Numerical Methods for Computational Science and Engineering

Autumn Semester 2018, Week 7

Prof. Rima Alaifari, SAM, ETH Zurich

Global Polynomial Interpolation

Before: subspace S → "search space" for the interpolating function

$$\mathcal{P}_n := \{ t \mapsto \underbrace{\sum_{i=0}^{n} \alpha_i t^i}, \ \alpha_i \in \mathbb{R} \}$$

polynomials of degree $\leq n$

monomial repr. of a polyn.

monomials form a basis for $\mathcal{P}_n$:

$t \mapsto t^i$

$p(t) = \alpha_n t^n + \dots + \alpha_1 t + \alpha_0$

linear combination of basis functions $t \mapsto t^i$

$\dim \mathcal{P}_n = n+1$

$\mathcal{P}_n \subset C^\infty(\mathbb{R})$

infinitely many times diff. functions

"smooth" functions

$\Rightarrow$ We need $n+1$ points to determine polynomial of degree n

advantages of polynomials:

- differentiation & integration both easy to compute
- efficient evaluation through Horner scheme
- approximation property of polynomials

Horner scheme:

$$p(t) = t\left(\dots t\left(t\left(\alpha_n t + \alpha_{n-1}\right) + \alpha_{n-2}\right) + \dots + \alpha_1\right) + \alpha_0$$

Computational complexity: $\Theta(n)$

General problem formulation:

Given the simple nodes $t_0, \dots, t_n$, $n \in \mathbb{N}$, $t_0 < t_1 < \dots < t_n$, and data values $y_0, \dots, y_n \in \mathbb{R}$, find $p \in \mathbb{P}_n$ s.t.

$$p(t_j) = y_j \qquad \text{for } j = 0, \dots, n.$$

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

First idea: monomial basis & build matrix A

but complexity: $\Theta(n^3)$

1. Lagrange interpolation

$A = I$?

$$L_i(t) := \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{t - t_j}{t_i - t_j} \qquad i = 0, \dots, n$$

- $L_i \in \mathbb{P}_n$ ✓
- Cardinal ? $\quad L_i(t_\ell) = \delta_{i\ell}$

$$(*) \quad L_i(t_\ell) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{t_\ell - t_j}{t_i - t_j} = \begin{cases} 0 & \text{if } \ell \neq i \\ 1 & \text{if } \ell = i \end{cases}$$

If $\mu_0 L_0(t) + \mu_1 L_1(t) + \dots + \mu_n L_n(t) = 0$

Can we imply $\mu_0 = \mu_1 = \dots = \mu_n = 0$???

Idea: Choose $t = t_i$

For t_i: $\mu_0 \underbrace{L_0(t_i)} + \mu_1 \underbrace{L_1(t_i)} + \dots + \mu_n L_n(t_i) = 0$

$$\overset{(*)}{\Longrightarrow} \quad \mu_i \underbrace{L_i(t_i)}_{=1} = 0$$

$$\Rightarrow \quad \mu_i = 0$$

Do this for all $i = 0, \dots, n$

$$\Rightarrow \mu_0 = \mu_1 = \dots = \mu_n = 0$$

$\Rightarrow$ $L_i(t)$, $i = 0, \dots, n$ are linearly independent

Since $\dim \mathcal{P}_n = n+1$

$\Rightarrow$ $\{L_i(t) : i = 0, \dots, n\}$ is a basis for $\mathcal{P}_n$

Moreover, it's a cardinal basis.

$$p(t) = \sum_{i=0}^{n} y_i L_i(t)$$

data points

Since $A = I \Rightarrow$ existence & uniqueness of interpolant

$$\left(A = \begin{bmatrix} b_0(t_0) & \cdots & b_n(t_0) \\ \vdots & & \\ b_0(t_n) & \cdots & b_n(t_n) \end{bmatrix} \right)$$

$b_0, \dots, b_n$ basis for S

Theorem 5.1.2 (Existence & uniqueness of Lagrange interpolation polynomial). *The general Lagrange polynomial interpolation problem admits a unique solution $p \in \mathcal{P}_n$.*

How do the L_i's look like?

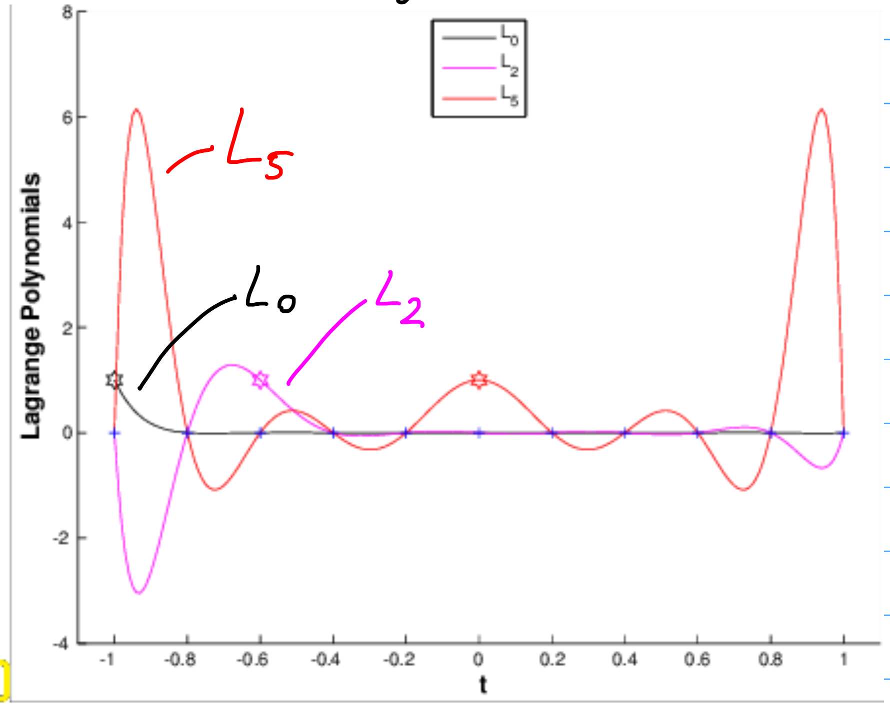


Fig. 170

A bit more on complexity:

- Monomial basis

$$A = \begin{bmatrix} 1 & t_0 & t_0^2 & \cdots & t_0^n \\ 1 & t_1 & t_1^2 & \cdots & t_1^n \\ \vdots & \vdots & & & \\ 1 & t_n & t_n^2 & \cdots & t_n^n \end{bmatrix}$$

Vandermonde matrix

$$A \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} y_0 \\ \vdots \\ y_n \end{bmatrix}$$

Effort for solving for $\underline{\alpha}$ is $\Theta(n^3)$.

