ETH Lecture 401-0663-00L Numerical Methods for CSE

Numerical Methods for Computational Science and Engineering

Prof. R. Hiptmair, SAM, ETH Zurich

(with contributions from Prof. P. Arbenz and Dr. V. Gradinaru)

Autumn Term 2016

(C) Seminar für Angewandte Mathematik, ETH Zürich

URL: http://www.sam.math.ethz.ch/~hiptmair/tmp/NumCSE/NumCSE16.pdf

VI. Approximation of Functions in 1D

Case study: Model reduction in circuit simulation

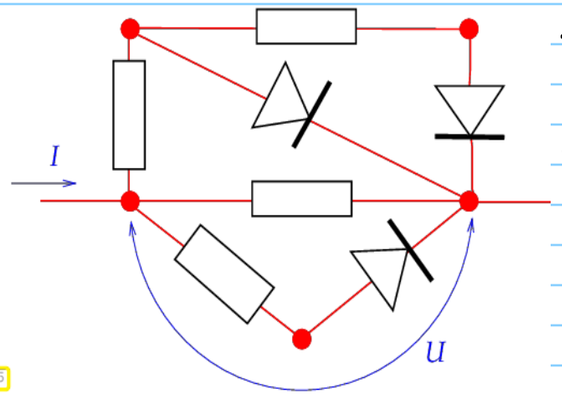


Fig. 205

◁ 2-port (non-linear) circuit element

$$I = I(U)$$

↳ available for any U through solving a system of equations (expensive!)

Goal: replace $I = I(U)$ with a function $U \to \tilde{I}(U)$ that allows fast point evaluation (also of derivative)

Approximation of functions: Generic view

Given: function $\mathbf{f} : D \subset \mathbb{R}^n \mapsto \mathbb{R}^d$ (often in procedural form **double** f(**double**), Rem. 5.1.4)

Goal: Find a "SIMPLE" function $\tilde{\mathbf{f}} : D \mapsto \mathbb{R}^d$ such that the approximation error $\mathbf{f} - \tilde{\mathbf{f}}$ is "SMALL"

a function

measured w.r.t. some norm

Here: $n = 1$

e.g. polynomials

- Supremum norm: $\|f\|_\infty = \max_{t \in D} |f(t)|$
- L^2-norm: $\|f\|_2^2 := \int_D |f(t)|^2 \, dt$

Policy: Approximation by interpolation

Mapping functions → functions

Interpolation scheme + sampling → approximation scheme : $A : C^0(I) \to V$

$$f : I \subset \mathbb{R} \to \mathbb{K} \xrightarrow{\text{sampling}} \overbrace{(t_i, y_i := f(t_i))_{i=0}^m}^{\text{data points}} \xrightarrow{\text{interpolation}} \tilde{f} := \mathsf{I}_T \mathbf{y} \quad (\tilde{f}(t_i) = y_i) .$$

free choice of nodes $t_i \in I$

New freedom: choice of nodes

②

6.1. Approximation by global Polynomials

Known: Approximation by Taylor polynomial

$$f \in C^{k+1} : f(t) \approx \underbrace{f(t_0) + f'(t_0)(t-t_0) + f''(t_0)\tfrac{1}{2}(t-t_0)^2 + \dots + f^{(k)}(t_0)\tfrac{1}{k!}(t-t_0)^k}_{\in \mathcal{P}_k}$$

smoothness requirement

4.1.1. Polynomial Approximation: Theory

① Uniform approximation = approx. in $\|\cdot\|_\infty = \|\cdot\|_{L^\infty(I)}$

Theorem 6.1.6. Uniform approximation by polynomials

For $f \in C^0([0,1])$, define the n-th Bernstein approximant as — Bernstein polynomials

$$p_n(t) = \sum_{j=0}^{n} f(j/n)\binom{n}{j} t^j (1-t)^{n-j}, \quad p_n \in \mathcal{P}_n . \tag{6.1.7}$$

It satisfies $\|f - p_n\|_\infty \to 0$ for $n \to \infty$. If $f \in C^m([0,1])$, then even $\left\|f^{(k)} - p_n^{(k)}\right\|_\infty \to 0$ for $n \to \infty$ and all $0 \le k \le m$.

$\to$ $p_n \to f$ converges "slowly"

② Quantitative uniform best approximation

Theorem 6.1.15. L^∞ polynomial best approximation estimate

If $f \in C^r([-1,1])$ (r times continuously differentiable), $r \in \mathbb{N}$, then, for any polynomial degree $n \ge r$,

$$\underbrace{\inf_{p \in \mathcal{P}_n} \|f - p\|_{L^\infty([-1,1])}}_{} \le (1 + \pi^2/2)^r \frac{(n-r)!}{n!} \left\|f^{(r)}\right\|_{L^\infty([-1,1])} .$$

(norm of) best approximation error — smoothness requ.

- Asymptotics for $n \to \infty$: $O(n^{-r})$ *

* = algebraic convergence (in degree) [with rate r]

Example for asymptotic behavior of family $\{A_n\}$ of approximation schemes:

How does $\|f - A_n f\|$ behave as a function of $n \to \infty$

- Note: Smoothness of f governs rate r of alg. cvg.

$$f \in C^\infty \Rightarrow \forall r \in \mathbb{N} : \exists C = C(r)$$

$$\inf_{p \in \mathcal{P}_n} \|f - p\| \le C(r)\, n^{-r}$$

$\to \infty$ as $r \to \infty$ possible

③

Transformation to arbitrary intervals $[a,b]$:

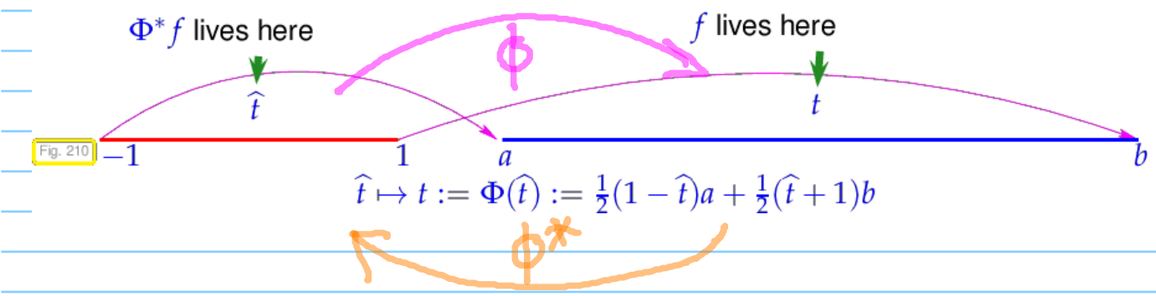
Tool: affine transformation $\Phi : [-1,1] \longrightarrow [a,b]$

$$\Phi(\hat{t}) = a + \tfrac{1}{2}(b-a)(\hat{t}+1) \quad , -1 \le \hat{t} \le 1$$

