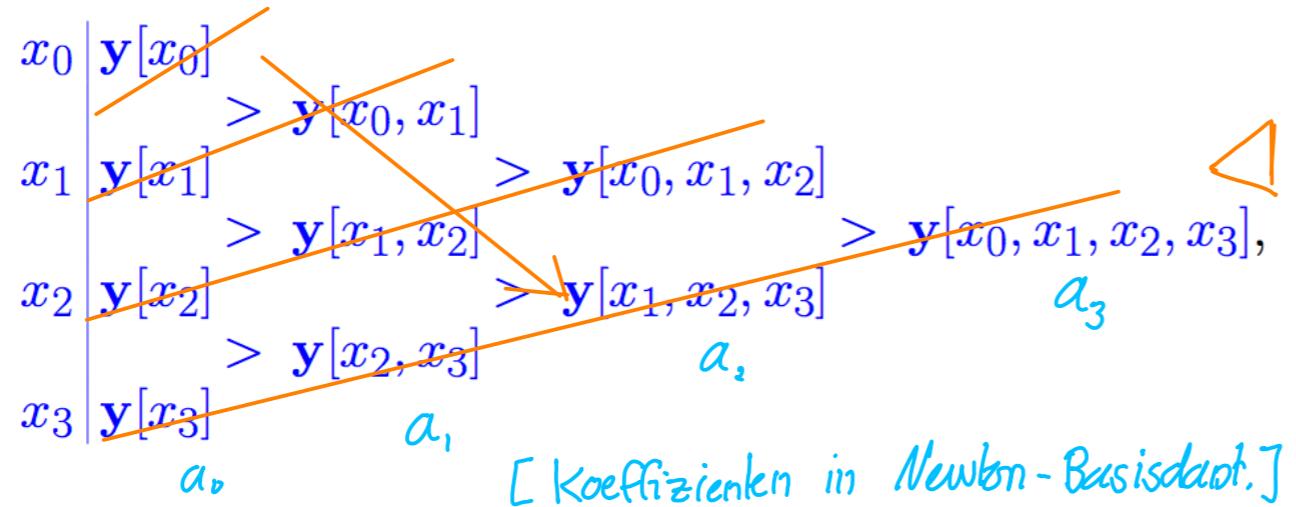
Zu einer Knotenmenge $\mathcal{N} = \{x_0, \dots, x_n\}$ und Stützwerten y_i , $i = 1, \dots, n$, sei $p_{j,k} \in \mathbb{P}_{k-j}$, $0 \leq j \leq k \leq n$, das Interpolationspolynom durch die Punkte (x_i, y_i) , $j \leq i \leq k$. Dann gilt

$$p_{j,k} = \begin{cases} y_j & , \text{falls } k = j, \\ \frac{(x_j - t)p_{j+1,k} + (x_k - t)p_{j,k-1}}{x_k - x_j} & , \text{falls } 0 \leq j < k \leq n. \end{cases}$$

▷ Rekursion für "führende Koeffizienten"

□

Rekursive Berechnung der dividierten Differenzen:



vgl. Aitken-Neville

```

1 function y = divdiff(x,y)
2 % in situ recursive computation of divided
3 % differences for nodes x(i) and data y(i)
4 n = length(y)-1;
5 if (n > 0)
6     y(1:n) = divdiff(x(1:n),y(1:n));
7     for j=n:-1:1
8         y(j) = (y(n+1)-y(j))/(x(n+1)-x(j));
9     end
10 end

```

[in situ]

Rechenaufwand ("add Point") = $O(j)$
[für j. Punkt]

Auswertung : $p(t) = \sum_{j=0}^n a_j N_j(t) = a_0 + a_1(t-x_0) + a_2(t-x_0)(t-x_1) + \dots$
 Horner schema ! $= a_0 + (t-x_0)(a_1 + (t-x_1)(\dots)) \dots$

```

1 function v = evaldivdiff(x,y,t)
2 % Evaluation of interpolating polynomial via Newton basis
3 % and divided differences
4 n = length(y); dd=divdiff(x,y); % Compute Newton coefficients
5 % Horner scheme for Newton representation
6 v=dd(1)*ones(size(t));
7 for j=2:n, v=(t-x(j)).*v + dd(j); end

```

} \Rightarrow Rechenaufwand $O(n)$

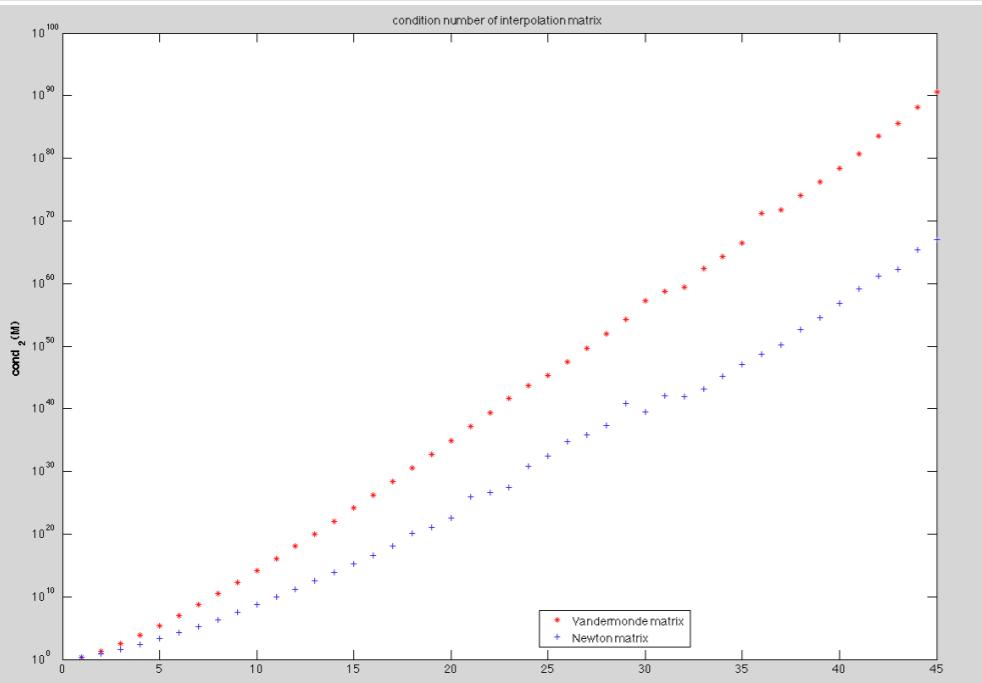
```

1  class PolyEval {
2    private:
3      std::vector<double> t; // Interpolation nodes
4      std::vector<double> y; // Coefficients in Newton representation
5    public:
6      PolyEval(); // Idle constructor
7      void addPoint(double t, double y); // Add another data point
8      // evaluate value of current interpolating polynomial at x,
9      double operator() (double x) const;
10     private:
11       // Update internal representation, called by addPoint()
12       void divdiff();
13   };
14
15 PolyEval::PolyEval() {}
16
17 void PolyEval::addPoint(double td, double yd)
18 { t.push_back(td); y.push_back(yd); divdiff(); }
19
20 void PolyEval::divdiff() {
21   int n = t.size();
22   for(int j=0; j<n-1; j++) y[n-1] = ((y[n-1]-y[j])/(t[n-1]-t[j]));
23 }
24
25 double PolyEval::operator() (double x) const {
26   double s = y.back();
27   for(int i = y.size()-2; i>= 0; --i) s = s*(x-t[i])+y[i];
28   return s;
29 }
```

\leftarrow vgl. Satz 3.4.B

\leftarrow HornerSchema

Bem:



\triangleleft "Äquidistante Knoten $x_j = j$, $j = 0, \dots, n$

$\{N_j\}$ instabil?

3.6. Funktionsapproximation durch Polynominterpolation

Nodaler Interpolationsoperator : $P_N := I_N \circ S_N : C^0([x_0, x_n]) \rightarrow P_{n+1} : P_N[f](x_j) = f(x_j), j=0, \dots, n$
 $(N = \{x_j\}_{j=0}^n$ gegeben)

\hookrightarrow lineare Abbildung

Bem : Skalierungsinvarianz : Affine Transformation $T(\hat{t}) = (1-\hat{t})a + \hat{t}b$, $T: [0, 1] \rightarrow [a, b]$, $a < b$

Pullback : $\hat{f}(\hat{t}) := f(T(\hat{t}))$, $\hat{f} \in C^0[0, 1]$ für $f \in C^0[a, b] \Rightarrow \|f\|_{\infty, [a, b]} = \|\hat{f}\|_{\infty, [0, 1]}$
 \hookrightarrow Für $\hat{N} := T^{-1}(N)$ gilt $P_{\hat{N}}[\hat{f}] = P_N[f] \circ T$ (kommutierende Diagramm)

$\triangleright \|f - P_N[f]\|_{\infty, [a, b]} = \|\hat{f} - P_{\hat{N}}[\hat{f}]\|_{\infty, [0, 1]} \rightarrow$ O.B.d.A : Referenzintervall
 ↑
 Interpolationsfehler

Satz 3.6 (Fehlerdarstellung für Polynominterpolation einer Funktion)

Seien $x_0 < x_1 < \dots < x_n$ Knoten und sei weiter $x_* \in [x_0, x_n]$ beliebig, und

$$f \in C^{n+1}([x_0, x_n]).$$

Dann gilt für das Interpolationspolynom $P_N^{(X)}[f] \in \mathbb{P}_n$ mit $P_N^{(X)}[f](x_i) = f(x_i)$ für $i = 0, \dots, n$
 folgende Fehlerdarstellung:

$$E_n[f](x_*) := f(x_*) - P_N^{(X)}[f](x_*) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega_{n+1}(x_*),$$

mit einer (von x_* stetig abhängigen) Zwischenstelle $\xi \in [x_0, x_n]$ und dem Knotenpolynom

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

← Glättheitsvoraussetzung

← ξ existiert

Beweis : $g(t) = f(t) - P_N[f](t) - c \omega_{n+1}(t) \in C^{n+1}$ hat Nullstellen $x_j \cdot \}$ g hat $n+2$ paarweise verschiedene Nullstellen in $[x_0, x_n]$
 Annahme : $x_* \notin N \Rightarrow \omega_{n+1}(x_*) \neq 0 \Rightarrow \exists c \in \mathbb{R} : g(x_*) = 0$

▷ [Satz von Rolle] $g^{(n+1)}$ hat mindestens eine Nullstelle $\xi \in [x_0, x_n]$:
 $g^{(n+1)}(\xi) = f^{(n+1)}(\xi) - 0 - c(n+1)! = 0 \Rightarrow c = \frac{f^{(n+1)}(\xi)}{(n+1)!}$ Beh. aus $g(x_*) = 0$ □

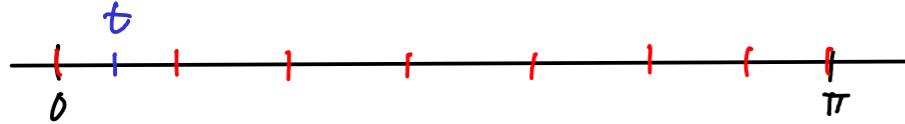
▷ [Maximum bzgl. x_*]: $\|f - P_n[f]\|_{\infty, [x_0, x_n]} \leq \frac{1}{(n+1)!} \|f^{(n+1)}\|_{\infty, [x_0, x_n]} \|c\omega_{n+1}\|_{\infty, [x_0, x_n]}$ (3.30)