- Lagrange polynomials:

$$p(t) = \sum_{i=0}^{n} y_i L_i(t)$$

Effort of evaluating $p(t)$ at some point t?

1. Evaluating each $L_i(t)$ with Horner scheme:

   $\Theta(n)$

2. Evaluating $p(t)$ is then: $\Theta(n^2)$

Series of interpolation problems:

with fixed nodes $t_0, \dots, t_n \in I$

N different data value vectors

$\{ y_0^k, \dots, y_n^k \}$ $k \in \{1, \dots, N\}$

For every k: find interpolant $p_k \in \mathcal{P}_n$

and evaluate $p_k(x_k)$, $x_k \in I$, $k \in \{1, \dots, N\}$

Complexity of Lagrange basis approach:

$$\underline{\Theta(n^2 N)}$$

Barycentric interpolation:

Lagrange interpolating polynomial:

$$p(t) = \sum_{i=0}^{n} y_i L_i(t) = \sum_{i=0}^{n} y_i \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{t - t_j}{t_i - t_j}$$

Define: $\lambda_i := \dfrac{1}{(t_i - t_0)(t_i - t_1) \cdots (t_i - t_{i-1})(t_i - t_{i+1}) \cdots (t_i - t_n)}$

$i = 0, \dots, n$

$$\boxed{p(t) = \sum_{i=0}^{n} y_i \lambda_i \prod_{\substack{j=0 \\ j\neq i}}^{n} (t-t_j)}$$

$\uparrow$ using λ_i

$$= \sum_{i=0}^{n} y_i \frac{\lambda_i}{t-t_i} \prod_{j=0}^{n} (t-t_j) \qquad (**)$$

Consider the constant polynomial $p_1(t) \equiv 1$

this would imply $y_{i,1} = 1$

and write down $(**)$:

$$1 = \sum_{i=0}^{n} 1 \cdot \frac{\lambda_i}{t-t_i} \underbrace{\prod_{j=0}^{n} (t-t_j)}_{\text{independent of } i}$$

$$1 = \left(\sum_{i=0}^{n} \frac{\lambda_i}{t-t_i}\right) \cdot \prod_{j=0}^{n} (t-t_j) \qquad /: \left(\sum_{i=0}^{n} \frac{\lambda_i}{t-t_i}\right)$$

$\Rightarrow$ Express $\prod_{j=0}^{n} (t-t_j)$ as

$$\prod_{j=0}^{n} (t-t_j) = \frac{1}{\sum_{i=0}^{n} \frac{\lambda_i}{t-t_i}}$$

Plug this into $(**)$:

$$p(t) = \frac{\sum_{i=0}^{n} y_i \frac{\lambda_i}{t-t_i}}{\sum_{i=0}^{n} \frac{\lambda_i}{t-t_i}}$$

Complexity?

- Cost of computing λ_i for some i: $\Theta(n)$
- Cost of computing $\lambda_0, \dots, \lambda_n$: in total $\Theta(n^2)$
- Cost of evaluating $p_k(x_k) = \dfrac{\sum_{i=0}^{n} y_i^k \frac{\lambda_i}{t-t_i}}{\sum_{i=0}^{n} \frac{\lambda_i}{t-t_i}}$ is $\Theta(n)$

(for each k) $\quad k = 1, \dots, N$

Total complexity: $\mathcal{O}(n^2 + nN)$

Note: If nodes t_i are close to each other $\longrightarrow$ numerical instabilities (either in the formula for $L_i(t)$ or in the formula λ_i)

Different approach: Newton basis

$$N_0(t) \equiv 1, \quad N_i(t) = \prod_{j=0}^{i-1} (t - t_j) \qquad i = 1, \dots, n$$

degree of N_i: i $\rightarrow$ linearly independent!

Find interpolant $p(t) = \sum_{i=0}^{n} a_i N_i(t)$

$$\forall\, l < i \qquad N_i(t_l) = \prod_{j=0}^{i-1} (t_l - t_j) = 0$$

$$A = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & N_1(t_1) & 0 & \cdots & \vdots \\ \vdots & N_1(t_2) & N_2(t_2) & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 1 & N_1(t_n) & N_2(t_n) & \cdots & N_n(t_n) \end{bmatrix} \quad \text{lower triang. matrix}$$

- Solving for $A \begin{bmatrix} a_0 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ \vdots \\ y_n \end{bmatrix}$ is $\mathcal{O}(n^2)$
- Assembling the matrix A: $\mathcal{O}(n^2)$ [N_i has deg i]

N evaluations for $[y_0^k, \dots, y_n^k]$ $\quad k \in \{1, \dots, N\}$

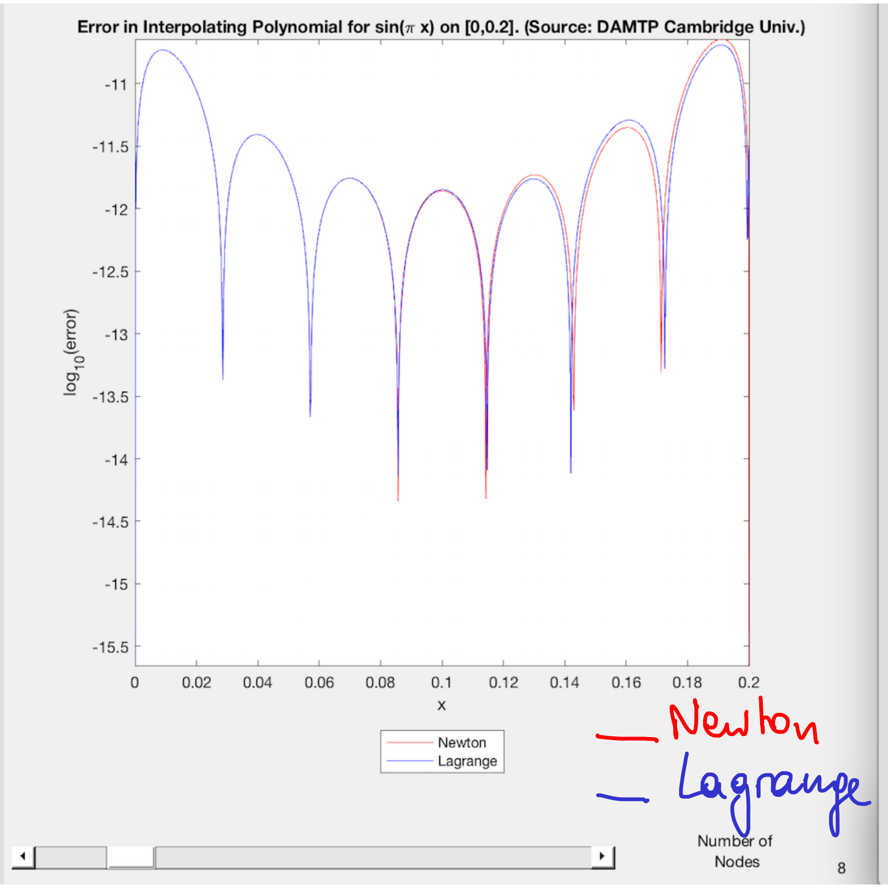
$$p_k(x_k) \quad : \quad \Theta(\underbrace{n^2}_{\text{assembling } A} + \underbrace{n^2 N}_{\text{solving}}) = \Theta(n^2 N)$$