Pullback of functions: $(\Phi^* f)(\hat{t}) := f(\Phi(\hat{t}))$

for $f : [a,b] \longrightarrow \mathbb{R}$

$$\boxed{\Phi^* : C^0([a,b]) \to C^0([-1,1]) \quad , \quad \Phi^*(f)(\hat{t}) := f(\Phi(\hat{t})) \, , \quad -1 \le \hat{t} \le 1} \, . \tag{6.1.20}$$



$$\boxed{\Phi^* f := f \circ \Phi}$$

- $\Phi^* : \mathcal{P}_n \longrightarrow \mathcal{P}_n$ bijective
- $\|\Phi^* f\|_{\infty,[-1,1]} = \|f\|_{\infty,[a,b]}$
- "$\Phi^* \leftrightarrow$ stretching/compressing"

$$(\Phi^* f)' = \Phi^*(f') \circ \Phi' = \tfrac{1}{2}(b-a)\, \Phi^*(f')$$

[chain rule]

$$(\Phi^* f)^{(r)} = \left(\frac{b-a}{2}\right)^r \Phi^*(f^{(r)}) \, . \tag{6.1.28}$$

Lemma 6.1.24 ▶

$$\left\|(\Phi^* f)^{(r)}\right\|_{L^\infty([-1,1])} = \left(\frac{b-a}{2}\right)^r \left\|f^{(r)}\right\|_{L^\infty([a,b])} \, , \quad f \in C^r([a,b]), r \in \mathbb{N}_0 \, . \tag{6.1.29}$$

$$\inf_{p \in \mathcal{P}_n} \|f - p\|_{\infty,[a,b]} = \inf_{p \in \mathcal{P}_n} \|\Phi^*(f-p)\|_{\infty,[-1,1]}$$

$[\Phi^*(\mathcal{P}_n) = \mathcal{P}_n]$ $$= \inf_{p \in \mathcal{P}_n} \|\Phi^* f - \cancel{\Phi^*} p\|_{\infty,[-1,1]}$$

[Thm 6.1.15] $$\le C n^{-r} \|(\Phi^* f)^{(r)}\|_{\infty,[-1,1]}$$

$$\le C n^{-r} \left(\frac{b-a}{2}\right)^r \|f^{(r)}\|_{\infty,[a,b]}$$

depends on length of interval

6.1.2. Error estimates for Polynomial Interpolation

Definition 6.1.32. Lagrangian (interpolation polynomial) approximation scheme

Given an interval $I \subset \mathbb{R}$, $n \in \mathbb{N}$, a node set $\mathcal{T} = \{t_0, \dots, t_n\} \subset I$, the Lagrangian (interpolation polynomial) approximation scheme $\mathsf{L}_{\mathcal{T}} : C^0(I) \to \mathcal{P}_n$ is defined by

$$\mathsf{L}_{\mathcal{T}}(f) := \mathsf{I}_{\mathcal{T}}(\mathbf{y}) \in \mathcal{P}_n \quad \text{with} \quad \mathbf{y} := (f(t_0), \dots, f(t_n))^T \in \mathbb{K}^{n+1}.$$

New: Consider families $\mathcal{T}_n = \{t_0^{(n)}, \dots, t_n^{(n)}\}$ of nodes, e.g.

equidistant nodes $t_j^{(n)} = a + \frac{b-a}{n} j$, $j = 0, \dots, n$

$\Rightarrow$ family of polynomial approx. schemes $\{A_n\}_{n \in \mathbb{N}}$

We study $\|f - A_n f\| \leq T(n)$ for $n \to \infty$

Ex:

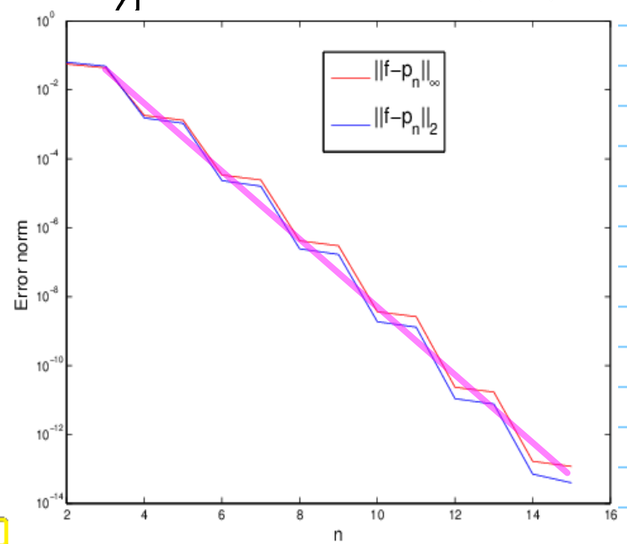


Fig. 211

$\triangleleft$ $f(t) = \sin(t)$

$I = [0, \pi]$

equid. nodes

[lin-log plot!]

$\varepsilon_n := \|f - \mathsf{L}_{\mathcal{T}_n} f\|$

Observed: $\log \varepsilon_n \approx C - Kn$, $K > 0$

$\varepsilon_n \approx e^C (e^{-K})^n$

$\Leftrightarrow$ $\varepsilon_n = O(q^n)$ for $n \to \infty$ for some $0 \leq q < 1$

$\hookrightarrow$ exponential cvg. (in degree n)

Algebraic convergence: $\|f - \mathsf{I}_{\mathcal{T}} f\| = O(n^{-p})$ (rate)

Exponential convergence: $\|f - \mathsf{I}_{\mathcal{T}} f\| = O(q^n)$

for $n \to \infty$ ("asymptotic!")