Bsp: Aquidistante Knoten $N = \{\gamma_n^k\}_{k=0}^n$,
 $f = \sin$

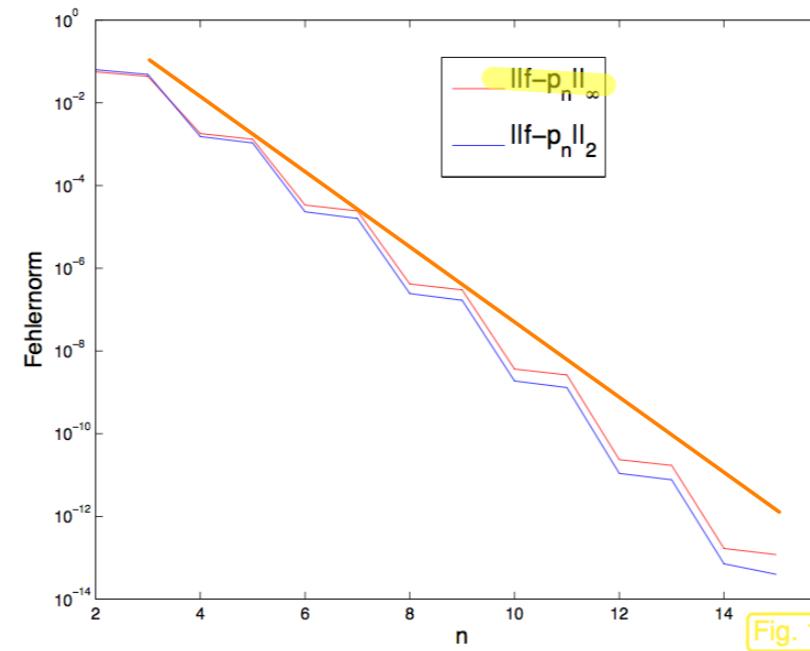
Superexponentielle Konvergenz ▷

$$\|f^{(n+1)}\|_{\infty} \leq 1$$

$$\|\omega_{n+1}\|_{\infty, [x_0, x_n]} = \max_{0 \leq t \leq \pi} \prod_{j=0}^n (t - \gamma_j^n)$$



$$\|\omega_{n+1}\|_{\infty, [0, \pi]} \leq (\pi/n)^{n+1} n!$$



MATLAB-Experiment:

Berechnung der Normen:

- L^∞ -Norm: Abtastung auf Gitter mit Maschenweite $\pi/1000$.
- L^2 -Norm: Numerische Quadratur (\rightarrow Abschnitt 4.4) mit Trapezregel auf Gitter mit Maschenweite $\pi/1000$.

$$(3.30) \Rightarrow \|\sin - P_n[\sin]\|_{\infty} \leq \frac{1}{(n+1)!} (\pi/n)^{n+1}$$

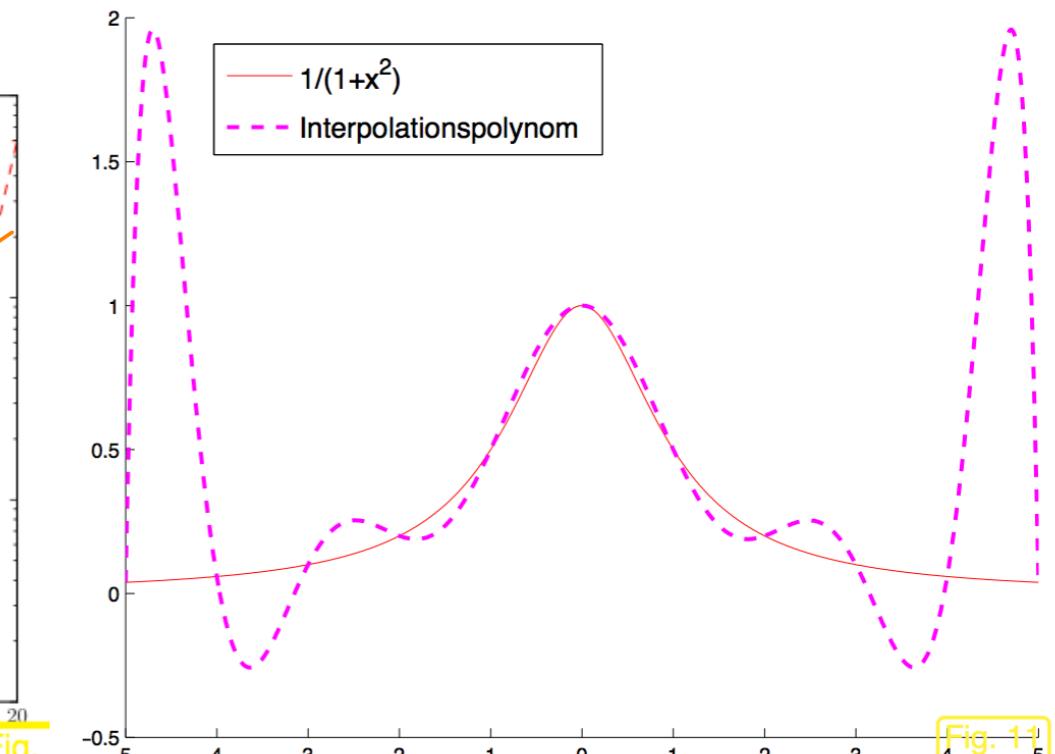
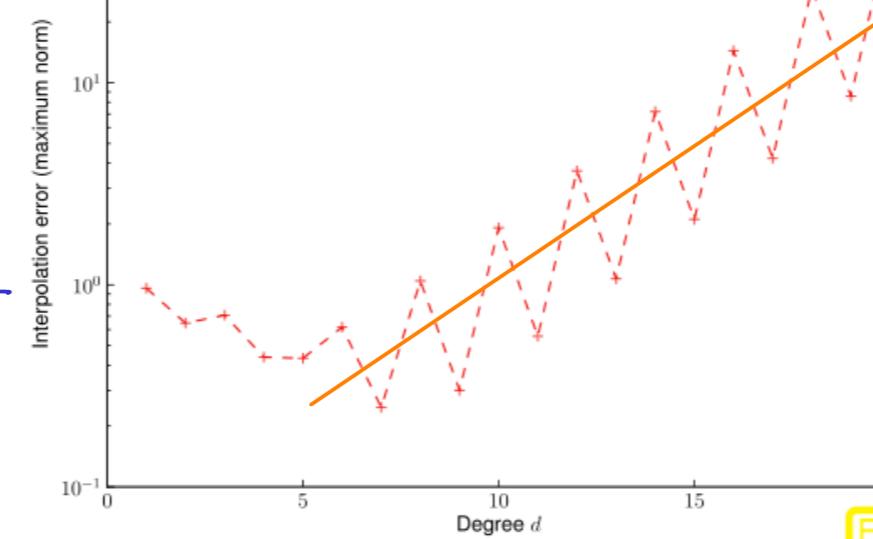
Bsp: $N = \{-5 + \frac{10}{n} j\} \subset [-5, 5]$
(Runge) $f(t) = \frac{1}{1+t^2}$

$$f(t) = \sum_{j=0}^n (-t^2)^j \Rightarrow f^{(2j)}(0) = \pm (2j)! \Rightarrow \|f^{(2j)}\|_{\infty} \geq (2j)!$$

$$\|\omega_{n+1}\|_{\infty} \geq \prod_{j=1}^n \left(\gamma_j^{\frac{10}{n}} - \frac{5}{n} \right) \left(\frac{5}{n} \right) = \left(\frac{1}{n} \right)^{n+1} 5^{\frac{n+1(2n-1)}{2n}} \underset{t = -5 + \frac{5}{n}}{\uparrow}$$

$$\geq C \frac{1}{n} (10/e)^n$$

↳ unabhängig von n ▷



Schranke in (3.30) $\geq C q^n$ mit $q > 1$

Stirling'sche Formel :

Sometimes, bounds for $n!$ rather than asymptotics are required: one has, for all $n \in \mathbb{N}_+$

$$\sqrt{2\pi} n^{n+1/2} e^{-n} \leq n! \leq e n^{n+1/2} e^{-n},$$

so for all $n \geq 1$ the ratio $\frac{n!}{n^{n+1/2} e^{-n}}$ is always e.g. between 2.5 and 3.

↓ ?

Andererseits :

Satz. (Satz von Weierstrass)

Sei $[a, b] \subset \mathbb{R}$ ein beschränktes Intervall. Für jede stetige Funktion $f \in C^0([a, b])$ gilt

$$\lim_{n \rightarrow \infty} \inf_{p \in \mathbb{P}_n} \|f - p\|_\infty = 0.$$

$$= (\text{Id} \cdot P_n)(f - q)$$

↳ Bestapproximationsfehler

$$\|f - P_n[f]\|_\infty = \|f - q - P_n[f - q]\|_\infty \quad \text{für beliebiges } q \in \mathbb{P}_n, \text{ Projektionseigenschaft } P_n[q] = q$$

$$\triangleright \|f - P_n[f]\|_\infty \leq \inf_{q \in \mathbb{P}_n} \|\text{Id} \cdot P_n\|_\infty \|f - q\|_\infty = \|\text{Id} \cdot P_n\|_\infty \inf_{q \in \mathbb{P}_n} \|f - q\|_\infty$$

↑
Interpolationsfehler

↑ Norm der linearen Abbildung $\text{Id} \cdot P_n : C^0([x_0, x_n]) \rightarrow C^0([x_0, x_n])$

Submultiplikativität der Operatormam: $\|\text{Id} \cdot P_n\|_\infty \leq \sup_{g \in C^0([x_0, x_n])} \frac{\|(Id \cdot P_n)g\|_\infty}{\|g\|_\infty}$

$$\sup_{g \in C^0([x_0, x_n])} \frac{\|(Id \cdot P_n)g\|_\infty}{\|g\|_\infty} \stackrel{?}{=} \|I_N\|_{(R^{n+1}, \|\cdot\|_\infty) \rightarrow (C^n, \|\cdot\|_\infty)}$$

Satz. (Norm der Polynominterpolation)

Gegeben seien ein beschränktes Intervall $I \subset \mathbb{R}$ und die Knotenmenge $N = (x_0, \dots, x_n)$ mit zugehörigen Lagrange-Polynomen $l_i \in \mathbb{P}_n$, $i = 0, \dots, n$.

Dann gilt für die folgenden Operatornorm des (linearen) Polynominterpolationsoperators $I_N^{(n)}$:
 $\mathbb{R}^{n+1} \rightarrow C^0(I)$

$$\|I_N^{(n)}\|_{\infty \rightarrow \infty} := \sup_{\mathbf{y} \in \mathbb{R}^{n+1} \setminus \{0\}} \frac{\|I_N^{(n)}(\mathbf{y})\|_{I, \infty}}{\|\mathbf{y}\|_\infty} = \left\| \sum_{i=0}^n l_i \right\|_{I, \infty}, \quad \text{Formel}$$

$$\left[\|I_N^{(n)}\|_{2 \rightarrow 2} := \sup_{\mathbf{y} \in \mathbb{R}^{n+1} \setminus \{0\}} \frac{\|I_N^{(n)}(\mathbf{y})\|_{L^2(I)}}{\|\mathbf{y}\|_2} \leq \left(\sum_{i=0}^n \|l_i\|_{L^2(I)}^2 \right)^{\frac{1}{2}}. \right]$$

Beweis:

$$\leq \|\text{Id} \cdot I_N\|_\infty = \max_{t \in I} \left| \sum_j y_j l_j(t) \right|$$

$$\leq \max_{t \in I} \sum_j |y_j| |l_j(t)|$$

$$\leq \|\mathbf{y}\|_\infty \max_{t \in I} \sum_j |l_j(t)|$$

$$\geq'': t^* = \operatorname{argmax}_{t \in I} \sum_j |\ell_j(t)|, \quad \gamma_j := \operatorname{sgn}(\ell_j(t^*)) \cdot \ell_j(t^*) \Rightarrow \frac{\|I_w(x)\|_\infty}{\|x\|_\infty} = \left\| \sum_j |\ell_j(\cdot)| \right\|_\infty \text{ durch Nachrechnen} \quad \square$$

Def. 3.18 (Lebesgue-Konstante)

Seien $a = x_0 < x_1 < \dots < x_n = b$ Knoten/Stützstellen und $\ell_j(x) \in \mathbb{P}_n$ die zugeordneten Lagrange-Polynome. Dann heisst

$$\Lambda_n = \Lambda_n(\{x_0, \dots, x_n\}) := \left\| \sum_{k=0}^n |\ell_k(x)| \right\|_{[a,b],\infty} = \|I_w\|_\infty$$

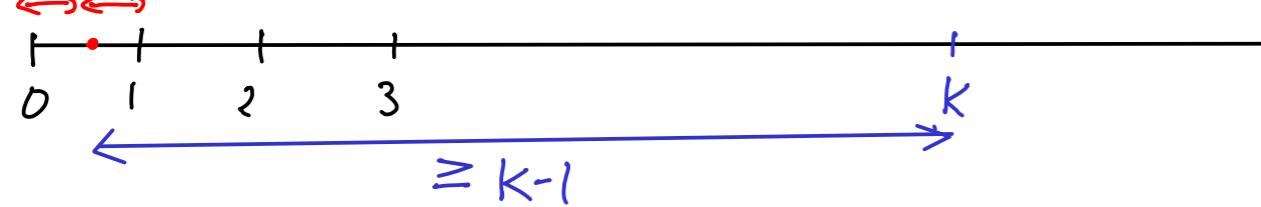
die **Lebesgue-Konstante** von $\{x_0, \dots, x_n\}$.