Remark : Although the interpolating polynomial is algebraically unique, numerically there is a diff.

Example : True underlying function $\sin(\pi x)$ on $[0, 0.2]$

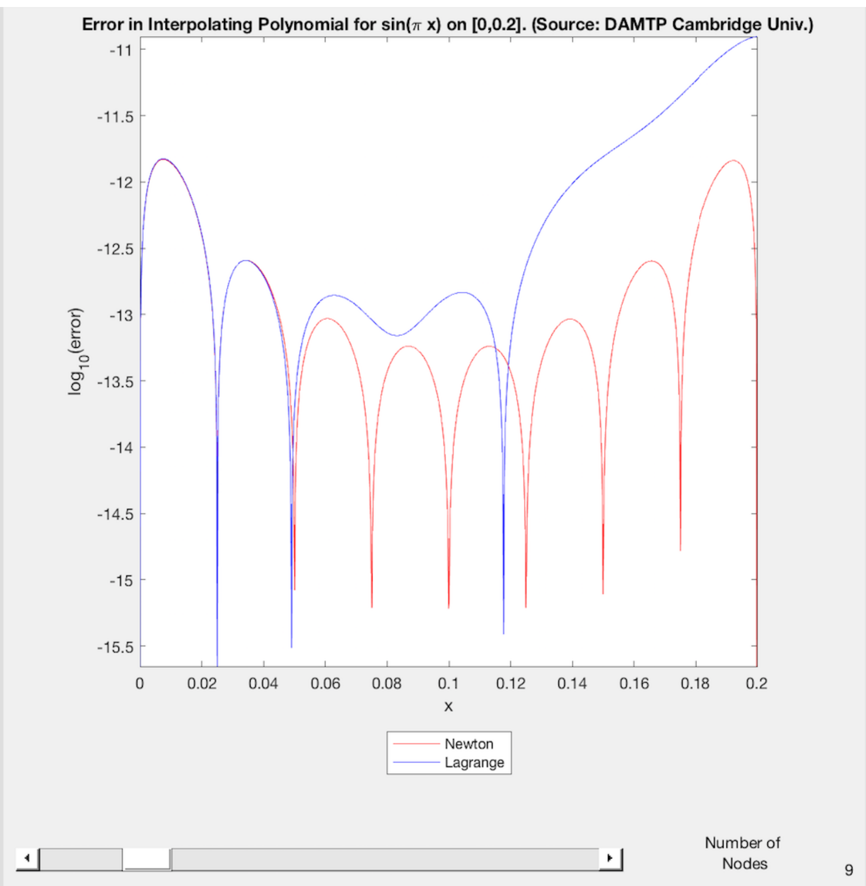
error of Lagrange vs. Newton for equidistant nodes