Ex: $f(t) = \frac{1}{1+t^2}$, $t \in \mathbb{R}$, $I = [-5, 5]$

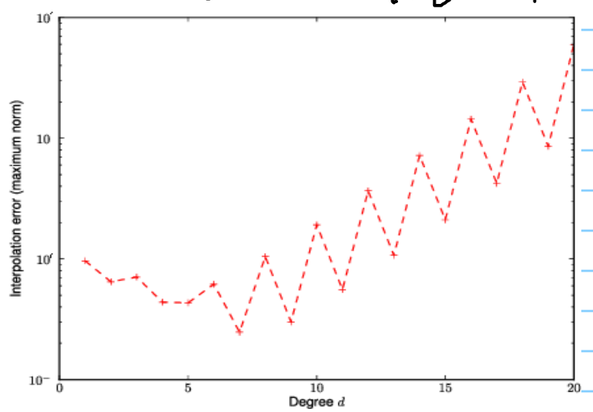


Fig. 213

"Equidistant interpolation"

Approximate $\|f - \mathsf{L}_{\mathcal{T}_n} f\|_\infty$ on $[-5, 5]$

5

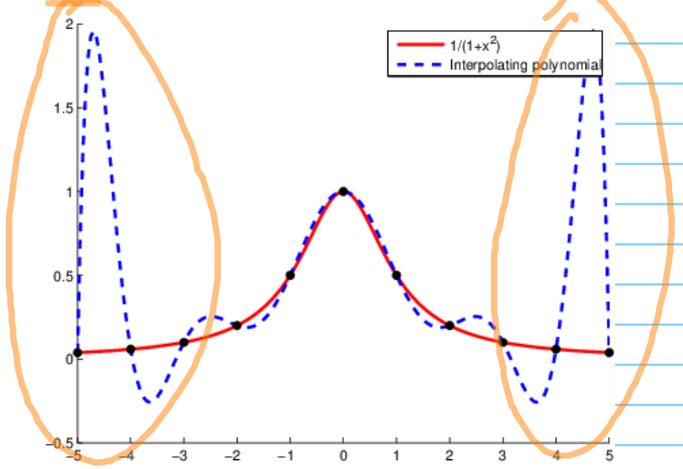


Fig. 212

Interpolating polynomial, $n = 10$

Theorem 6.1.44. Representation of interpolation error [?, Thm. 8.22], [?, Thm. 37.4]

We consider $f \in C^{n+1}(I)$ and the Lagrangian interpolation approximation scheme (→ Def. 6.1.32) for a node set $\mathcal{T} := \{t_0, \dots, t_n\} \subset I$. Then,

for every $t \in I$ there exists a $\tau_t \in]\min\{t, t_0, \dots, t_n\}, \max\{t, t_0, \dots, t_n\}[$ such that

smoothness requirement

$$\underbrace{f(t) - \mathsf{L}_{\mathcal{T}}(f)(t)}_{\text{interpolation error in } t} = \frac{f^{(n+1)}(\tau_t)}{(n+1)!} \cdot \prod_{j=0}^{n}(t - t_j) \,. \qquad (6.1.45)$$

$n \hat{=}$ degree

$=: w(t)$ [nodal polynomial]

Proof: fix $t \in I \setminus \mathcal{T}$

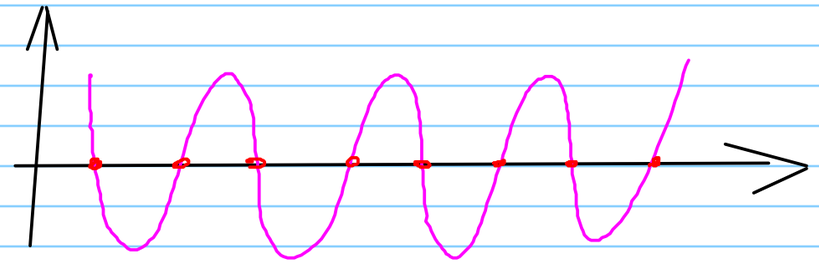
choose $c \in \mathbb{R}$ s.t. $f(t) - \mathsf{L}_{\mathcal{T}} f(t) - c\, w(t) = 0$

$$\varphi(x) := f(x) - \underbrace{\mathsf{L}_{\mathcal{T}} f(x)}_{\in \mathcal{P}_n} - c\, w(x) \in C^{n+1}$$

$\varphi(t_j) = 0$ by interpolation conditions

$\varphi(t) = 0$ by choice of c

↳ has $n+2$ distinct zeros



→ φ' has $n+1$ zeros

φ'' has n zeros

⋮

$\varphi^{(n+1)}$ has at least 1 zero τ_t

$\varphi^{(n+1)}(x) = f^{(n+1)}(x) - 0 - c\,(n+1)!$

$\varphi^{(n+1)}(\tau_t) = 0$ gives the formula

$$c = \frac{f^{(n+1)}(\tau_t)}{(n+1)!}$$

□

6

Take sup over $t \in [t_0, t_n]$ in (6.1.45)

Thm. 6.1.44 $\Rightarrow$ $$\boxed{\|f - \mathsf{L}_T f\|_{L^\infty(I)} \le \frac{\|f^{(n+1)}\|_{L^\infty(I)}}{(n+1)!} \max_{t\in I} |(t-t_0)\cdot\cdots\cdot(t-t_n)|} \quad . \qquad (6.1.50)$$

- $f(t) = \sin(t)$ $\triangleright$ $\|f^{(n)}\|_\infty \le 1 \quad \forall n \in \mathbb{N}$

  $f \in C^\infty$

$\left\|f^{(k)}\right\|_{L^\infty(I)} \le 1, \quad \forall k \in \mathbb{N}_0$ $\Rightarrow$ $$\|f-p\|_{L^\infty(I)} \le \frac{1}{(1+n)!} \max_{t\in I} \underbrace{\left|(t-0)(t-\tfrac{\pi}{n})(t-\tfrac{2\pi}{n})\cdot\cdots\cdot(t-\pi)\right|}_{\text{Extremal for } t \approx \frac{\pi}{2n}} \le \frac{1}{n+1}\left(\frac{\pi}{n}\right)^{n+1} .$$

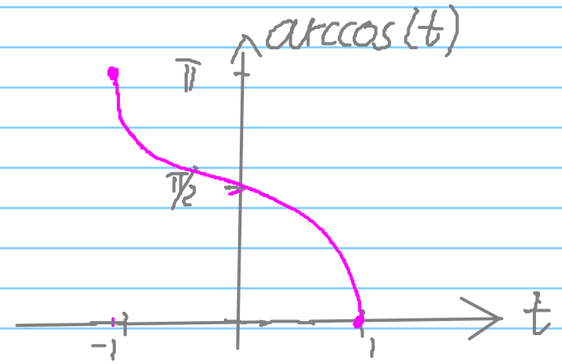
$I = [0,\pi]$

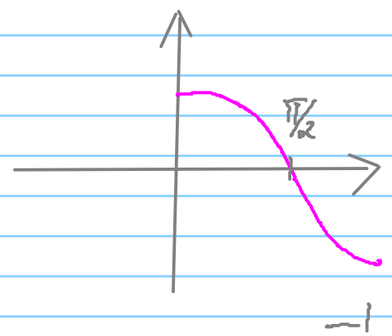
superexponential convergence

exp. cvg. $\|\text{error}\| = O(q^n)$

- $f(t) = \frac{1}{1+t^2}$ $\Rightarrow$ $\|f^{(n)}\|_\infty \sim 2^n n!$ for $n \to \infty$

$\to$ Exponential blow-up of bound in (6.1.50)





4.1.3. Chebychev Interpolation

Goal: n/f-independent a priori "optimal" choice of J

$\hookrightarrow$ beyond our control

Thm. 6.1.44 $\Rightarrow$ $$\boxed{\|f - \mathsf{L}_T f\|_{L^\infty(I)} \le \frac{\|f^{(n+1)}\|_{L^\infty(I)}}{(n+1)!} \underbrace{\max_{t\in I} |(t-t_0)\cdot\cdots\cdot(t-t_n)|}_{= \|w\|_\infty}} \quad . \qquad (6.1.50)$$

$\leftarrow$ make this as small as possible

$w \in \mathcal{P}_{n+1}$ w/ leading coeff $= 1$

Optimal choice of interpolation nodes independent of interpoland

Idea: choose nodes $t_0, \dots, t_n$ such that $\|w\|_{L^\infty(I)}$ is minimal!