Lebesgue - Konstante äquidistanter Knoten (Abschätzung): O.B.d.A : $x_j = j, j=0, \dots, n$, wegen Skalierungsinvarianz
 \Leftrightarrow von unten

$$\Rightarrow \ell_j(t) = \prod_{i \neq j} \frac{t-i}{j-i} \quad (\text{Lagr. Pol.})$$

$$\Lambda_n(\{x_j\}) \geq |\ell_m(\frac{1}{2})| \quad \text{für } n=2m$$

$$|\ell_m(\frac{1}{2})| = \prod_{\substack{i=0 \\ i \neq m}}^{2m} \frac{|\frac{1}{2}-i|}{|m-i|} \geq \left(\frac{1}{2}\right)^2 \frac{(n-1)!}{(m!)^2} \geq C m^{-\frac{3}{2}} \left(\frac{3}{2}\right)^{2m}$$



exponentielles Wachstum in m !

Präzisere Abschätzung : Für äquidistante Knoten $\Lambda_n = O\left(\frac{2^{n+1}}{n \log n}\right)$ für $n \rightarrow \infty$

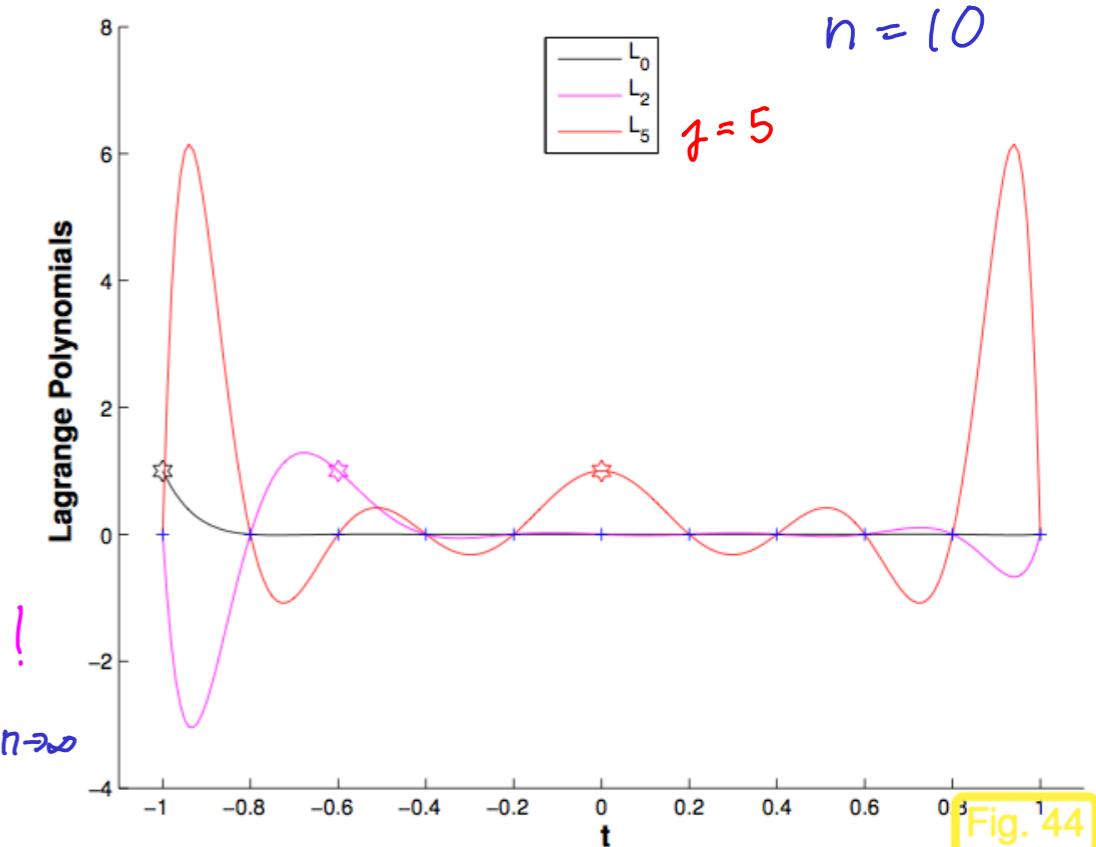


Fig. 44

3.7. Bestapproximation durch Polynome

→ Satz von Weierstrass:

$$\lim_{n \rightarrow \infty} \underbrace{\inf_{p_n \in P_n} \|f - p_n\|_{\infty, I}}_{\text{Wie schnell } \rightarrow 0 \text{ (Rate!)}} = 0 \quad \text{für jedes } f \in C^0(I), I \subset \mathbb{R} \text{ kompakt}$$

Satz 3.21. (Existenz bestapproximierender Polynome) $n \in \mathbb{N}$ fixiert.

Für alle $f \in C^0([a, b])$, $-\infty < a < b < \infty$, existiert mindestens ein $p_n \in \mathbb{P}_n$ derart, dass

$$\bar{\zeta} := \inf_{q \in \mathbb{P}_n} \|f - q\|_{\infty} = \|f - p_n\|_{\infty}.$$

Beweis: Kompaktheitargument

$$\begin{aligned} \text{Polynomfolge } (q_j)_{j \in \mathbb{N}} &\subset \mathbb{P}_n: \lim_{n \rightarrow \infty} \|f - q_n\|_{\infty} = \bar{\zeta} \\ \|q_j\|_{\infty} &\leq \|f\|_{\infty} + 2\bar{\zeta} =: C \quad \text{für hinreichend großes } j \\ q_j &\in \{p \in \mathbb{P}_n : \|p\|_{\infty} \leq C\} \end{aligned}$$

$$\Rightarrow \exists \text{ Teilfolge: } (q_{j_k})_k : q_{j_k} \rightarrow p_n : \|p_n - f\|_{\infty} = \bar{\zeta}$$

Kompakt \Leftarrow beschränkt, abgeschlossen, \subset endl. dim. Raum \square

Sensor für Glattheit (empfindlicher, allgemeiner als $\|f^{(n)}\|_{\infty}$):

Def. 3.22 (Stetigkeitsmodul)

Sei $-\infty < a < b < \infty$ und sei $f \in C^0([a, b])$. Der Stetigkeitsmodul von $f \in C^0([a, b])$ zur Schrittweite $h > 0$ ist definiert durch

$$\omega(f; h) = \max_{\substack{x, x' \in [a, b] \\ |x-x'| \leq h}} \{|f(x) - f(x')|\}.$$

Bsp: $f(t) = t^{\alpha}$, $0 < \alpha < 1$ (Wurzelfunktion)
 $\Rightarrow f \in C^1([0, 1])$

$$\omega(f, h) = \max_{0 \leq t \leq 1-h} (t+h)^{\alpha} - t^{\alpha} = h^{\alpha}$$

↑
Kurvendiskussion

Lemma (Eigenschaften des Stetigkeitsmoduls)

(SM1) Für $f \in C^0([a, b])$ ist die Funktion $h \mapsto \omega(f; h)$ von $h \in \mathbb{R}_+$ nichtnegativ, monoton wachsend und **subadditiv**, d.h.

$$\forall h_1, h_2 \geq 0 : \omega(f; h_1 + h_2) \leq \omega(f; h_1) + \omega(f; h_2).$$

(SM2) Für alle $n \in \mathbb{N}$ und für alle $\lambda > 0$ gilt

$$(i): \omega(f; nh) \leq n\omega(f; h) \quad \text{und} \quad \omega(f; \lambda h) \leq (1 + \lambda)\omega(f; h). \quad (\ddot{\alpha})$$

(SM3) für $f \in C^0([a, b])$ ist $h \mapsto \omega(f; h)$ eine stetige Funktion von $h > 0$, für die der (rechtsseitige) Grenzwert bei $h = 0$ existiert: $\lim_{h \rightarrow 0} \omega(f; h) = 0$,

(SM4) Für $f \in C^1([a, b])$ ist

$$\omega(f; h) \leq h \|f'\|_\infty.$$

$$\omega(f, h) = \max_{\substack{a \leq t, t' \leq b \\ |t-t'| \leq h}} |f(t) - f(t')|$$

→ A-Ungl.

→ (i) aus (SM1), (ii) aus (i) wg. Monotonie mit $n = \lceil \lambda \rceil$

→ glm. Stetigkeit

→ $|f(t) - f(t')| = f'(\bar{x})(t - t')$: Mittelwertsatz

Satz 3.23. (Satz von Jackson)

Sei $-\infty < a < b < \infty$ und sei $f \in C^0([a, b])$. Dann existiert eine Konstante $C_J > 0$ (die unabhängig von f, a, b und n ist) derart, dass für alle $n \in \mathbb{N}$ gilt

$$\min_{v \in P_n} \|f - v\|_\infty \leq C_J \omega(f, \frac{b-a}{n}).$$

▷ (SM3) : Satz von Weierstrass

▷ (SM4) : $f \in C^1([a, b])$

$$\min_{p \in P_n} \|f - p_n\|_\infty = O(\frac{1}{n}) \quad \text{für } n \rightarrow \infty$$

Beweis: (Fouriertechniken) (i) $g(t) = f(\overbrace{|cost|}^{=: x})$, π -periodisch, gerade

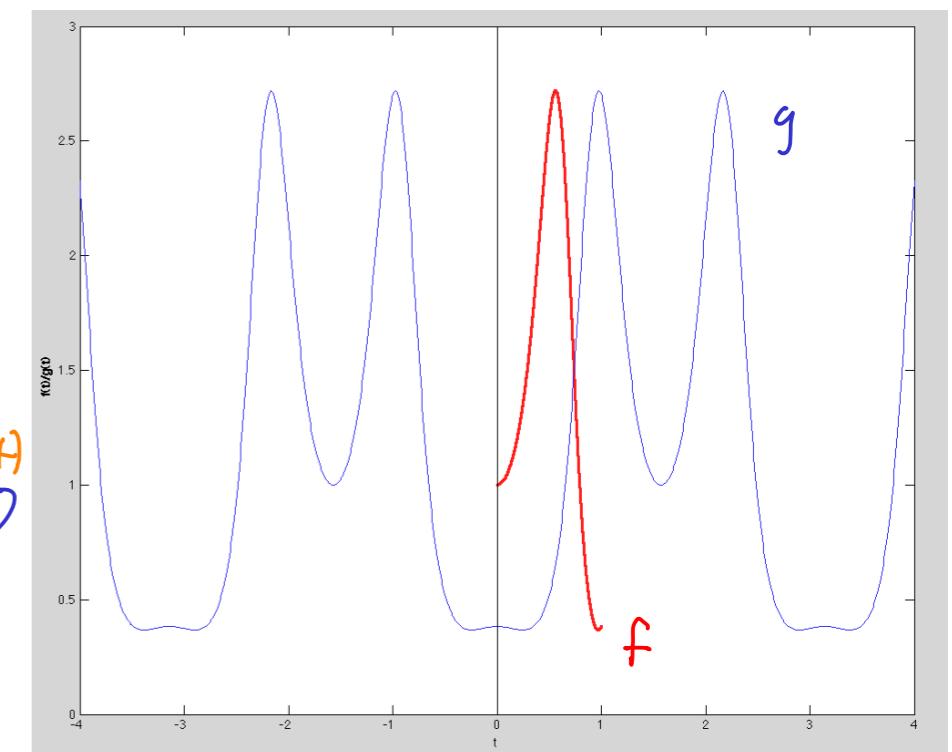
O.B.d.A: $a=0, b=1$

$$g(t) \approx \sum_{j=0}^N c_j \cos(2jt) = \sum_{j=0}^N c_j (2\cos(jt)^2 - 1)$$

Es gilt: $\cos(jt) = T_j(\cos t)$ mit $T_j \in P_j$
 $f(x) = g(t) \approx q_{2N}(\cos^2 t) = q_{2N}(x^2)$ mit $q_{2N} \in P_{2N}$, $x \in [0, 1]$

(ii) Erinnerung: Fejer-Kern, Hier verwenden wir: $J_n(s) = \frac{1}{2\pi} \left(\frac{\sin((k+1)s)}{\sin(ks)} \right)^4 \geq 0$
 $[$ Für gerades $n=2p$ $]$ $\alpha_n: \int_{-\pi}^{\pi} J_n(t) dt = 1 \quad (*)$

$$\text{Es gilt: } \exists \alpha_{n,k} \in \mathbb{R} \quad J_n(s) = \sum_{k=0}^n \alpha_{n,k} \cos(ks)$$