8 nodes

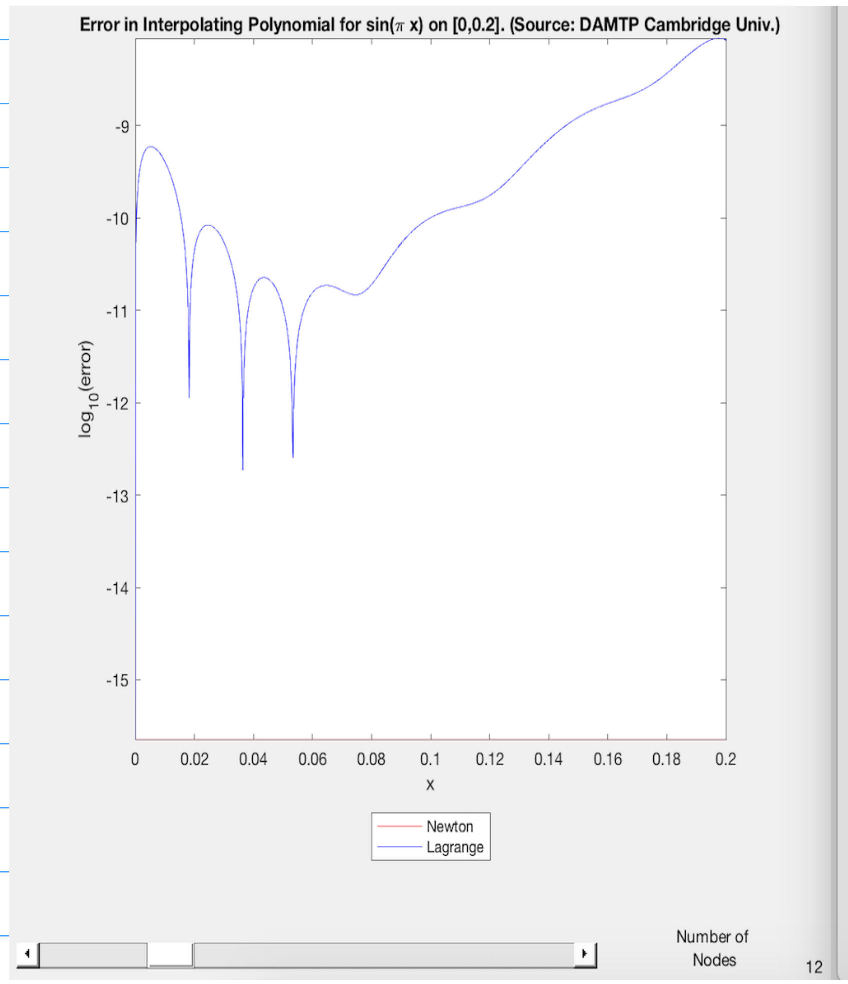


— Newton
— Lagrange

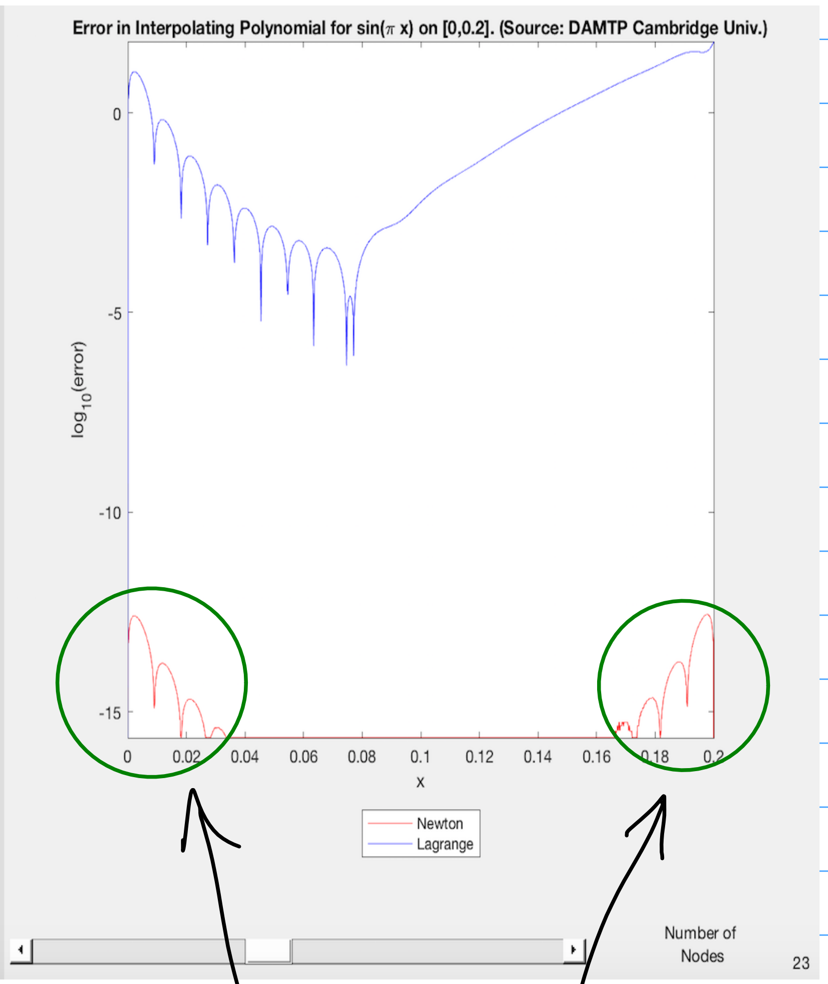
9 nodes



12 nodes



23 nodes



1.) Newton is more robust

2.) interpolating with higher degree polyn. $\not\Rightarrow$ better approximation

3.) Runge's phenomenon

Runge's phenomenon: interpolating with high degree polyn. & equidistant nodes $\rightsquigarrow$ oscillations at boundaries

Two remedies for Runge's phenomenon

1.) Different choice of nodes

$\rightsquigarrow$ Chebychev interpolation

2.) Piecewise polynomial interpolation

Approximation of functions by polynomial interpolation schemes

Given a function $f : I \subset \mathbb{R} \to \mathbb{K}$, $f \in C^0(I)$

How good is approximation by an interpolating polynomial?

Definition 6.1.2 (Lagrangian (interpolation polynomial) approximation scheme). Given an interval $I \subset \mathbb{R}$, $n \in \mathbb{N}$, a node set $\mathcal{T} = \{t_0, \ldots, t_n\} \subset I$, the *Lagrangian (interpolation polynomial) approximation scheme* $\mathsf{L}_\mathcal{T} : C^0(I) \to \mathcal{P}_n$ is defined by

$$\underbrace{\mathsf{L}_\mathcal{T}(f)}_{\text{approx. of } f} := \underbrace{I_\mathcal{T}(\mathbf{y})}_{\text{interpolating polynomial}} \in \mathcal{P}_n \quad \text{with} \quad \mathbf{y} := (f(t_0), \ldots, f(t_n))^T \in \mathbb{K}^{n+1}.$$

Goal: Understand $\|f - L_{\mathcal{T}}(f)\|_{L^\infty(I)}$?

$$\|g\|_{L^\infty(I)} = \sup_{x\in I} |g(x)|$$

$\mathcal{T}_n = \{ t_0^{(n)}, \dots, t_n^{(n)} \}$ node set with $n+1$ nodes

How does $\|f - L_{\mathcal{T}_n}(f)\|_{L^\infty(I)}$ behave in relation to n?

Equidistant nodes: $\mathcal{T}_n = \{ t_j^{(n)} := a + (b-a)\frac{j}{n}, j = 0,\dots,n \}$