This is equivalent to finding a polynomial $q \in \mathcal{P}_{n+1}$

- with leading coefficient $= 1$,
- such that it minimizes the norm $\|q\|_{L^\infty(I)}$.

$\blacktriangleright$ Then choose nodes $t_0, \dots, t_n$ as zeros of q (caution: t_j must belong to I).

$\hookrightarrow$ fix q already!

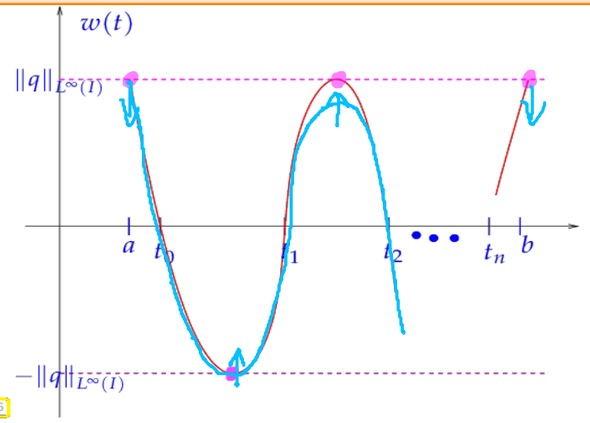


Fig. 215

Heuristic demands:

- t^* extremal point of q $\rightarrow$ $|q(t^*)| = \|q\|_{L^\infty(I)}$,
- q has $n+1$ zeros in I $(*)$,
- $|q(-1)| = |q(1)| = \|q\|_{L^\infty(I)}$.

- $|q| = \|q\|_\infty$ at all extrema

7 Solution polynomials:

Definition 6.1.75. Chebychev polynomials → [?, Ch. 32]

The nth Chebychev polynomial is $T_n(t) := \cos(n \arccos t)$, $\quad -1 \le t \le 1, n \in \mathbb{N}$.

▷ $|T_n(t)| \le 1 \quad \forall\, -1 \le t \le 1$

Theorem 6.1.76. 3-term recursion for Chebychev polynomials → [?, (32.2)]

The function T_n defined in Def. 6.1.75 satisfy the 3-term recursion

$$T_{n+1}(t) = 2t\,T_n(t) - T_{n-1}(t) \quad , \quad T_0 \equiv 1\,, \quad T_1(t) = t\,, \quad n \in \mathbb{N}\,. \tag{6.1.77}$$

▷ $T_n \in \mathcal{P}_n$ with leading coeff. 2^{n-1}

[$\cos(n+1)x = 2\cos nx \cdot \cos x - \cos(n-1)x$]

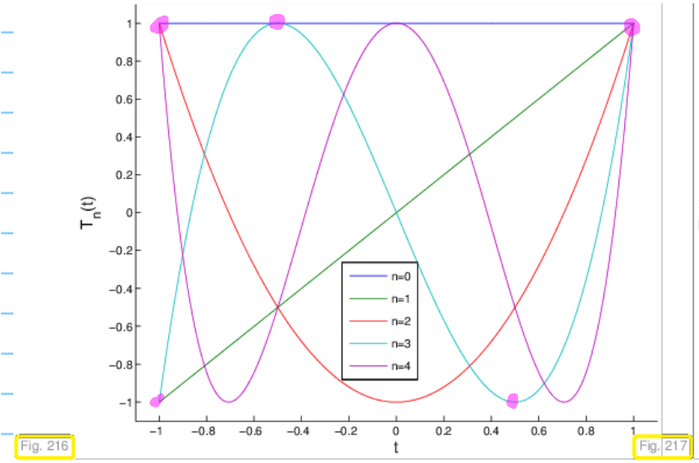


Fig. 216

Chebychev polynomials $T_0, \dots, T_4$

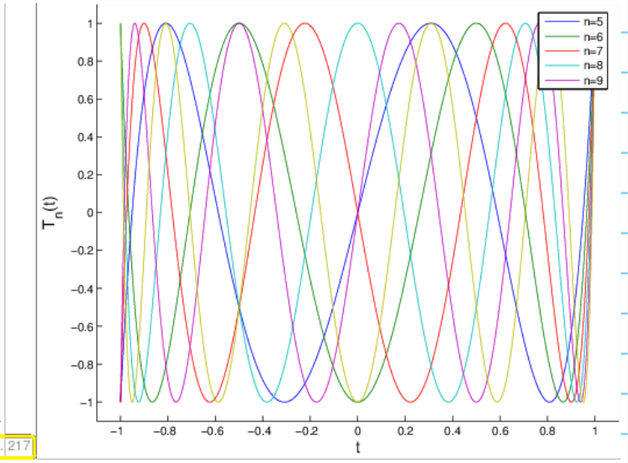


Fig. 217

Chebychev polynomials $T_5, \dots, T_9$

▷ Choose t_j s.t. $W := \frac{1}{2^n} T_{n+1}$

$\cos x = 0 \iff x \in (2\mathbb{N}+1)\pi/2$

▷ $t_j = \cos \frac{2j+1}{2n}\pi$ [Chebychev nodes]

$j = 0, \dots, n$

Optimal, because

Theorem 6.1.81. Minimax property of the Chebychev polynomials [?, Section 7.1.4.], [?, Thm. 32.2]

The polynomials T_n from Def. 6.1.75 minimize the supremum norm in the following sense:

$$\|T_n\|_{L^\infty([-1,1])} = \inf\{\|p\|_{L^\infty([-1,1])} : p \in \mathcal{P}_n, p(t) = 2^{n-1}t^n + \cdots\}\,, \quad \forall n \in \mathbb{N}\,.$$