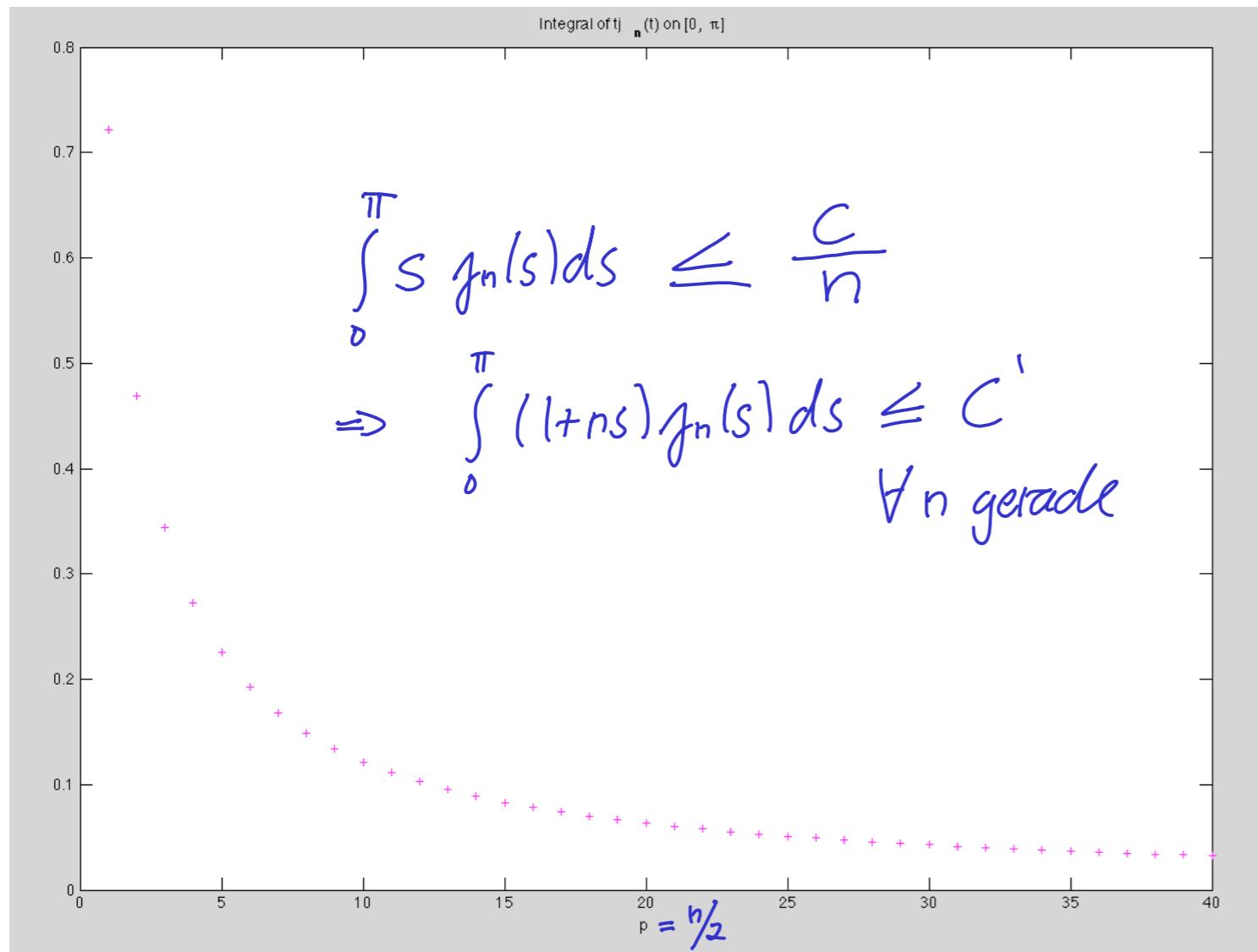


```

In[8]:= (var[#] = FourierCosSeries[(Sin[#t/2]/Sin[t/2])^4, t, 3*#]) & /@ {3, 4, 5, 6, 7, 8}
Out[8]= {19 + 32 Cos[t] + 20 Cos[2t] + 8 Cos[3t] + 2 Cos[4t], 44 + 80 Cos[t] + 62 Cos[2t] + 40 Cos[3t] + 20 Cos[4t] + 8 Cos[5t] + 2 Cos[6t],
85 + 160 Cos[t] + 136 Cos[2t] + 104 Cos[3t] + 70 Cos[4t] + 40 Cos[5t] + 20 Cos[6t] + 8 Cos[7t] + 2 Cos[8t],
146 + 280 Cos[t] + 250 Cos[2t] + 208 Cos[3t] + 160 Cos[4t] + 112 Cos[5t] + 70 Cos[6t] + 40 Cos[7t] + 20 Cos[8t] + 8 Cos[9t] + 2 Cos[10t],
231 + 448 Cos[t] + 412 Cos[2t] + 360 Cos[3t] + 298 Cos[4t] + 232 Cos[5t] + 168 Cos[6t] + 112 Cos[7t] + 70 Cos[8t] + 40 Cos[9t] + 20 Cos[10t] + 8 Cos[11t] + 2 Cos[12t],
344 + 672 Cos[t] + 630 Cos[2t] + 568 Cos[3t] + 492 Cos[4t] + 408 Cos[5t] + 322 Cos[6t] + 240 Cos[7t] + 168 Cos[8t] + 112 Cos[9t] + 70 Cos[10t] + 40 Cos[11t] + 20 Cos[12t] + 8 Cos[13t] + 2 Cos[14t]}

```

$$\begin{aligned}
g(t) &\approx \underbrace{\int_{-\pi}^{\pi} f_n(s) g(t-s) ds}_{=: \varphi_n(t)} = \sum_{k=0}^n a_{n,k} \int_{-\pi}^{\pi} \cos(ks) g(t-s) ds = \sum_{k=0}^n a_{n,k} \int_{-\pi}^{\pi} g(s) \cos(k(t-s)) ds \\
&= \sum_{k=0}^n a_{n,k} \int_{-\pi}^{\pi} g(s) [\cos(kt) \cos(ks) + \sin(kt) \sin(ks)] ds = \sum_{j=0}^n c_j \cos(2jt) \\
&\quad \stackrel{(+) \text{ für } k \text{ ungerade}}{=} 0 \quad \stackrel{=0}{\text{für } k \text{ gerade}} \quad \Leftrightarrow \lambda \Leftrightarrow h \\
|g(t) - \varphi_n(t)| &\stackrel{(*)}{=} \left| \int_{-\pi}^{\pi} (g(t) - g(t-s)) f_n(s) ds \right| \stackrel{(+)}{\leq} \int_{-\pi}^{\pi} |g(t) - g(t-s)| f_n(s) ds \leq 2 \int_0^{\pi} \omega(g, (n|s|)^{\frac{1}{n}}) f_n(s) ds \\
(\text{SM2}) \quad &\leq 2 \int_0^{\pi} (1+ns) \omega(g, \frac{1}{n}) f_n(s) ds
\end{aligned}$$



$$\begin{aligned}
&\triangleright |g(t) - \varphi_n(t)| \leq C'' \omega(g, \frac{1}{n}) \\
&\Leftrightarrow |f(x) - \overset{\uparrow}{\varphi_n}(x^2)| \leq C''' \omega(f, \frac{1}{2n}) ; g(t) = f(\lceil \cos t \rceil) \\
&[|t-t'| \leq h \Rightarrow |\lceil \cos t \rceil - \lceil \cos t' \rceil| \leq h] \\
&\omega(g, h) = \max_{|t-t'| \leq h} |g(t) - g(t')| \leq \max_{0 \leq x, x' \leq 1} |f(x) - f(x')| \\
&\triangleright |f(x) - \varphi_n(x^2)| \leq C'''' \omega(f, \frac{1}{2n}) \\
&\text{(iii) } n \text{ ungerade} \\
&\inf_{p \in P_n} \|f - p_n\|_\infty \stackrel{(ii)}{\leq} \inf_{p \in P_{n-1}} \|f - p\|_\infty \leq C''' \omega(f, \frac{1}{n-1}) \\
&\leq 2C''' \omega(f, \frac{1}{n}) \quad \square
\end{aligned}$$

Korollar 3.24. (Polynomiale Bestapproximation glatter Funktionen)

Sei $r \in \mathbb{N}$ und $f \in C^r([-1, 1])$. Dann gilt mit $C > 0$ unabhängig von f und n

$$\min_{v \in \mathbb{P}_n} \|f - v\|_\infty \leq C n^{-r} \|f^{(r)}\|_\infty \quad \forall n \geq r.$$

Beweis : (Induktion)

$r = 1$, siehe oben, 3.23 & (SM4)

$$\forall q \in \mathbb{P}_n : \inf_{p \in \mathbb{P}_n} \|f - q - p\|_\infty \stackrel{\text{IH. } t=1}{\leq} C_1 \frac{1}{n} \|f' - q'\|_\infty \Rightarrow \inf_{p \in \mathbb{P}_n} \|f - p\|_\infty \leq C_1 \frac{1}{n} \inf_{q \in \mathbb{P}_{n-1}} \|f' - q\|_\infty \stackrel{\text{IH}}{\leq} C_1 \frac{1}{n(n-1)^{r-1}} \|f^{(r)}\|_\infty$$

$$[n(n-1)^{r-1} \geq \frac{1}{r} e^r n^r, ? = \frac{t(r-1)^{r-1}}{r^r}, q(x) = \frac{x(x-1)^{r-1}}{x^r} = (1-\frac{1}{x})^{r-1} \geq (1-\frac{1}{r})^{r-1} \geq \frac{1}{r} e^{-r}] \leq C_1 \frac{1}{n^r} \|f^{(r)}\|_\infty \square$$

3.8 Tschebyscheff-Interpolation

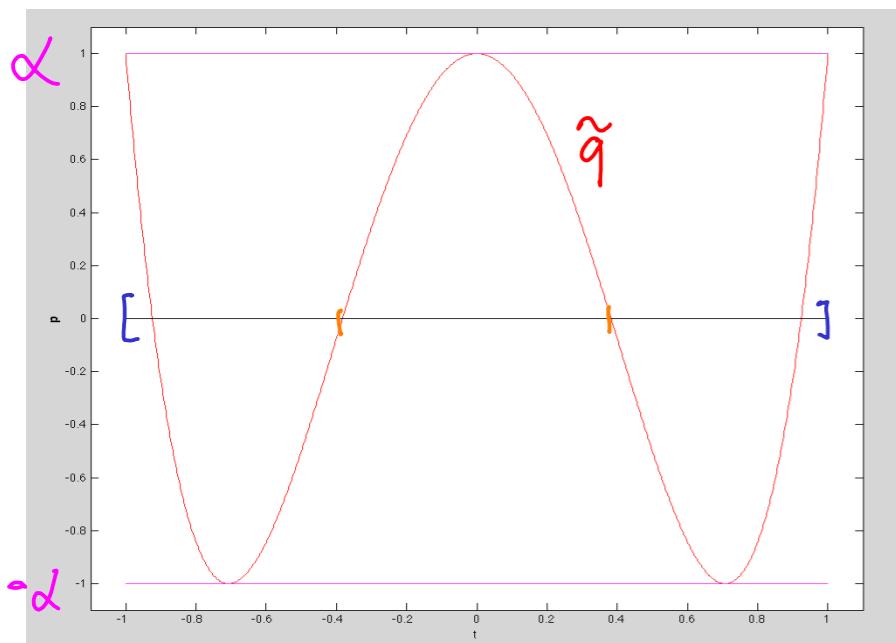
Bei Funktionsinterpolation : \mathcal{N} frei \Rightarrow Finde "optimales" \mathcal{N} ohne Wissen von f (a priori Wahl)

$$(3.30) \quad \|f - P_N f\|_\infty \leq \frac{1}{(n+1)!} \|f^{(n+1)}\|_\infty \|\omega_{n+1}\|_\infty, \text{ wenn } \#\mathcal{N} = n+1$$

\square Einzige Option: minimiere diese Norm \Leftrightarrow

Finde ein Polynom vom Grad $n+1$ mit führendem Koeffizienten 1 und $\|\cdot\|_\infty$ minimal

Heuristik : $\alpha = \min \{ \|q\|_\infty, q \in \mathbb{P}_{n+1}, \text{ führender Koeffizient} \}$, O.B.d.A : Intervall $[-1, 1]$