$I = [a,b]$

Runge's Example: $f(t) = \dfrac{1}{1+t^2}$ $t \in \mathbb{R}$ $I = [-5,5]$

polyn. interp. with equidistant nodes

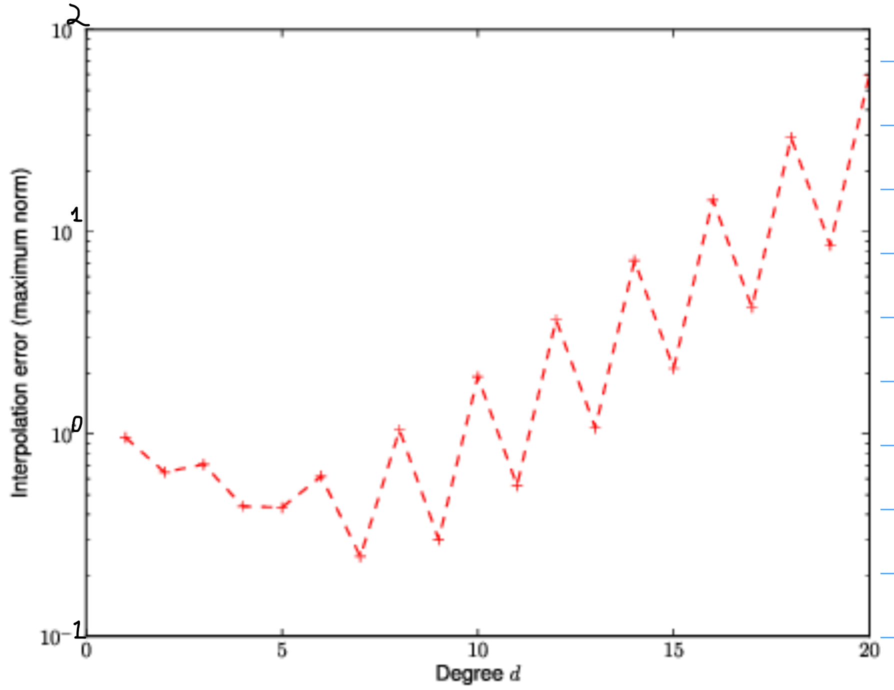


Fig. 214

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

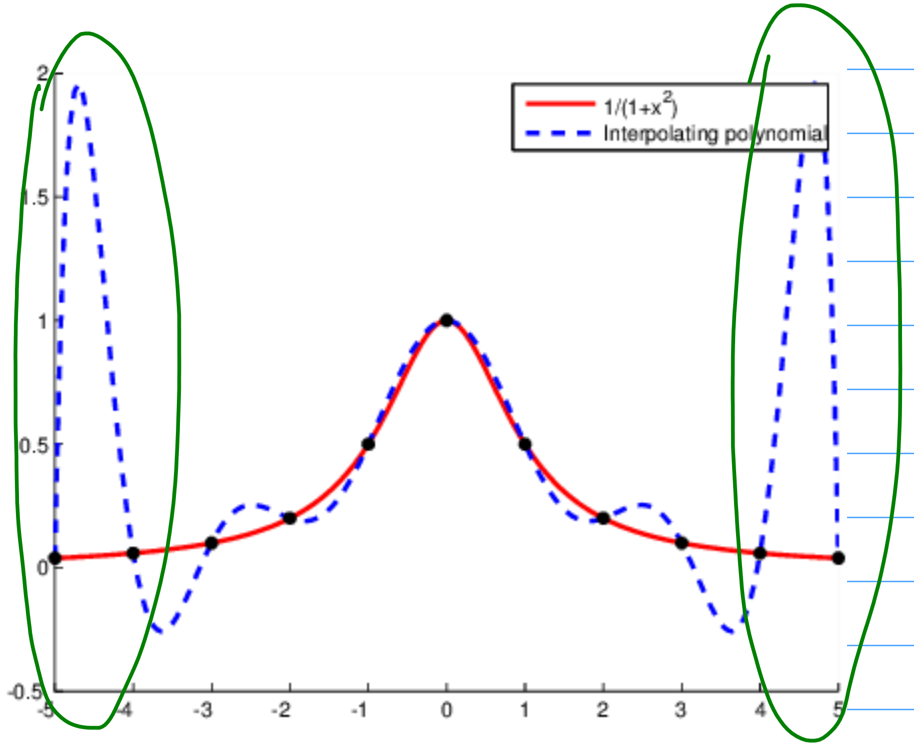


Fig. 213

Interpolating polynomial, $n = 10$

Towards understanding the interpolation error:

Theorem 6.1.4 (Representation of interpolation error). *We consider $f \in C^{n+1}(I)$ and the Lagrangian interpolation approximation scheme (see definition 6.1.2) 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*

$$f(t) - \mathsf{L}_{\mathcal{T}}(f)(t) = \underbrace{\frac{f^{(n+1)}(\tau_t)}{(n+1)!}}_{=c} \cdot \underbrace{\prod_{j=0}^{n}(t - t_j)}_{=:w(t)} . \qquad (6.17)$$

c depends on function only

$\omega(t)$ depends on nodes only

$\omega(t)$: "nodal polynomial"

Proof: Fix $t \in I \setminus \mathcal{T}$ (t is not a node)

We can choose $c \in \mathbb{R}$

$$f(t) - L_{\mathcal{T}} f(t) - c\,\omega(t) = 0$$

fixed t

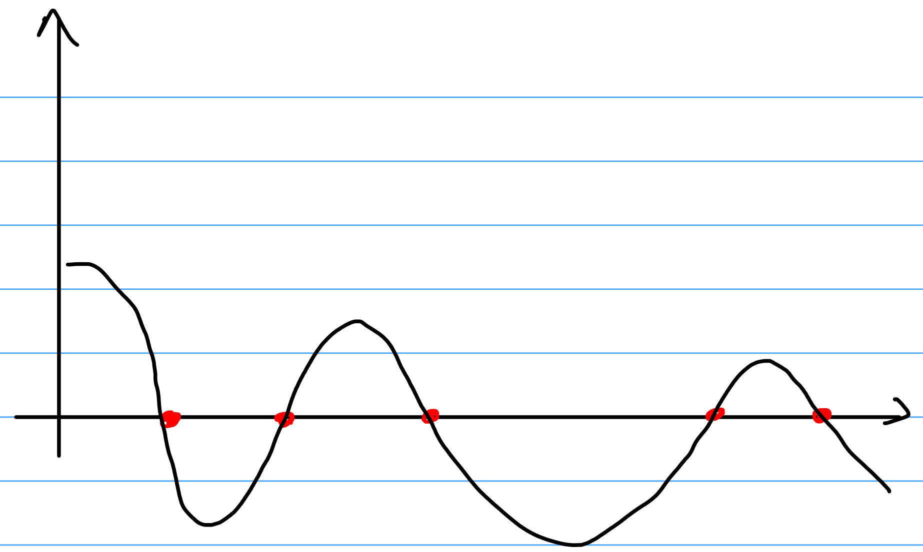
Define a function

$x^{n+1} + x^n \cdot \alpha_n + \dots$

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

How many zeros does φ have at least?

$$\left.\begin{array}{ll} \varphi(t) = 0 & (\text{by def.}) \\ \varphi(t_j) = 0 & j = 0, \dots, n \end{array}\right\} \begin{array}{l} n+2 \text{ distinct} \\ \text{zeros} \end{array}$$