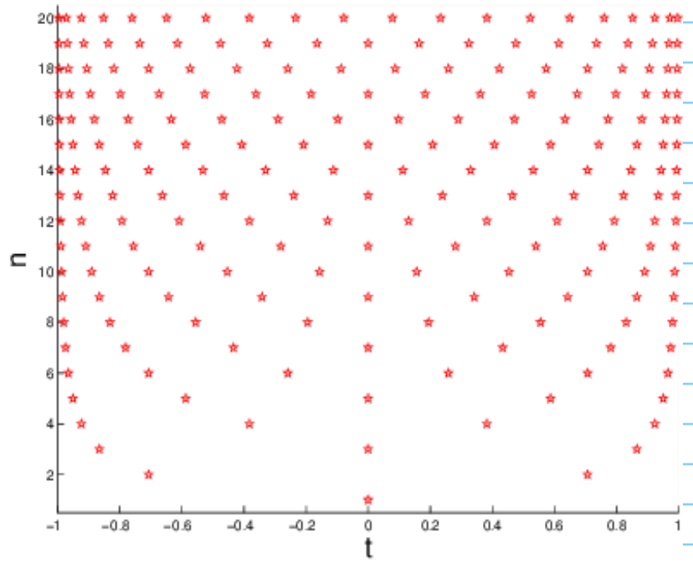


Fig. 218

△ Chebychev nodes cluster at endpoints

8 Example: $f(t) = \frac{1}{1+t^2}$ on $I = [-5,5]$

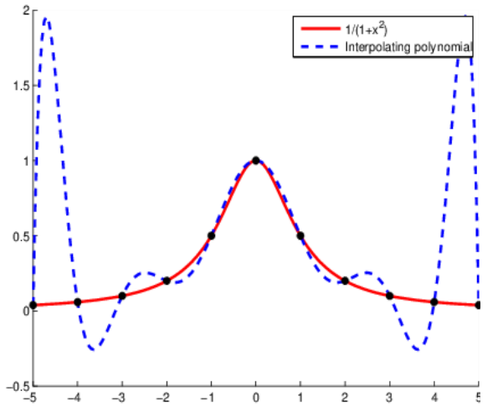


Fig. 220

Equidistant nodes

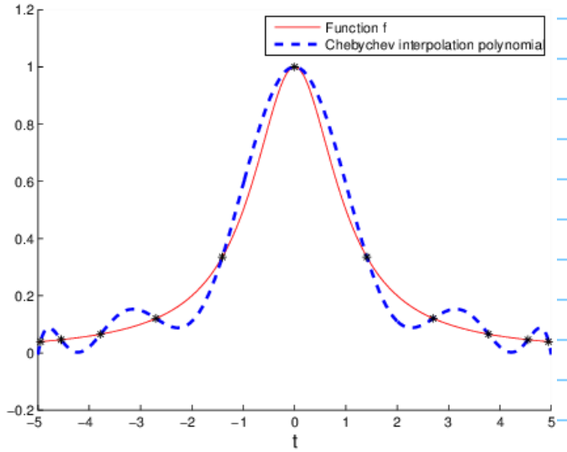


Fig. 221

Chebychev nodes

On $[a,b]$: transform nodes by $\hat{t} \to a + \frac{1}{2}(b-a)(\hat{t}+1)$

The Chebychev nodes in the interval $I = [a,b]$ are

$$t_k := a + \tfrac{1}{2}(b-a)\left(\cos\left(\frac{2k+1}{2(n+1)}\pi\right)+1\right), \quad k = 0,\dots,n. \tag{6.1.86}$$

For Chebychev interpolation

$$w = \frac{1}{2^n} T_n$$

$$\triangleright \ \|w\|_\infty = \frac{1}{2^n}$$

$$\triangleright \ \|f - I_T(f)\|_{L^\infty(I)} = \left\|\hat{f} - I_{\hat{T}}(\hat{f})\right\|_{L^\infty([-1,1])} \le \frac{2^{-n}}{(n+1)!}\left\|\frac{d^{n+1}\hat{f}}{d\hat{t}^{n+1}}\right\|_{L^\infty([-1,1])}$$

Refined estimate for Chebychev interpolation

$$\le \frac{2^{-2n-1}}{(n+1)!}|I|^{n+1}\left\|f^{(n+1)}\right\|_{L^\infty(I)}. \tag{6.1.87}$$

$\hookrightarrow |I| = |b-a|$ for $I = [a,b]$

Interpolation error estimates and Lebesgue constant

$I_J \hat{=}$ polynomial interpolation operator for $J = \{t_0,\dots,t_n\}$

$$\lambda_J := \sup_{x \in \mathbb{R}^n} \frac{\|I_J x\|_\infty}{\|x\|_\infty}$$

$[\ I_J : \mathbb{R}^{n+1} \to \mathcal{P}_n\ ]$

$L_J \hat{=}$ approximation scheme based on I_J

$[\ L_J : C^0(I) \to \mathcal{P}_n : L_J = I_J[f(t_0),\dots,f(t_n)]^T$

Note: $\|L_J g\|_\infty = \|I_J[g(t_0),\dots,g(t_n)]^T\|_\infty$

$\le \lambda_J \max_j |g(t_j)|$ (*)

$\le \lambda_J \|g\|_\infty$

$\|f - L_J f\|_\infty = \|(f-p) - (L_J f - L_J p)\|_\infty$, any $p \in \mathcal{P}_n$

interpolation error $= \|(f-p) - L_J(f-p)\|_\infty$ by linearity

Note: $L_J p = p$ $\le \|f-p\|_\infty + \|L_J(f-p)\|_\infty$

$\overset{(*)}{\le} (1+\lambda_J)\|f-p\|_\infty$

$$\|f - L_T f\|_{L^\infty(I)} \le (1+\lambda_T) \inf_{p \in \mathcal{P}_n} \|f-p\|_{L^\infty(I)} \quad \forall f \in C^0(I), \tag{6.1.61}$$