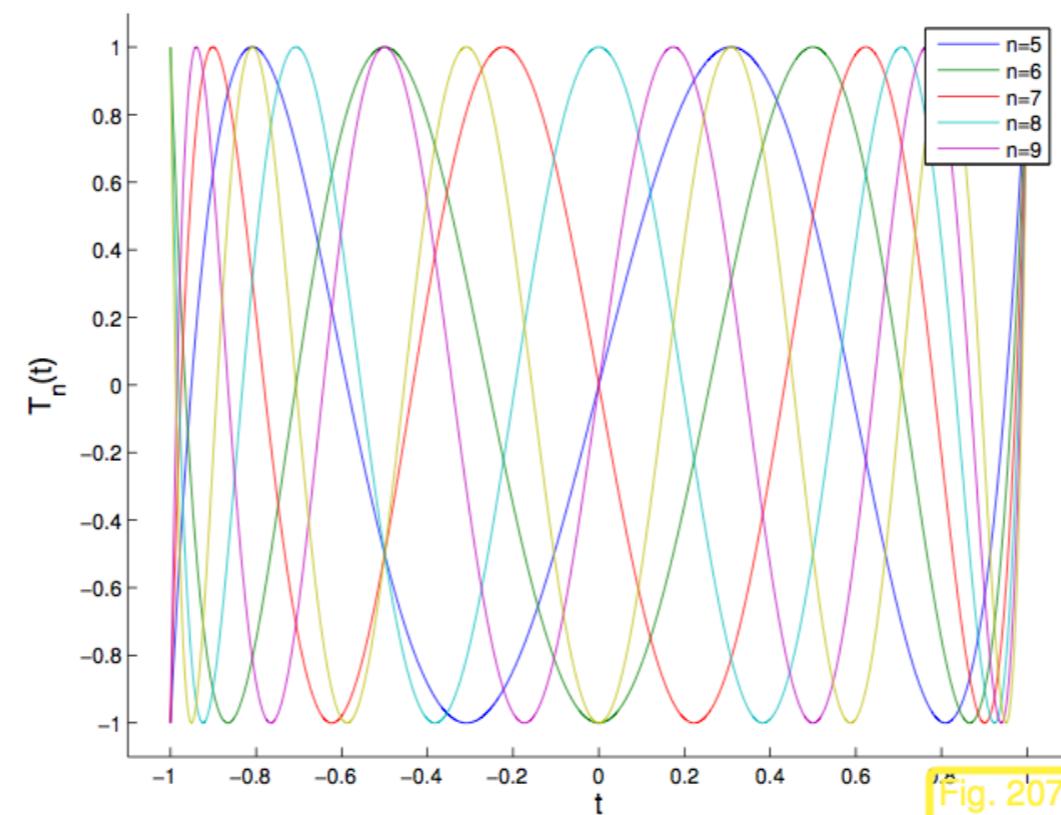
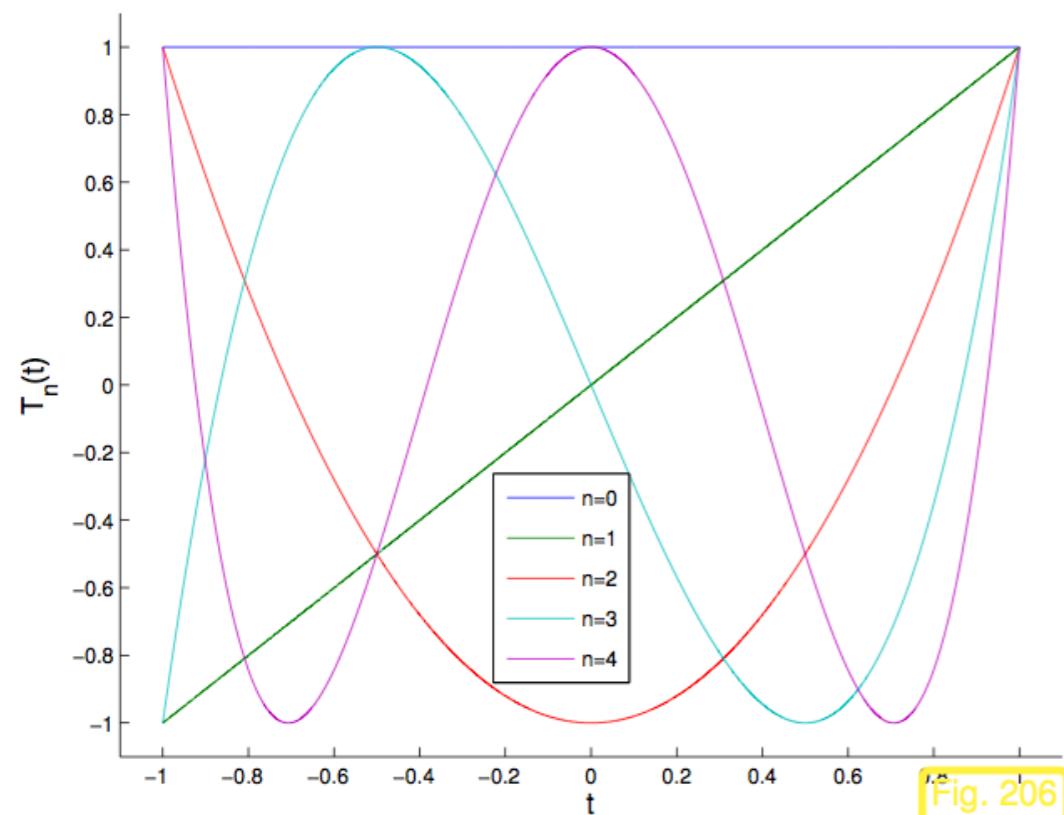


- $n+1$ Nullstellen in $[-1, 1]$
- $n+2$ Extrema x_i^* in $[-1, 1]$, mit alternierendem Vorzeichen, $|\tilde{q}(x_i^*)| = \alpha$, $|\tilde{q}(1)| = |\tilde{q}(-1)| = \alpha$

Def. 3.27 (Tschebyscheffpolynome)

Für $k \in \mathbb{N}_0$ ist das k -te Tschebyscheff-Polynom $T_k(x)$ für $-1 \leq x \leq 1$ definiert durch

$$T_k(t) = \cos(k \arccos t).$$



Lemma 3.28 (Eigenschaften der Tschbyscheff-Polynome) $T_k(t) = \cos(k \arccos(t))$, $-1 \leq t \leq 1$

1. Die Tschebyscheff-Polynome haben ganzzahlige Koeffizienten.
2. Der höchste Koeffizient von T_n ist $a_n = 2^{n-1}$.
3. T_n ist eine gerade Funktion, falls n gerade, und eine ungerade Funktion, falls n ungerade.
4. Es gilt $T_0(t) = 1$ und $T_1(t) = t$ und die Rekursionsformel

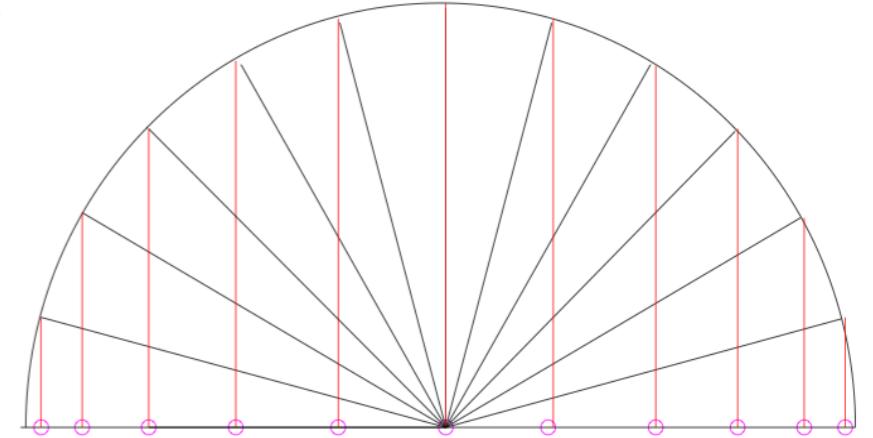
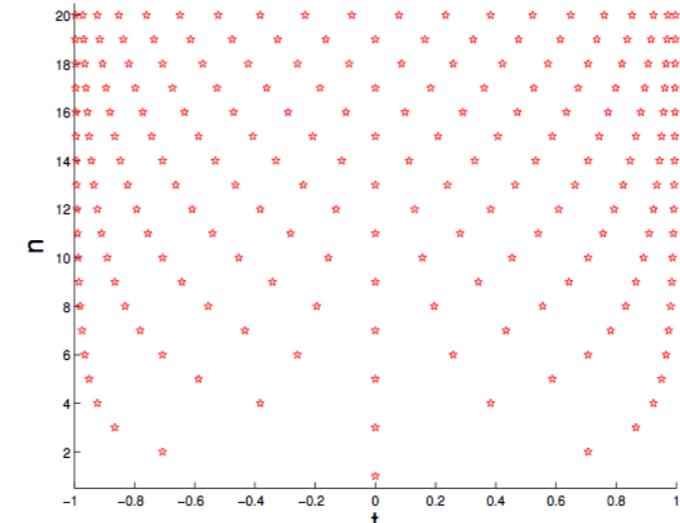
$$T_k(t) = 2tT_{k-1}(t) - T_{k-2}(t), \quad k \geq 2. \quad \rightarrow T_k \in P_K$$

5. $|T_n(t)| \leq 1$ für $t \in [-1, 1]$.
6. Für $n \geq 1$ nimmt $|T_n(x)|$ den (extremen) Wert 1 in $[-1, 1]$ genau an den $n+1$ Clenshaw-Curtis Punkten

$$\bar{x}_j^n := \cos(j\pi/n) \quad \text{für } j = 0, \dots, n \text{ an.}$$

7. Die n Nullstellen von T_n sind explizit gegeben durch die Tschebyscheffknoten

$$\hat{x}_j^{(n)} = \cos\left(\frac{(2j-1)\pi}{2n}\right), \quad j = 1, \dots, n. \quad \rightarrow \text{optimales } N$$



Satz 3.26 (min-max Eigenschaft der Tschbyscheff-Polynome)

$$\text{Für alle } k \in \mathbb{N}_0: \quad \|2^{1-k}T_k\|_{\infty, [-1, 1]} = \inf_{q \in P_k^1} \|q\|_{\infty, [-1, 1]}.$$

↑ Polynome mit fühlendem Koeffizienten 1

T_k hat $k+1$ Extrema $\Rightarrow p$ hat k N.S. $\Rightarrow p \equiv 0 \Leftrightarrow \hat{x}_0^k = -1 \quad \hat{x}_n^k = 1$ □

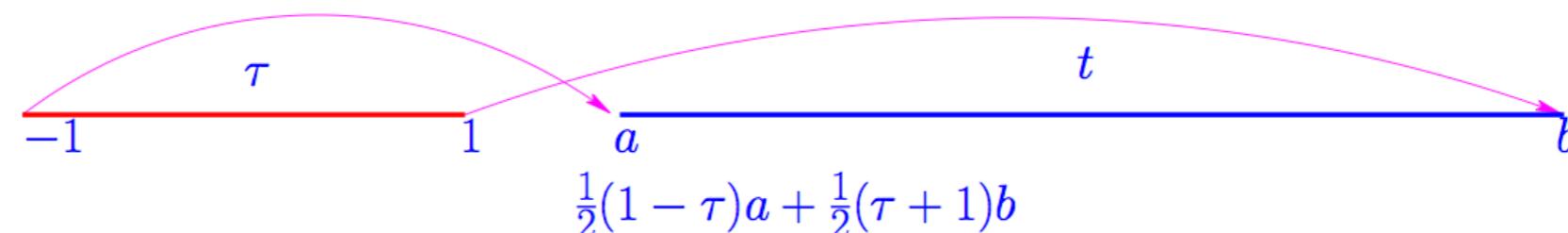
Beweis: (indirekt) $\hat{x}_j^k \stackrel{!}{=} \text{Extrema von } T_k \text{ in } [-1, 1]$

Annahme: $\exists q \in P_{k-1}^1, \|q\|_{\infty, [-1, 1]} < \|2^{1-k}T_k\|_{\infty, [-1, 1]}$
Behachte: $p = 2^{1-k}T_k - q \in P_{k-1}^1$

Vorzeichen von p
in Extrema von T_k

Bem: Tschebyscheff- Knoten auf $[a, b]$:

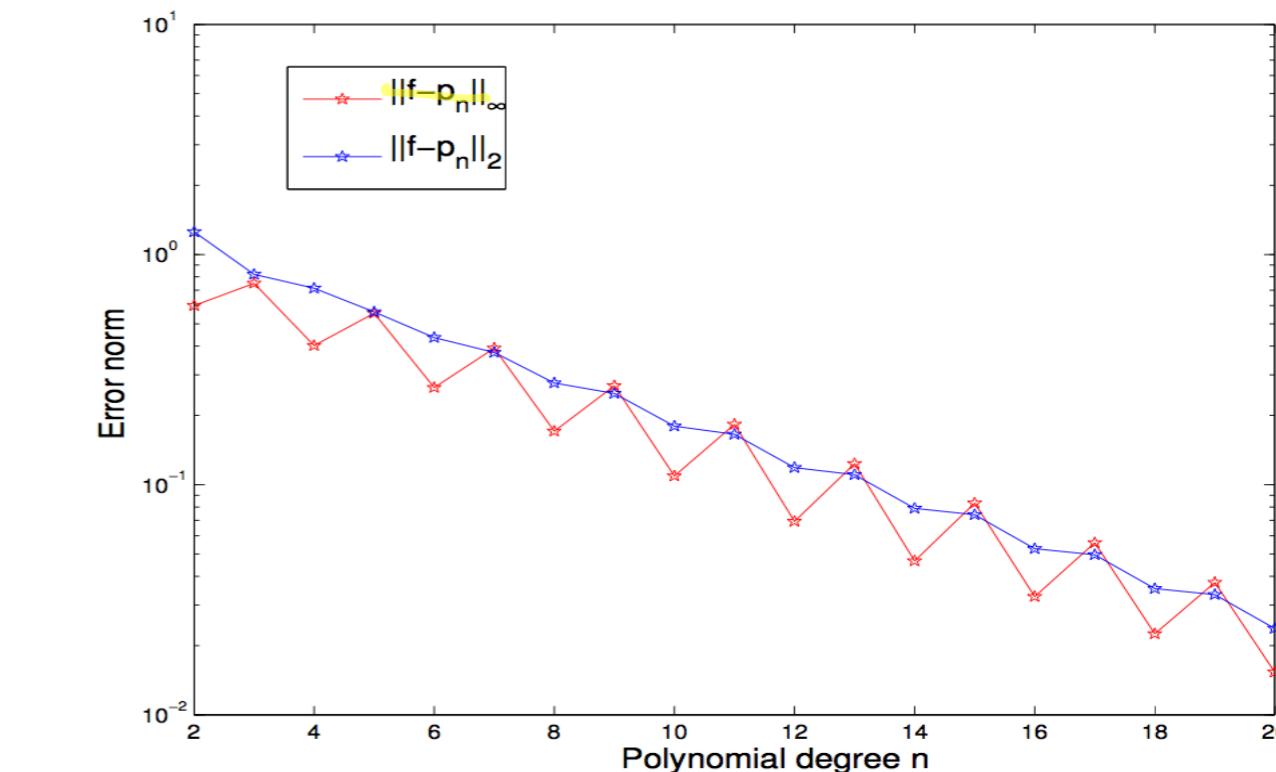
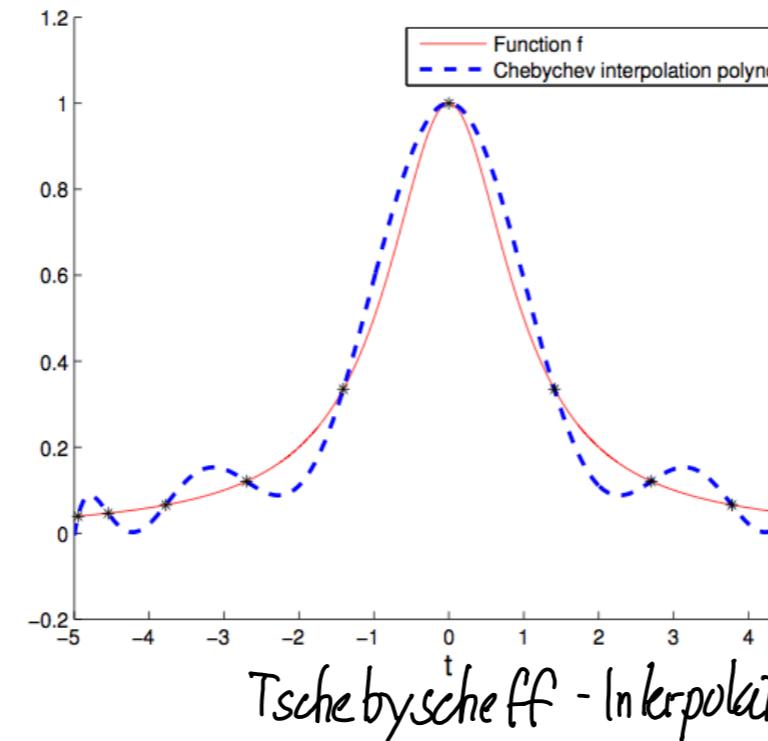
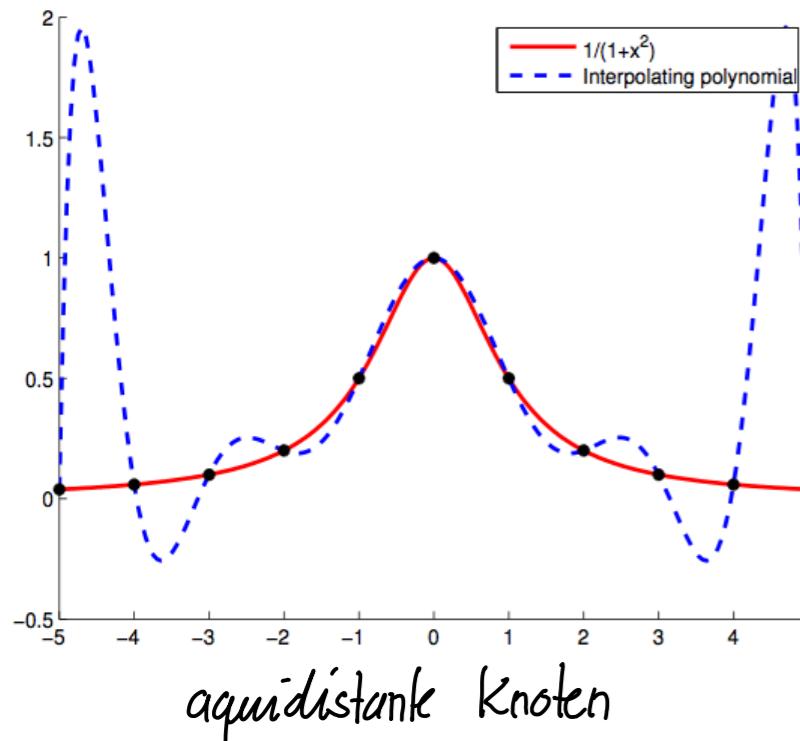
$$x_j^{(n)} := a + \frac{1}{2} (\tilde{x}_j^{(n)} + 1) (b - a)$$



Interpolationsfehlerabschätzung für Tschebyscheff-Interpolation (auf $[-1, 1]$)

$$(3.30) \quad \|f - P_{N_{\text{ches}}} f\| \leq \frac{1}{(n+1)!} \|f^{(n+1)}\|_\infty \|2^{-n} T_{n+1}\|_\infty \leq \frac{1}{(n+1)!} \|f^{(n+1)}\|_\infty 2^{-n}$$

Bsp : $f(t) = \frac{1}{1+t^2}$ auf $[-5, 5]$,

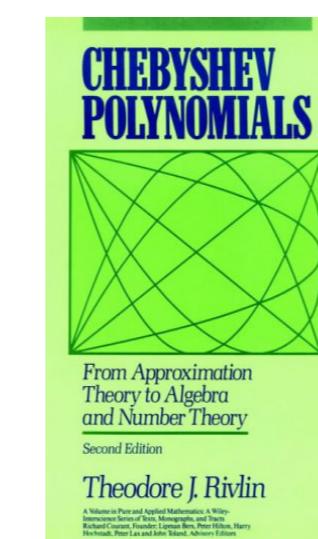


Exponentielle Konvergenz des Interpolationsfehlers
 $\|f - p_n\|_\infty = O(e^{-\alpha n})$ für ein $\alpha > 0$

auf $[-1, 1]$

$[\log \|f - p_n\|_\infty] \approx -\alpha n$

	n	Λ_n für äquidist. Stützst.	Λ_n für Tschebyscheff Stützst.
Λ_n = Lebesgue-Konstante	5	3.106	2.104
Def. 3.33	10	29.89	2.489
	15	512.05	2.728
	20	10986.53	2.901



Lemma 3.31 (Lebesgue-Konstante für Tschebyscheff-Knoten)

Die Lebesgue-Konstante der Tschebyscheff-Knoten erfüllt die Schranke

$$\Lambda_n \leq \frac{2}{\pi} \log(n-1) + 4.$$

3.9. Splineinterpolation

Vorwort: *Nichtlokalität* der (globalen) Polynominterpolation von Datenpunkten (\leftrightarrow Oszillationen)

Abhilfe: Stückweise Polynome (z.B. Polygonzug) \rightarrow Stückweise Polynominterpolation

(i) Partitionierung: $[a, b] = \bigcup_{k=1}^n [x_{k-1}, x_k]$, $a = x_0 < x_1 < \dots < x_n = b$

(ii) [Einfache Version]: Fixiere $m \in \mathbb{N}$, $\forall J_k$ wähle $x_{k-1} = \xi_0^{(k)} < \xi_1^{(k)} < \dots < \xi_{m-1}^{(k)} < \xi_m^{(k)} = x_k$
 \Rightarrow Stückweiser Polynominterpolant $q: [a, b] \rightarrow \mathbb{R}$: $q|_{J_k} \in P_m$, $q(\xi_j^{(k)}) = f(\xi_j^{(k)})$ für gegebenes $f \in C^0([a, b])$
 Es gilt: $q \in C^0([a, b])$

Satz 3.9.A (Fehlerabschätzung fürstückweise Polynominterpolation)

Sei $f \in C^{m+1}([a, b])$, und p_m das **stückweise Interpolationspolynom** vom Grad $\leq m$ zu
 äquidistanten \star Interpolationsknoten in den Teilintervallen $J_k := [x_{k-1}, x_k]$, $k = 1, \dots, n$.

Dann gilt die Fehlerabschätzung

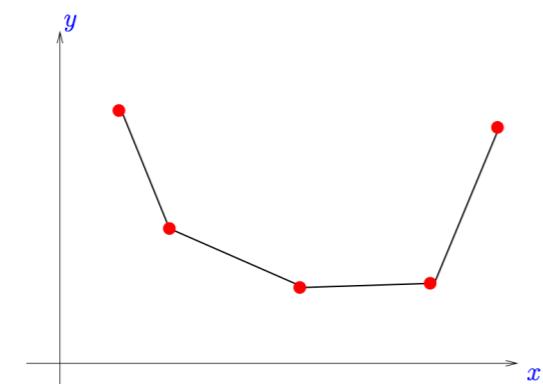
$$\|f - p_m\|_{\infty, [a, b]} \leq C(m) h^{m+1} \|f^{(m+1)}\|_{\infty, [a, b]}, \quad h := \max_{k=1, \dots, n} |x_k - x_{k-1}|.$$

mit $C(m) := \frac{1}{(m+1)!} \left(\frac{e}{m}\right)^2 \sqrt{m-1}$.

\hookrightarrow Maschenweite

Algebraische Konvergenz $O(h^{m+1})$

$m = 1$: Polygonzug



Vorteil: Lokalität, oszillationsfrei für kleines m

Nachteil: "knicke", i.a. $q \notin C^1([a, b])$

Kompromiss: Splineinterpolation

* d.h. $\xi_j^{(k)} = x_{k-1} + \frac{j}{m} (x_k - x_{k-1})$

\rightarrow (3.30) + Abschätzung $\|w_m\| = h^{m+1}(n-1)!$
 + Stirlingsche Formel

* \rightarrow Glattheitbanforderung

Def. 3.35(1) (Kubischer Spline)

$s : [a, b] \rightarrow \mathbb{R}$ heisst **kubischer Spline** bzgl. der Knotenmenge $\mathcal{N} := \{a = x_0 < x_1 < x_2 < \dots < x_{n-1} < x_n = b\}$, wenn

(i) s auf $[a, b]$ zweimal stetig differenzierbar ist und

(ii) s eingeschränkt auf $[x_{k-1}, x_k]$, $k = 1, \dots, n$, ein Polynom vom Grad ≤ 3 ist.