φ' has at least $n+1$ distinct zeros (mean value thm)

φ'' n zeros

$\vdots$

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

$\rightarrow$ call it τ_t ($\Leftarrow$ depends on our fixed t)

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

Plug in τ_t:

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

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

Global estimate:

$$\|f - \mathsf{L}_{\mathcal{T}} f\|_{L^\infty(I)} \leq \frac{\left\|f^{(n+1)}\right\|_{L^\infty(I)}}{(n+1)!} \overbrace{\max_{t\in I} |(t-t_0)\cdot \dots \cdot (t-t_n)|}^{\|\omega\|_{L^\infty(I)}} . \qquad (6.18)$$

$$\left[\; f(t) - L_{\mathcal{T}} f(t) \underset{\substack{\uparrow \\ \text{Thm } 6.1.4}}{=} \frac{f^{(n+1)}(\tau_t)}{(n+1)!}\, \omega(t) \; \right]$$

Example of $\sin(t)$ on $[0,\pi]$: (equidistant nodes)

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

$$\|f - L_{\mathcal{J}} f\|_{L^\infty(I)} \leq \frac{1}{(n+1)!} \underbrace{\max_{t\in[0,\pi]} \left| t\cdot\left(t-\frac{\pi}{n}\right)\left(t-\frac{2\pi}{n}\right)\cdot\ldots\cdot(t-\pi)\right|}_{\text{extremal at } \approx \frac{\pi}{2n}}$$

$$\leq \frac{1}{(n+1)!} \left| \frac{\pi}{2n}\left(\frac{\pi}{2n}-\frac{\pi}{n}\right)\left(\frac{\pi}{2n}-\frac{2\pi}{n}\right)\cdot\ldots\cdot\left(\frac{\pi}{2n}-\pi\right)\right|$$

$$\leq \frac{1}{(n+1)!}\left(\frac{\pi}{n}\right)^{n+1} \cdot \underbrace{\left| \frac{1}{2}\cdot\left(\frac{1}{2}-1\right)\overbrace{\left(\frac{1}{2}-2\right)}^{|\,|\leq 2}\cdot\ldots\cdot\overbrace{\left(\frac{1}{2}-n\right)}^{|\,|\leq n}\right|}_{\leq n!}$$

$$\leq \frac{1}{n+1}\left(\frac{\pi}{n}\right)^{n+1}$$

converges (faster than exp.)

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

(equidistant nodes)

$$\|f^{(n+1)}\|_{L^\infty(I)} \sim 2^{n+1}(n+1)!$$

$$(6.18):\ \text{RHS} \sim \frac{2^{n+1}(n+1)!}{(n+1)!}\, n! \left(\frac{5}{n}\right)^{n+1} 2^n$$

$$= n!\,\frac{10}{n}\left(\frac{20}{n}\right)^n$$

$$\geq c\cdot 20^n$$

↑ Stirling's formula

exponential growth of RHS

$\Downarrow$

possible divergence of interpolation error

Chebychev interpolation

Recall : RHS of (6.18) depended on

$$\|f^{(n+1)}\|_{L^\infty(I)} \quad \text{and}$$

not in our control

$$\max_{t\in I} |\omega(t)|$$

some freedom here → by choosing the nodes

$$\omega(t) := \prod_{j=0}^{n} (t-t_j)$$

Task : Given n and interval I, where to place the nodes $t_0, \dots, t_n$ so that $\max_{t\in I} |\omega(t)|$ is as small as possible?

$$q := \underset{\omega \in \mathcal{P}_n}{\operatorname{argmin}} \left\{ \|\omega\|_{L^\infty(I)} \right\}$$

Properties of q?

- $q(t) = \prod_{j=0}^{n} (t-t_j)$ has $n+1$ zeros: $t_0 < \dots < t_n$
- All these zeros are in $I = [-1,1]$

(1) Suppose $t_0 < -1$

Then, we can define

$$p(t) = (t+1)(t-t_1)(t-t_2)\cdot \dots \cdot (t-t_n)$$

$$|p(t)| = |q(t)| \cdot \underbrace{\frac{|t+1|}{|t-t_0|}}_{<1}$$

$$\Rightarrow \quad |p(t)| < |q(t)| \quad \forall\, t \in I$$

$$\Rightarrow \quad \|p\|_{L^\infty} < \|q\|_{L^\infty} \quad ↯ \text{ opt. choice of } q.$$

② Suppose $t_n > 1$:

Define

$$p(t) := (t-t_0)(t-t_1)\cdot\ldots\cdot(t-1)$$

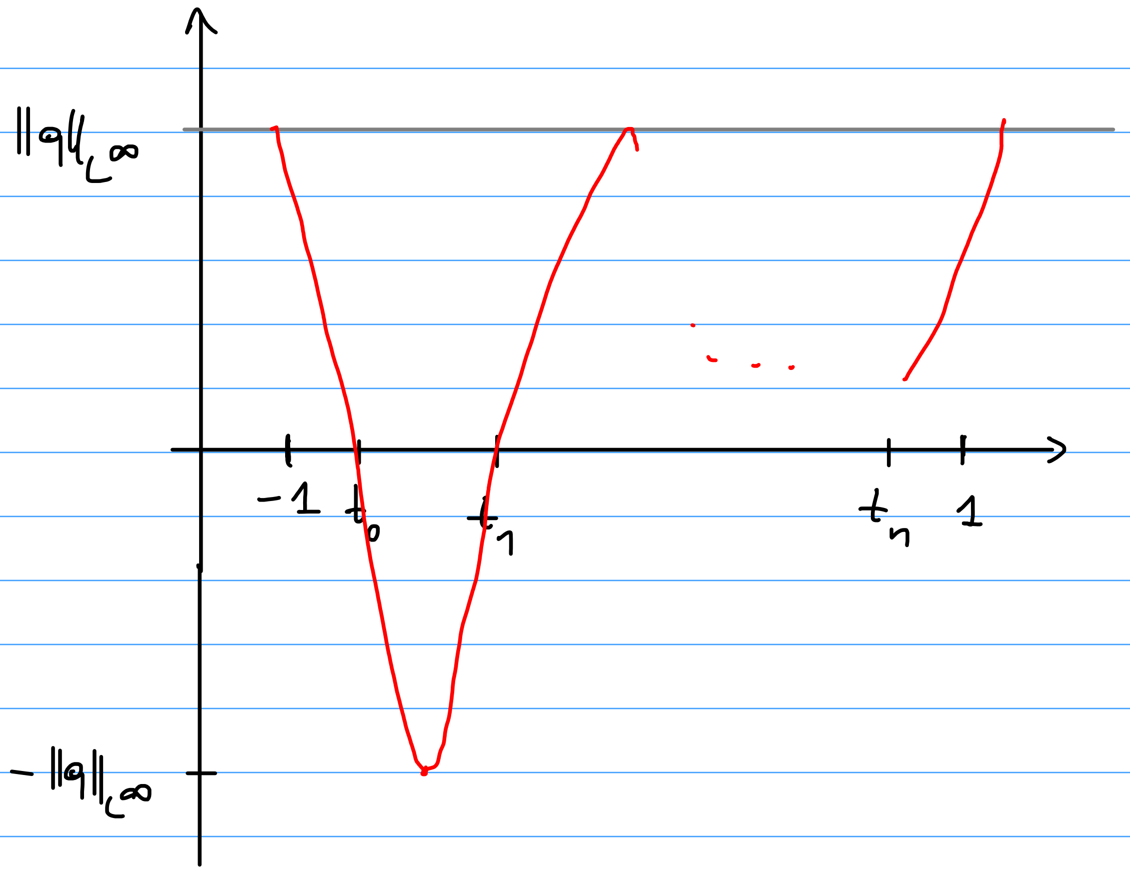
$$|p(t)| = |q(t)| \cdot \underbrace{\frac{|t-1|}{|t-t_n|}}_{<1}$$