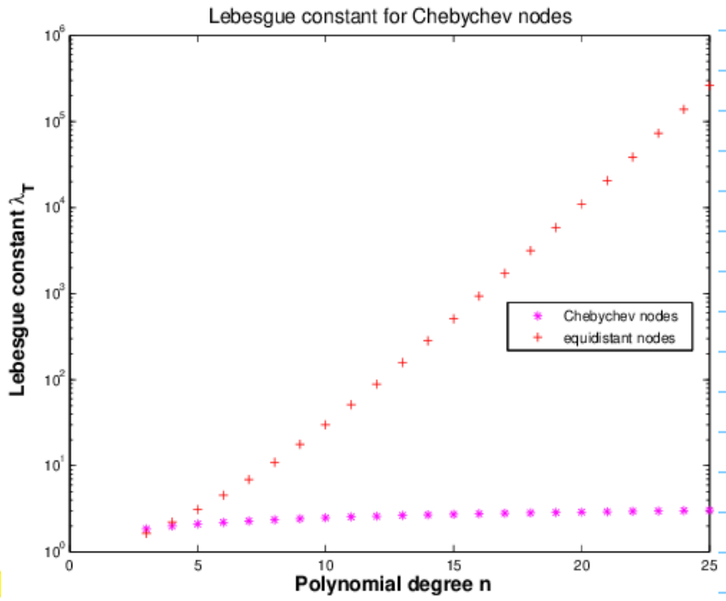


Fig. 173

Equidistant nodes

$\lambda_J \geq C e^{n/2}$

Chebychev nodes

$\lambda_T \leq \frac{2}{\pi} \log(1+n) + 1$

combine with Jackson estimates

Thm 6.1.15

$$\|f - \mathsf{L}_T f\|_{L^\infty([-1,1])} \leq (2/\pi \log(1+n) + 2)(1 + \pi^2/2)^r \frac{(n-r)!}{n!} \left\| f^{(r)} \right\|_{L^\infty([-1,1])} . \qquad (6.1.91)$$

▷ Applies to $f \in C^r$ ⇒ obtain n-asymptotics

$$\|f - \mathsf{I}_T(f)\|_{L^\infty(I)} = \left\| \widehat{f} - \mathsf{I}_{\widehat{T}}(\widehat{f}) \right\|_{L^\infty([-1,1])} \leq \frac{2^{-n}}{(n+1)!} \left\| \frac{\mathrm{d}^{n+1}\widehat{f}}{\mathrm{d}\widehat{t}^{n+1}} \right\|_{L^\infty([-1,1])}$$

$$\leq \frac{2^{-2n-1}}{(n+1)!} |I|^{n+1} \left\| f^{(n+1)} \right\|_{L^\infty(I)} . \qquad (6.1.87)$$

(usually) prevents us from extracting behavior for $n \to \infty$

Experiments: "n-cvg"

(i) $f(t) = \frac{1}{1+t^2}$, $I = [-5,5]$

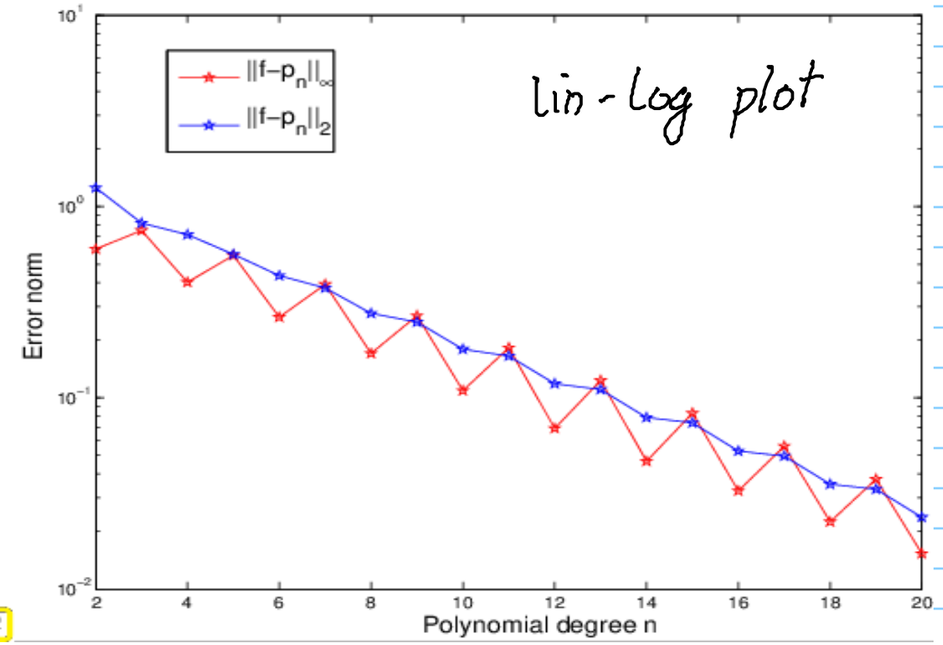


Fig. 222

⇒ exp. cvg.

("rather generally true" for $f \in C^\infty$)

(10) "Tent function"

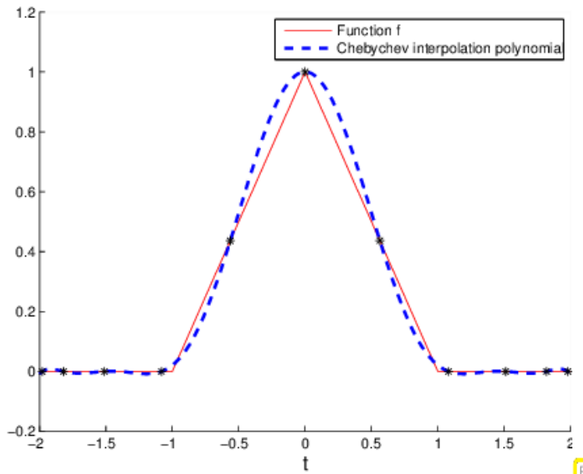


Fig. 223

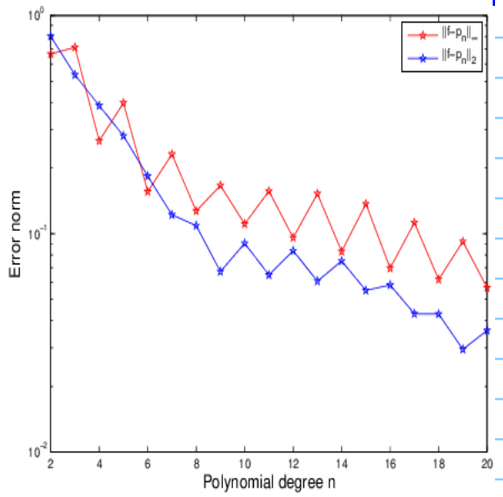


Fig. 224

still converges

6.1.3.3. Algorithms for Chebychev interpolation

C++11 code 6.1.99: Definition of class for Chebychev interpolation

```
class ChebInterp {
  private:
  // various internal data describing Chebychev interpolating
     polynomial p
  public:
    // Constructor taking function f and degree n as arguments
    template <typename Function>
      PolyInterp(const Function &f,unsigned int n);
    // Evaluation operator:  y_j = p(x_j),  j = 1,...,m  (m "large")
    void eval(const vector<double> &x,vector <double> &y) const;
};
```

Cheb. interp. polynomial

Chebychev expansion : $p(t) = \sum_{j=0}^{n} \alpha_j T_j(t)$

Fast evaluation : "Chebychev Horner scheme"