→ Glättigkeit $s \in C^2([a, b])$

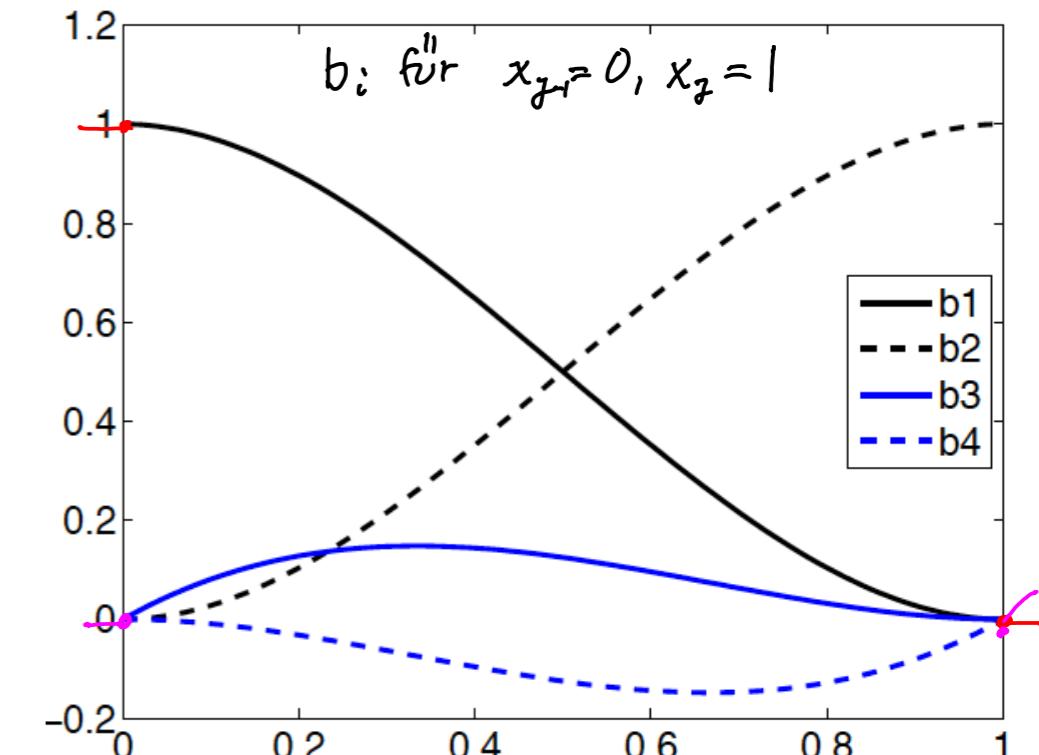
→ Bzgl. $\cup [x_{k-1}, x_k]$ stckw. kubisches Polynom

Basis für \mathbb{P}_3 ($x_j \neq x_{j-1}$) : $\phi(t) = 3t^2 - 2t^3$, $\psi(t) = t^3 - t^2$, $h_j := x_j - x_{j-1}$,

kubische Polynom $\leftarrow \begin{cases} b_1(x) = \phi\left(\frac{x_j - x}{h_j}\right), & b_2(x) = \phi\left(\frac{x - x_{j-1}}{h_j}\right) \\ b_3(x) = -h_j \psi\left(\frac{x_j - x}{h_j}\right), & b_4(x) = h_j \psi\left(\frac{x - x_{j-1}}{h_j}\right) \end{cases}$

(*)

$$\begin{aligned} b_1(x_{j-1}) &= 1, & b_1(x_j) &= 0, & b'_1(x_{j-1}) &= 0, & b'_1(x_j) &= 0, \\ b_2(x_{j-1}) &= 0, & b_2(x_j) &= 1, & b'_2(x_{j-1}) &= 0, & b'_2(x_j) &= 0, \\ b_3(x_{j-1}) &= 0, & b_3(x_j) &= 0, & b'_3(x_{j-1}) &= 1, & b'_3(x_j) &= 0, \\ b_4(x_{j-1}) &= 0, & b_4(x_j) &= 0, & b'_4(x_{j-1}) &= 0, & b'_4(x_j) &= 1. \end{aligned}$$



Ansatz für kubischen Spline auf $[x_{j-1}, x_j]$: $s(t) = y_{j-1}b_1(t) + y_jb_2(t) + c_{j-1}b_3(t) + c_jb_4(t)$, $x_{j-1} \leq t \leq x_j$
mit Koeffizienten $y_{j-1}, y_j, c_{j-1}, c_j \in \mathbb{R}$

$$(*) \Rightarrow s(x_{j-1}) = y_{j-1} = f(x_{j-1}), \quad s(x_j) = y_j = f(x_j), \quad s'(x_{j-1}) = c_{j-1}, \quad s'(x_j) = c_j$$

Wenn Steigungen c_j eindeutig $\Rightarrow s \in C^1([a, b])$: $c_j \triangleq$ Freiheitsgrade

Lemma 3.36

Sei s ein stückweise kubisches Polynom bzgl. der Knotenmenge $\mathcal{N} := \{a = x_0 < x_1 < x_2 < \dots < x_{n-1} < x_n = b\}$. Genügt es für gegebene Stützwerte y_j , $j = 0, \dots, n$, dem Ansatz

$$s(t) = y_{j-1}b_1(t) + y_jb_2(t) + c_{j-1}b_3(t) + c_jb_4(t) \quad \text{mit } c_{j-1}, c_j \in \mathbb{R}$$

auf jedem Teilintervall $[x_{k-1}, x_k]$, dann ist s eine kubische Splinefunktion genau dann, wenn

$$\frac{1}{h_j}c_{j-1} + \left(\frac{2}{h_j} + \frac{2}{h_{j+1}}\right)c_j + \frac{1}{h_{j+1}}c_{j+1} = 3\left(\frac{y_j - y_{j-1}}{h_j^2} + \frac{y_{j+1} - y_j}{h_{j+1}^2}\right), \quad j = 1, \dots, n-1$$

Beweis : $\left[\begin{array}{l} s''|_{[x_{j-1}, x_j]}(x_j) = s''|_{[x_j, x_{j+1}]}(x_j), \quad j = 1, \dots, n-1 \\ \text{Lineare Gleichung für } c_{j-1}, c_j, c_{j+1}, \underbrace{y_{j-1}, y_j, y_{j+1}}_{\text{bekannt aus I.B.}} \end{array} \right]$

▷ LGS für unbekannte Steigungen $s'(x_j) = c_j$, $j = 0, \dots, n$

$$\begin{array}{c}
 \text{n-1 Gleichungen} \\
 \uparrow \\
 \left(\begin{array}{ccccccc}
 b_0 & a_1 & b_1 & 0 & \cdots & \cdots & 0 \\
 0 & b_1 & a_2 & b_2 & & & \\
 0 & 0 & \ddots & \ddots & & & \\
 \vdots & & \ddots & \ddots & & & \\
 & & & \ddots & a_{n-2} & b_{n-2} & 0 \\
 & & & \cdots & 0 & b_{n-2} & a_{n-1} \\
 0 & \cdots & \cdots & 0 & b_{n-2} & a_{n-1} & b_{n-1}
 \end{array} \right) \left(\begin{array}{c} c_0 \\ \vdots \\ c_n \end{array} \right) = \left(\begin{array}{c} 3 \left(\frac{y_1 - y_0}{h_1^2} + \frac{y_2 - y_1}{h_2^2} \right) \\ \vdots \\ 3 \left(\frac{y_{n-1} - y_{n-2}}{h_{n-1}^2} + \frac{y_n - y_{n-1}}{h_n^2} \right) \end{array} \right)
 \end{array}$$

n+1 Unbekannte

Rang n-1, da "strikte diagonaldominant" \triangleright Existenz eines 2-dim. Lösungsraums

$$\begin{aligned}
 b_i &:= \frac{1}{h_{i+1}}, \quad i = 0, 1, \dots, n-1, \\
 a_i &:= \frac{2}{h_i} + \frac{2}{h_{i+1}}, \quad i = 0, 1, \dots, n-1. \\
 [\quad b_i, a_i > 0 \quad , \quad a_i = 2(b_i + b_{i-1}) \quad]
 \end{aligned}$$

Satz 3.9.B (Dimension des kubischen Splineraumes)

Die kubischen Splinefunktionen zur Knotenmenge $\mathcal{N} := \{a = x_0 < x_1 < x_2 < \dots < x_{n-1} < x_n = b\}$ bilden einen Untervektorraum von $C^2([a, b])$ der Dimension $n+3$.

\downarrow
2 + (n+1) Interpolationsbedingungen

Def. 3.35(2) (Kubische Splineinterpolanten)

Ein kubischer Spline zur Knotenmenge $\mathcal{N} := \{a = x_0 < x_1 < x_2 < \dots < x_{n-1} < x_n = b\}$ erfülle die Interpolationsbedingungen $s(x_k) = y_k$, $k = 0, \dots, n$, zu gegebenen Stützwerten y_k .

- s heisst **vollständiger** interpolierender Spline, falls gilt: $s'(a) = c_0$ und $s'(b) = c_n$ für vorgegebene Endpunktsteigungen c_0 und c_n .
- s heisst **periodischer** interpolierender Spline mit Periode $b - a$, falls gilt $s^{(j)}(a) = s^{(j)}(b)$ für $j = [0, 1, 2]$.
- s heisst **natürlicher** interpolierender Spline, wenn gilt: $s''(a) = s''(b) = 0$.

$\rightarrow c_0 = c_n$, Aus $s''(a) = s''(b) \Rightarrow$ eine Gleichung

\rightarrow Zwei zusätzliche Gleichungen

LGS zum vollständigen Spline :

spd LGS mit Tridiagonalmatrix \Rightarrow

$$\left(\begin{array}{ccccccccc}
 b_0 & a_1 & b_1 & 0 & \cdots & \cdots & 0 & & \\
 0 & b_1 & a_2 & b_2 & & & & & \\
 0 & 0 & \ddots & \ddots & \ddots & & & & \\
 \vdots & & \ddots & \ddots & \ddots & & & & \\
 0 & & \ddots & a_{n-2} & b_{n-2} & 0 & & & \\
 & & \cdots & 0 & b_{n-2} & a_{n-1} & b_{n-1} & &
 \end{array} \right) \left(\begin{array}{c} c_0 \\ \vdots \\ c_n \end{array} \right) = \left(\begin{array}{c} 3 \left(\frac{y_1 - y_0}{h_1^2} + \frac{y_2 - y_1}{h_2^2} \right) - c_0 b_0 \\ \vdots \\ 3 \left(\frac{y_{n-1} - y_{n-2}}{h_{n-1}^2} + \frac{y_n - y_{n-1}}{h_n^2} \right) - c_n b_{n-1} \end{array} \right)$$

• LGS zum periodischen Spline-IP:

↳ strikt diagonal dominant

$$\begin{pmatrix} a_1 & b_1 & 0 & \dots & 0 & b_0 \\ b_1 & a_2 & b_2 & & & 0 \\ 0 & \dots & \dots & \dots & & \vdots \\ \vdots & & \dots & \dots & \dots & 0 \\ 0 & & \dots & a_{n-1} & b_{n-1} & b_0 \\ b_0 & 0 & \dots & 0 & b_{n-1} & a_0 \end{pmatrix} \begin{pmatrix} c_1 \\ c_{n-1} \\ c_n \end{pmatrix} = \dots$$

Extremaleigenschaft:

Satz 3.9C (Extremaleigenschaft des ~~natürlichen~~ kubischen Splines)

Gegeben seien die Knotenmenge $\mathcal{N} := \{a = x_0 < x_1 < x_2 < \dots < x_{n-1} < x_n = b\}$ und Stützwerten $y_j, j = 0, \dots, n$.

Bezeichne s_N den natürlichen kubische interpolierende Spline und s_V den vollständigen interpolierenden Spline, für den zusätzlich $s'_V(x_0) = c_0$ und $s'_V(x_n) = c_n$ für gegebene c_0, c_n gefordert wird. Dann gilt

$$\int_a^b |s''_N(t)|^2 dt = \inf \left\{ \int_a^b |f''(t)|^2 dt : f \in C^2([a, b]), f(x_k) = y_k, k = 0, \dots, n \right\},$$

$$\int_a^b |s''_V(t)|^2 dt = \inf \left\{ \int_a^b |f''(t)|^2 dt : f \in C^2([a, b]), f(x_k) = y_k, k = 0, \dots, n, f'(x_0) = c_0, f'(x_n) = c_n \right\}.$$

Beweis: "Satz von Pythagoras"

Lemma. Sei $s : [a, b] \rightarrow \mathbb{R}$ ein **kubischer Spline** zur Knotenmenge $\mathcal{N} := \{a = x_0 < x_1 < x_2 < \dots < x_{n-1} < x_n = b\}$ und $f \in C^2([a, b])$ mit $f(x_k) = 0, k = 0, \dots, n$.