again: contradiction to opt. choice of q.

Plan: Once we've found a polynomial q s.t. $\|q\|_{L^\infty}$ minimal

$\rightarrow$ we will choose

$t_0, \ldots, t_n$ as the zeros of q.



Answer to finding minimizing q:

Chebychev polynomials

The n-th Chebychev polynomial: $\boxed{T_n(t) := \cos(n \cdot \arccos t) \quad t \in [-1,1] \quad n \in \mathbb{N}}$

Chebychev polynomials satisfy a recurrence relation:

$$T_0 \equiv 1 \;, \; T_1(t) = t$$

$$T_{n+1}(t) = 2t\,T_n(t) - T_{n-1}(t)$$

By induction: $T_2(t) = 2t^2 - 1$

If $T_{n-1} \in \mathcal{P}_{n-1}$, $T_n \in \mathcal{P}_n \Rightarrow T_{n+1} \in \mathcal{P}_{n+1}$

$(T_0 \in \mathcal{P}_0, \; T_1 \in \mathcal{P}_1)$

$$T_3(t) = 4t^3 - 2t - t$$

What is the leading coefficient of T_n?

$$\boxed{2^{n-1}}$$

Zeros of $T_n(t)$

$$n \arccos(t_j) = \frac{2j+1}{2}\pi \qquad j = 0, \dots, n$$

$$\arccos(t_j) = \frac{2j+1}{2n}\pi$$

$$t_j = \cos\Big(\underbrace{\frac{2j+1}{2n}\pi}_{\text{equidistant}}\Big) \qquad \text{"Chebychev nodes"}$$