Idea : Use 3-term recursion

$$T_j(x) = 2x\,T_{j-1}(x) - T_{j-2}(x)$$

$$p(x) = \sum_{j=0}^{n-1} \alpha_j T_j(x) + \alpha_n T_n(x)$$

$$\overset{(6.1.77)}{=} \sum_{j=0}^{n-1} \alpha_j T_j(x) + \alpha_n (2xT_{n-1}(x) - T_{n-2}(x))$$

$$= \underbrace{\sum_{j=0}^{n-3} \alpha_j T_j(x) + (\alpha_{n-2} - \alpha_n) T_{n-2}(x) + (\alpha_{n-1} + 2x\alpha_n) T_{n-1}(x)}_{\text{another Cheb.-exp.} \;\rightarrow\; \text{recursion}} .$$

C++11 code 6.1.103: Recursive evaluation of Chebychev expansion (6.1.100)

```
// Recursive evaluation of a polynomial p = Σ_{j=1}^{n+1} a_j T_{j-1} at point x
// based on (6.1.102)
// IN : Vector of coefficients a
// evaluation point x
// OUT: Value at point x
double recclenshaw(const VectorXd& a, const double x) {
  const VectorXd::Index n = a.size() - 1;
  if      (n == 0) return a(0);              // Constant polynomial
  else if (n == 1) return (x*a(1) + a(0));   // Value α_1 * x + α_0
  else {
    VectorXd new_a(n);
    new_a << a.head(n - 2), a(n - 2) - a(n), a(n - 1)+ 2*x*a(n);
    return recclenshaw(new_a,x); // recursion
  }
}
```

$\Rightarrow$ Cost $= O(n)$

10

Computing α_j :

I.C. $\Leftrightarrow$ $p(t_k) = f(t_k)\,,\quad k = 0,\dots,n\,,$ for $t_k := \cos\left(\frac{2k+1}{2(n+1)}\pi\right)$. (6.1.106)

Chebychev nodes

$\hookrightarrow (n\times 1)\times(n+1)$ LSE

Trick : Periodize p : $T_n(t) = \cos(n\cos^{-1}(t))$

$$q(s) := p(\cos 2\pi s) = \sum_{j=0}^{n} \alpha_j T_j(\cos 2\pi s) \overset{\text{Def. 6.1.75}}{=} \sum_{j=0}^{n} \alpha_j \cos(2\pi j s)$$

$$= \sum_{j=0}^{n} \tfrac{1}{2}\alpha_j\left(\exp(2\pi\imath j s) + \exp(-2\pi\imath j s)\right) \quad [\text{ by } \cos z = \tfrac{1}{2}(e^{z} + e^{-z})\,]$$

(6.1.107)

$$= \sum_{j=-n}^{n+1} \beta_j \exp(-2\pi\imath j s)\,, \quad \text{with} \quad \beta_j := \begin{cases} 0 & \text{, for } j = n+1\,, \\ \frac{1}{2}\alpha_j & \text{, for } j = 1,\dots,n\,, \\ \alpha_0 & \text{, for } j = 0\,, \\ \frac{1}{2}\alpha_{n-j} & \text{, for } j = -n,\dots,-1\,. \end{cases}$$

[trigonometric polynomial]

I.C. for q $\Longrightarrow$ I.C. for q at nodes

$t = \cos(2\pi s)$ $\longrightarrow$

$t_k = \cos\left(\frac{2k+1}{4(n+1)}2\pi\right)$ $\longrightarrow$ $s_k = \frac{2k+1}{4(n+1)}$

[equidistant nodes]

▷ FFT-based alg. for comp. $\beta_j \Longrightarrow \alpha_j$

Cost $O(n\log n)$

MATLAB-code 6.1.111: Efficient computation of Chebychev expansion coefficient of Chebychev interpolant

```
// efficiently compute coefficients α_j in the Chebychev expansion
// p = Σ_{j=0}^{n} α_j T_j of p ∈ P_n based on values y_k,
// k = 0,...,n, in Chebychev nodes t_k, k = 0,...,n
// IN: values y_k passed in y
// OUT: coefficients α_j
VectorXd chebexp(const VectorXd& y) {
  const int n = y.size() - 1;          // degree of polynomial
  const std::complex<double> M_I(0, 1); // imaginary unit
  // create vector z, see (6.1.109)
  VectorXcd b(2*(n + 1));
  const std::complex<double> om = -M_I*(M_PI*n)/((double)(n+1));
  for (int j = 0; j <= n; ++j) {
    b(j) = std::exp(om*double(j))*y(j); // this cast to double is
      necessary!!
    b(2*n+1-j) = std::exp(om*double(2*n+1-j))*y(j);
  }

  // Solve linear system (6.1.110) with effort O(n log n)
  Eigen::FFT<double> fft;  // EIGEN's helper class for DFT
  VectorXcd c = fft.inv(b); // -> c = ifft(z), inverse fourier
    transform
  // recover β_j, see (6.1.110)
  VectorXd beta(c.size());
  const std::complex<double> sc = M_PI_2/(n + 1)*M_I;
  for (unsigned j = 0; j < c.size(); ++j)
    beta(j) = ( std::exp(sc*double(-n+j))*c[j] ).real();
  // recover α_j, see (6.1.107)
  VectorXd alpha = 2*beta.tail(n); alpha(0) = beta(n);
  return alpha;
}
```

6.5. Approximation by piecewise polynomials

For Chebychev interpolation:

$$\|f - \mathsf{I}_{\mathcal{T}}(f)\|_{L^\infty(I)} = \left\|\widehat{f} - \mathsf{I}_{\widehat{\mathcal{T}}}(\widehat{f})\right\|_{L^\infty([-1,1])} \leq \frac{2^{-n}}{(n+1)!}\left\|\frac{\mathrm{d}^{n+1}\widehat{f}}{\mathrm{d}\widehat{t}^{n+1}}\right\|_{L^\infty([-1,1])}$$

$$\leq \frac{2^{-2n-1}}{(n+1)!}|I|^{n+1}\left\|f^{(n+1)}\right\|_{L^\infty(I)} . \qquad (6.1.87)$$

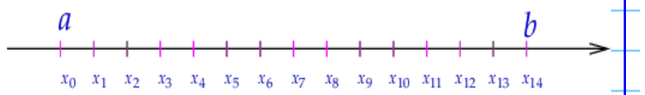
▷ reduce interpolation by shrinking the interval

Idea: piecewise interpolation on $I := [a,b]$

Tool: $\mathcal{M} = \{a = x_0 < x_1 < \dots < x_m = b\}$

Terminology:

- $x_j \hat{=}$ nodes of the mesh $\mathcal{M}$,
- $[x_{j-1}, x_j[\hat{=}$ intervals/cells of the mesh,
- $h_{\mathcal{M}} := \max_j |x_j - x_{j-1}| \hat{=}$ mesh width,
- If $x_j = a + jh \hat{=}$ equidistant (uniform) mesh with meshwidth $h > 0$



Then: (local) Lagrange interpolation on cells

General local Lagrange interpolation on a mesh

1. Choose local degree $n_j \in \mathbb{N}_0$ for each cell of the mesh, $j = 1, \dots, m$.
2. Choose set of *local* interpolation nodes

$$\mathcal{T}^j := \{t_0^j, \dots, t_{n_j}^j\} \subset I_j := [x_{j-1}, x_j] , \quad j = 1, \dots, m ,$$

for each mesh cell/grid interval I_j.