Dann gilt

$$\int_a^b f''(t)s''(t) dt = 0,$$

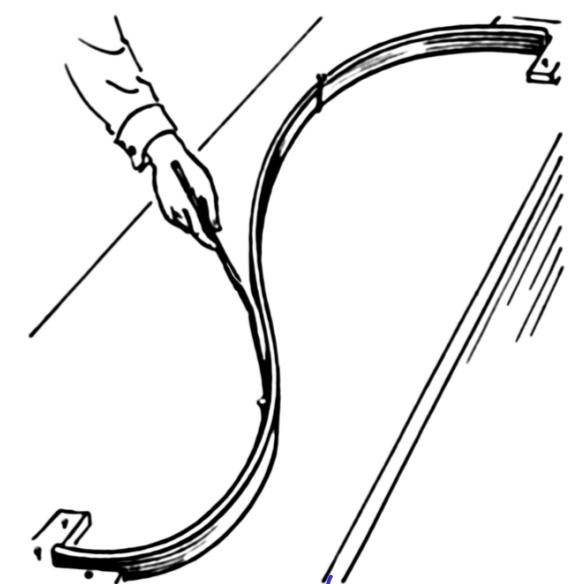
wenn

$$(i) \quad f'(a) = f'(b) = 0 \quad \text{oder} \quad (ii) \quad s''(a) = s''(b) = 0$$

Beweis: [partielle Integration (lokal)]

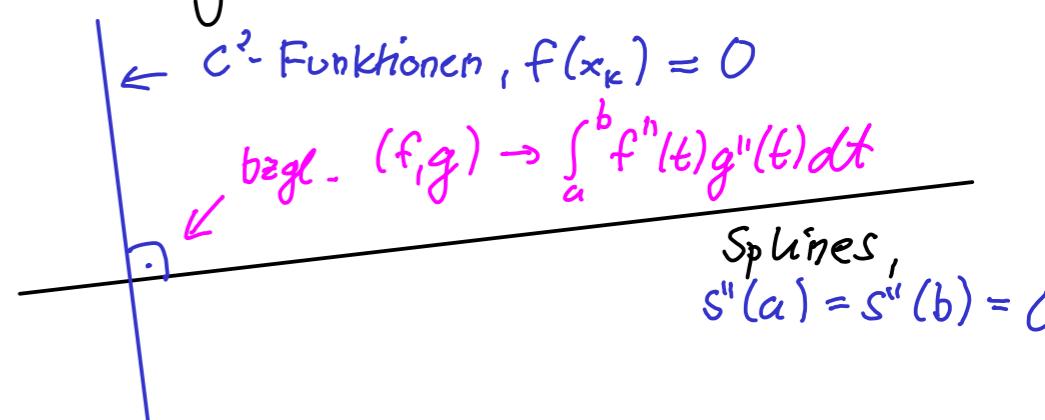
L^2 -Norm* einer Funktion $\|g\|_{L^2([a, b])} := \sqrt{\int_a^b g(t)^2 dt}$

Biegeenergie eines elastischen Stabs dessen Form durch den Graphen von f beschrieben wird $\|f''\|_{L^2([a, b])}^2$!



* aus Skalarprodukt $(f, g)_{L^2} := \int_a^b f(t) g(t) dt$

→ Orthogonalität



$$\int_a^b f''(t) s''(t) dt = \sum_{j=1}^n \int_{x_{j-1}}^{x_j} f''(t) s''(t) dt \stackrel{\text{P.I. } 2x}{=} \sum_{j=1}^n [f' s'']_{x_{j-1}}^{x_j} - [f s'']_{x_{j-1}}^{x_j} + \int_{x_{j-1}}^{x_j} f(t) s'''(t) dt$$

$$= \sum_{j=1}^n f'(x_j) s''(x_j) - f'(x_{j-1}) s''(x_{j-1}) - \underbrace{f(x_j) s'''(x_j^-)}_{=0} + \underbrace{f(x_{j-1}) s'''(x_{j-1}^+)}_{=0} = f'(b) s''(b) - f'(a) s''(a) = 0 \quad \square$$

Beweis 3.9.C: $s \hat{=} \text{interpolierender Spline zu } f \in C^2([a,b])$: $\|f''\|_{L^2}^2 = \|(\underbrace{f-s}_\text{orthogonal} + s'')\|_{L^2}^2 \stackrel{\substack{\text{Pythagoras} \\ \text{orthogonal gemäß Lemma}}}{{\leq}} \|(\underbrace{f-s}_\text{orthogonal})''\|_{L^2}^2 + \|s''\|_{L^2}^2 \geq 0$ \square

Approximation durch (vollständigen) kubischen Splineinterpolanten

$\hookrightarrow s'(x_k) = f(x_k), s'(a) = f'(a), s'(b) = f'(b)$ für $f \in C^1([a,b]), s \hat{=} \text{kubischer Spline}$

Bsp.: $f(t) = \frac{1}{1+t^2}, x_j = -1 + \frac{2}{n} j, j=0, \dots, n$ (äquidistant mit Maschenweite $h := \frac{2}{n}$)

Behaute: $\|f-s_n\|_{L^2([a,b])} := \left(\int_a^b (f-s_n)(t)^2 dt \right)^{1/2}$
(Mittelwert quadratischer Fehler)

- 1 Maximum von $f-s_n$ auf "sehr feinem Gitter"
- 2 Mit Trapezregel auf sehr feinem Gitter

$\alpha > 0$: "Ordnung der Kitz."



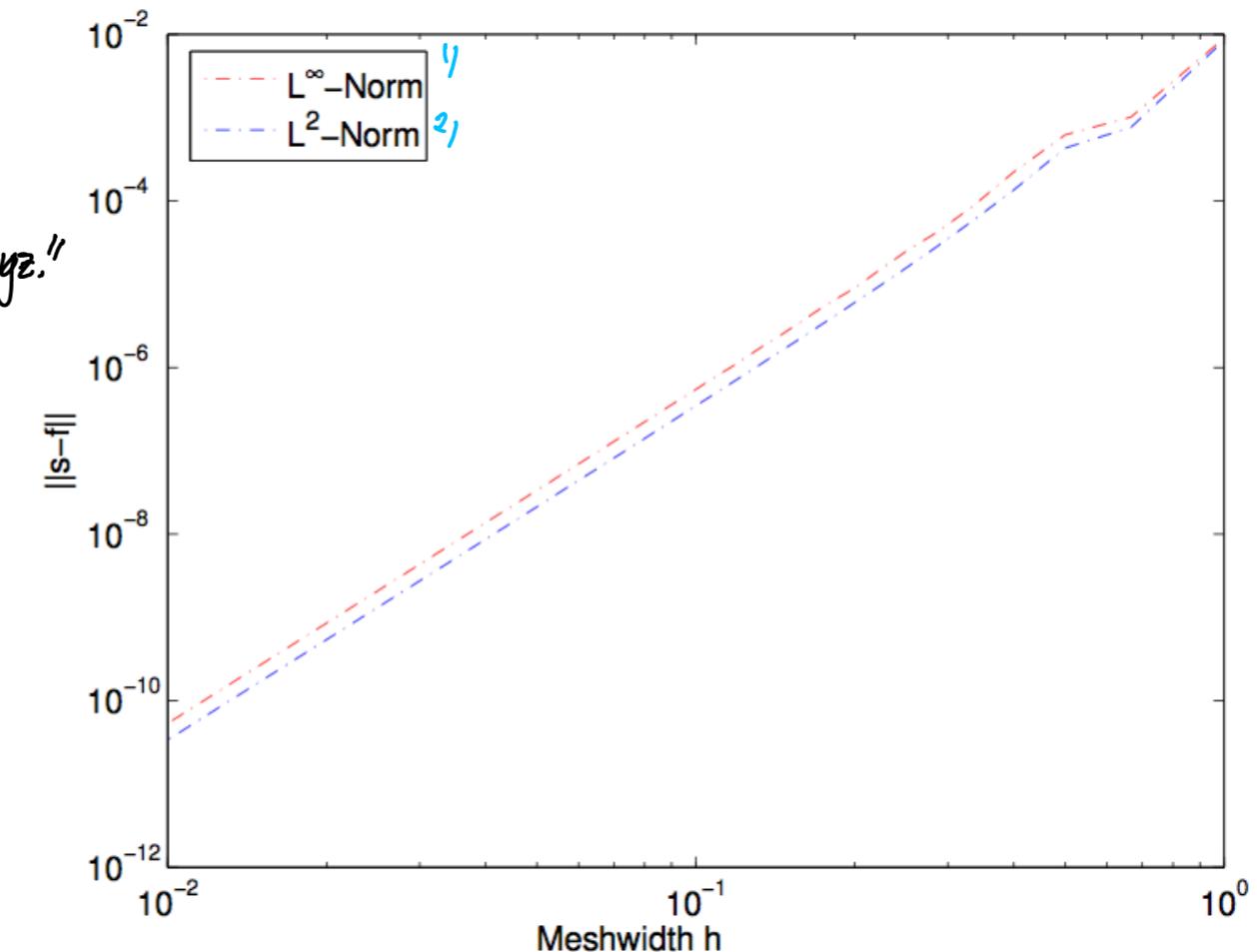
Algebraische Konvergenz: $\|\text{Fehler}\| = O(h^\alpha)$



$\alpha > 0$ aus der Steigung der Geraden im doppellogarithmischen Plot

$$\|\cdot\| \approx Ch^\alpha \Leftrightarrow \log \|\cdot\| = \alpha \log h + \log C$$

"Abmessen": $\Rightarrow \alpha = 4$



Satz 3.9E (Interpolationsfehlerabschätzung für vollständigen kubischen Splineinterpolanten)

Gegeben seien die Knotenmenge $\mathcal{N} := \{a = x_0 < x_1 < x_2 < \dots < x_{n-1} < x_n = b\}$ und $f \in C^4([a, b])$. Für den vollständigen kubischen Splineinterpolanten von f auf \mathcal{N} , d.h. $f(x_j) = s(x_j)$, $j = 0, \dots, n$, $s'(a) = f'(a)$ und $s'(b) = f'(b)$, gilt

$$\|f - s\|_{L^2([a, b])} \leq h^4 \|f^{(4)}\|_{L^2([a, b])}^*$$

mit der Maschenweite $h := \max_{j=1, \dots, n} |x_j - x_{j-1}|$.

* \rightarrow Glattheitseinforderung

Beweis: (via Orthogonalität)

$q \stackrel{!}{=} \text{kubischer Spline, beliebig}$

Orthogonalität (Lemma 3.9D): $\int_a^b (f-s)'' q'' dt = 0$
 $\in C^2$, $=0$ an Knoten, $'=0$ in a, b

orthogonal bzgl.

$$\|(f-s)''\|_{L^2}^2 + \underbrace{\|(s-q)''\|_{L^2}^2}_{\text{spline}} = \|(f-q)''\|_{L^2}^2 \quad (\text{Pythagoras}) \Rightarrow \|(f-s)''\|_{L^2} \leq \|(f-q)''\|_{L^2} \quad (+)$$

$$P: p|_{[x_{k-1}, x_k]} \in P_1, \quad p(x_k) = f''(x_k) \quad [\text{Polygonzuginterpolant von } f''] \Rightarrow \|f'' - p\|_{L^2([x_{k-1}, x_k])}^2 \leq (x_k - x_{k-1})^4 \|f^{(4)}\|_{L^2([x_{k-1}, x_k])}^2$$

Anwendung auf $f'' - p|_{[x_{k-1}, x_k]}$

$$\begin{aligned} \|f'' - p\|_{L^2([a, b])}^2 &\leq h^4 \|f^{(4)}\|_{L^2([a, b])}^2 \\ \|\!(f-s)''\|_{L^2([a, b])}^2 &\leq h^4 \|f^{(4)}\|_{L^2([a, b])}^2 \end{aligned}$$

Lemma 3.9D α

Sei $-\infty < a < b < \infty$ und $g \in C^2([a, b])$ mit $g(a) = g(b) = 0$. Dann gilt

$$\|g\|_{L^2([a, b])} \leq (b-a)^2 \|g''\|_{L^2([a, b])}.$$

Beweis \rightarrow HA

$$\hookrightarrow \text{mit } g = f - s \text{ auf } [x_{k-1}, x_k] \Rightarrow \|f - s\|_{L^2([x_{k-1}, x_k])}^2 \leq (x_k - x_{k-1})^4 \|(f-s)''\|_{L^2([x_{k-1}, x_k])}^2$$

$$\stackrel{\Sigma_k}{\Rightarrow} \|f - s\|_{L^2([a, b])}^2 \leq h^4 \|(f-s)''\|_{L^2([a, b])}^2 \leq h^8 \|f^{(4)}\|_{L^2([a, b])}^2$$

□