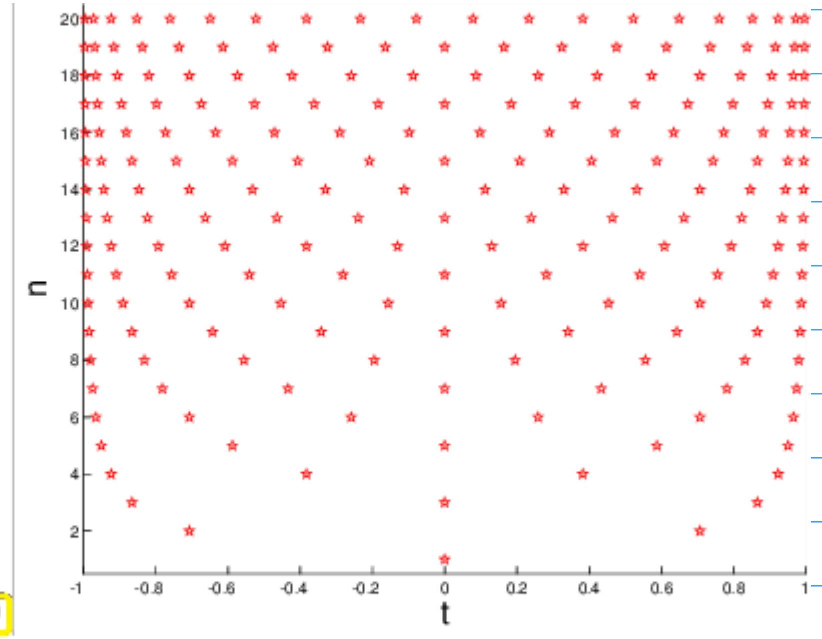


Fig. 219

nodes more densely distributed at boundaries

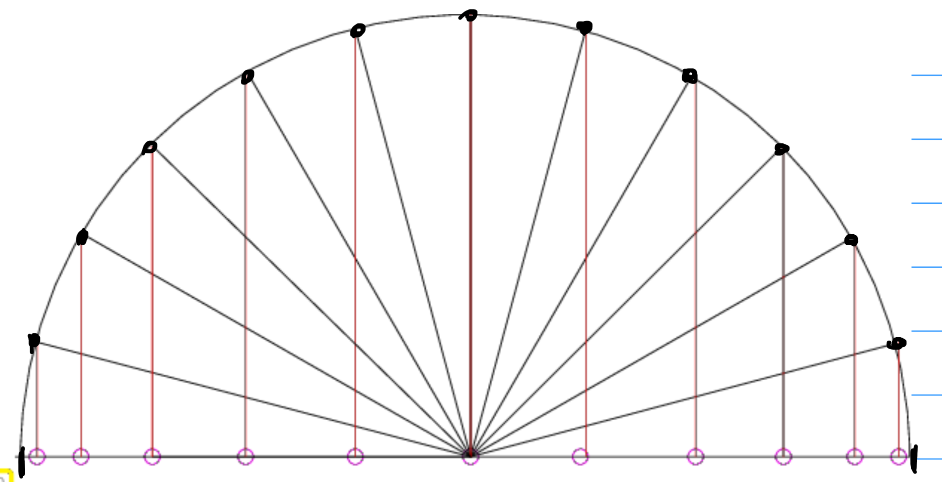


Fig. 220

-1 $\quad$ 1

equidistant on a circle

$\rightsquigarrow$ more dense at edges of interval

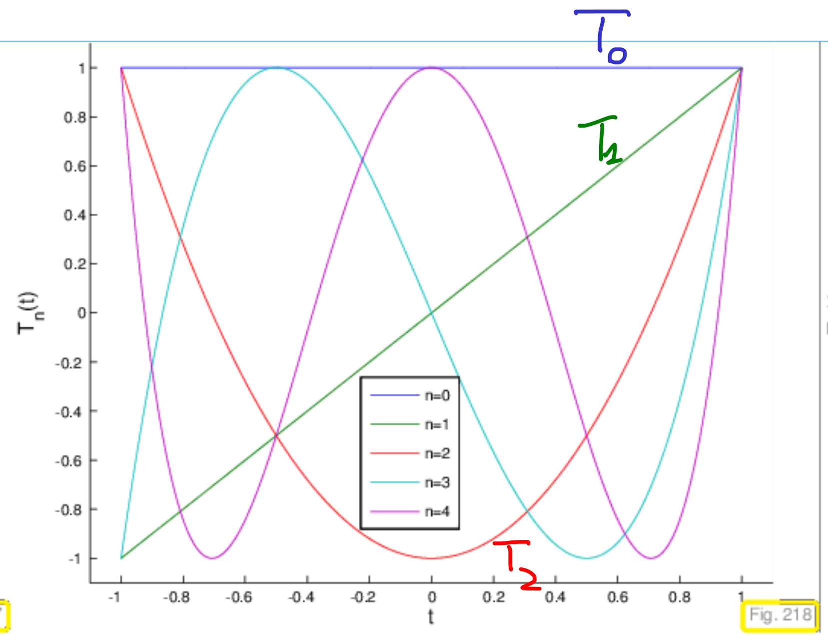


Fig. 217

Chebychev polynomials $T_0, \dots, T_4$

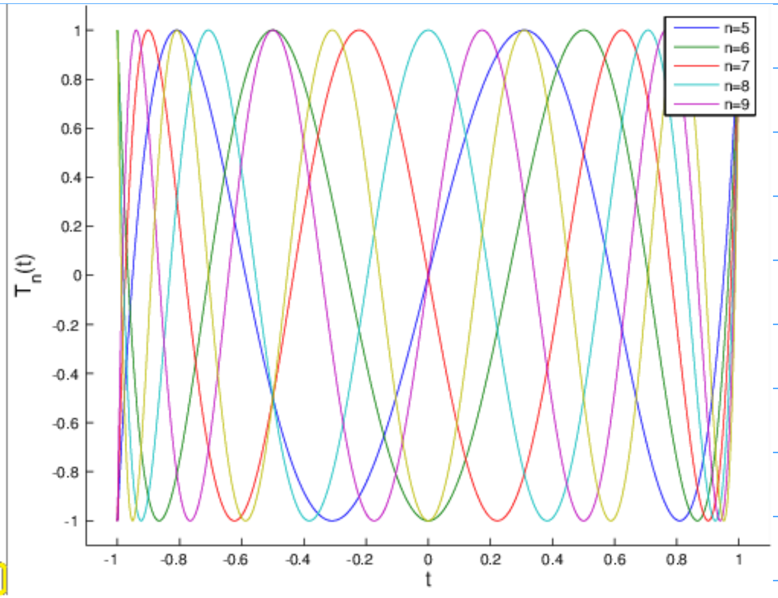


Fig. 218

Chebychev polynomials $T_5, \dots, T_9$

Is this choice optimal?

Theorem [Minimax property of Chebychev polynomials]

$$\|T_n\|_{L^\infty(I)} = \inf\left\{ \|p\|_{L^\infty([-1,1])} : p \in \mathcal{P}_n,\ p(t) = 2^{n-1} t^n + \dots \right\}$$

$\forall\, n \in \mathbb{N}$

Choice: $\quad q_n(t) = 2^{-n} T_{n+1}(t)$

$$q_n(t) := \prod_{j=0}^{n} (t - t_j) \in \mathcal{P}_{n+1}$$

$T_{n+1}(t)$ has leading coeff. 2^n

$\Rightarrow q_n(t)$ — " — 1

$\|T_n\|_{L^\infty([-1,1])} = 1 \quad (T_n(t) = \cos(n \arccos(t)))$

$\|q\|_{L^\infty([-1,1])} = 2^{-n}$

If $f \in C^{n+1}([-1,1])$

$$\|f - L_\mathcal{J} f\|_{L^\infty([-1,1])} \leq \boxed{\frac{2^{-n}}{(n+1)!}} \|f^{(n+1)}\|_{L^\infty([-1,1])}$$

Chebychev nodes instead of equidistant nodes

On arbitrary intervals $[a,b]$:

$$[-1,1] \to [a,b]$$

$$\hat{t} \mapsto a + \frac{b-a}{2}(\hat{t}+1) \qquad \hat{t} \in [-1,1]$$

(affine lin. trafo)

Chebychev nodes on $I = [a,b]$:

$$t_j := a + \frac{1}{2}(b-a)\left(\cos\left(\frac{2j+1}{2(n+1)}\pi\right) + 1\right)$$

Interpolation error estimate on $I = [a,b]$:

$$\|f - I_\mathcal{J} f\|_{L^\infty(I)} \leq \frac{2^{-2n-1}}{(n+1)!} |I|^{n+1} \|f^{(n+1)}\|_{L^\infty(I)} \qquad (*)$$

Still: possible divergence if $\|f^{(n)}\|_{L^\infty}$ grows too fast

(but has to be much worse than for equidistant nodes)

Example : $\frac{1}{1+t^2}$ $\quad t \in [-5,5]$

$$\|f^{(n+1)}\|_{L^\infty([-5,5])} \sim 2^{n+1}(n+1)!$$

RHS of $(*)$: $\frac{2^{-2n-1}}{(n+1)!} 10^{n+1} 2^{n+1}(n+1)!$

$= 5^n \cdot 10$ (still exp. growth)

BUT : does not necessarily mean divergence