3. Define piecewise polynomial interpolant $s : [x_0, x_m] \to \mathbb{K}$:

$$s_j := s_{|I_j} \in \mathcal{P}_{n_j} \quad \text{and} \quad s_j(t_i^j) = f(t_i^j) \quad i = 0, \dots, n_j , \quad j = 1, \dots, m . \qquad (6.5.5)$$

Owing to Thm. 5.2.14, s_j is well defined.

p.w. linear intp.: $n_j = 1$, $t_0^j = x_{j-1}$, $t_1^j = x_j$

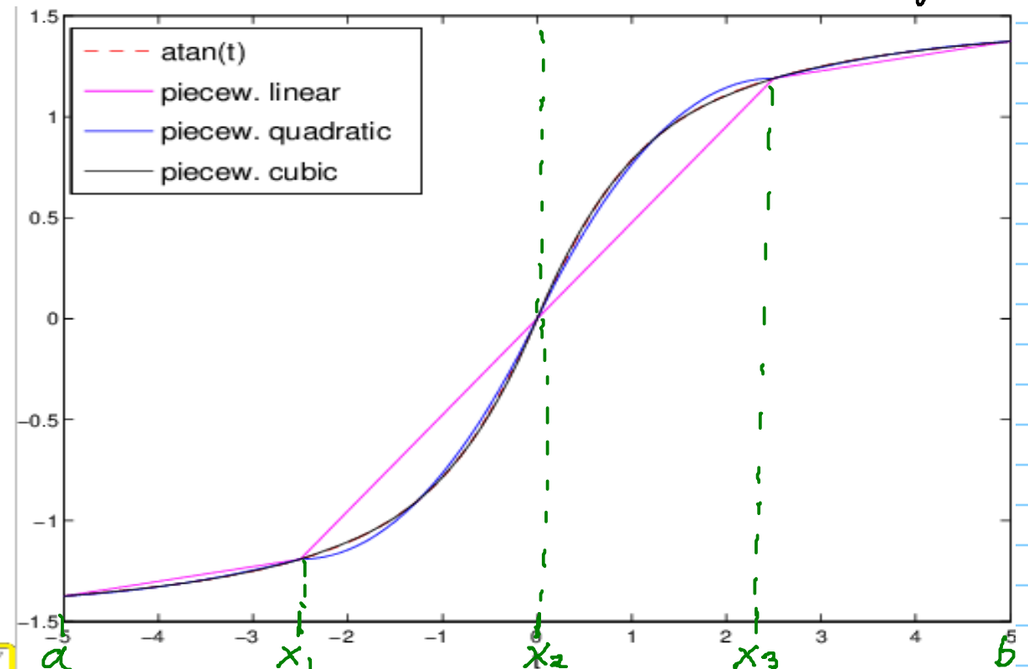


Fig. 247

p.w. quadr : $n_j = 2$: $t_0^j = x_{j-1}$

$t_1^j = \frac{1}{2}(x_{j-1} + x_j)$

$t_2^j = x_j$

C^0-p.w. interpolant guaranteed if all x_j's are interpolation nodes!

[No C^1 interpolant, though!]

Error estimates for special case $n_j = n$ (fixed):

Improve accuracy by $h_M \longrightarrow 0$ [h-convergence]

Example: $f(t) = \operatorname{atan}(t)$ on $[-5,5]$

Equidistant mesh with $10/h_M$ cells, equid int. nodes

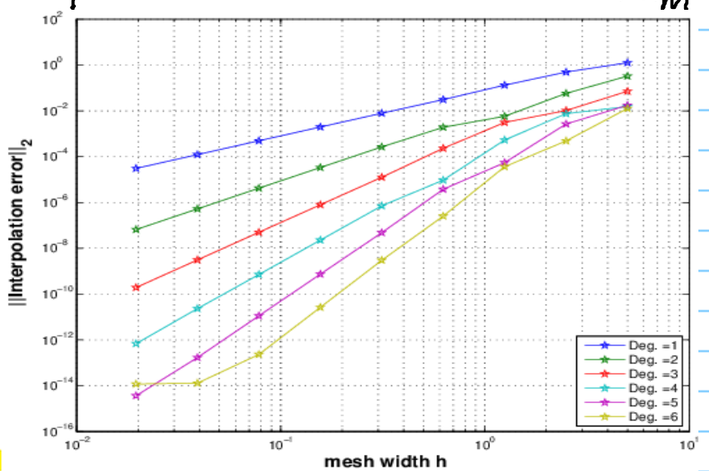


Fig. 248

log-log plot & linear error curves

$\triangleright$ alg. cvg. in h_M

Theory : Apply on $I := [x_{j-1}, x_j]$, $x_{j-1} \le t_j \le x_j$

Thm. 6.1.44 $\Rightarrow$ $$\boxed{\|f - \mathsf{L}_T f\|_{L^\infty(I)} \le \frac{\|f^{(n+1)}\|_{L^\infty(I)}}{(n+1)!} \underbrace{\max_{t\in I}|(t-t_0)\cdot\cdots\cdot(t-t_n)|}_{\le h_M^{n+1}}} . \qquad (6.1.50)$$

then max over all cells:

$$(6.1.50) \Rightarrow \|f - s\|_{L^\infty([x_0,x_m])} \le \frac{h_{\mathcal{M}}^{n+1}}{(n+1)!} \left\|f^{(n+1)}\right\|_{L^\infty([x_0,x_m])} , \qquad (6.5.12)$$

with mesh width $h_{\mathcal{M}} := \max\{|x_j - x_{j-1}| : j = 1,\dots,m\}$.

$\downarrow$

Alg. cvg. in h_M with rate $n+1$!

Advantage : (I) Poor smoothness of f still guarantees alg. cvg. in h_M

(II) Suitable also for merely piecewise smooth functions

(III) Locality $\Rightarrow$ update friendly

(IV) simple & cheap : cost $= O(m)$

Drawback : "Slow" algebraic convergence

p.w. quadratic $(n = 2)$