Numerical Methods for Computational Science and Engineering

Fall Semester 2017 (HS17)

Prof. Rima Alaifari, SAM, ETH Zurich

5.5 Splines

Piecewise polynomial interpolation

- on each subinterval $[t_{i-1}, t_i]$ polynomial of degree d
- matching of first $d-1$ derivatives at nodes t_i

Definition 5.5.1. Spline space → [?, Def. 8.1]

Given an interval $I := [a,b] \subset \mathbb{R}$ and a knot set/mesh $\mathcal{M} := \{a = t_0 < t_1 < \ldots < t_{n-1} < t_n = b\}$, the vector space $\mathcal{S}_{d,\mathcal{M}}$ of the spline functions of degree d (or order $d+1$) is defined by

$$\mathcal{S}_{d,\mathcal{M}} := \{s \in C^{d-1}(I) \colon s_j := s_{|[t_{j-1},t_j]} \in \mathcal{P}_d \;\forall j = 1,\ldots,n\}\,.$$

$d-1$-times continuously differentiable

locally polynomial of degree d

Relationship between consecutive spline spaces through differentiation & integration:

$$s \in \mathcal{S}_{d,\mathcal{M}} \;\Rightarrow\; s' \in \mathcal{S}_{d-1,\mathcal{M}} \;\wedge\; \int_a^t s(\tau)\,\mathrm{d}\tau \in \mathcal{S}_{d+1,\mathcal{M}}\,.$$

The first 3 spline spaces:

Spline spaces of the lowest degrees:

- $d = 0$: $\mathcal{M}$-piecewise constant *discontinuous* functions
- $d = 1$: $\mathcal{M}$-piecewise linear *continuous* functions
- $d = 2$: *continuously differentiable* $\mathcal{M}$-piecewise quadratic functions

Dimension of $\mathcal{S}_{d,\mathcal{M}}$?

\# of intervals: n

degrees of freedom on each interval: $d+1$

at every interior point: d constraints

$\Rightarrow \dim \mathcal{S}_{d,\mathcal{M}} = n(d+1) - (n-1)d = n+d$.

5.5.1 Cubic spline interpolation

$$\mathcal{S}_{3,\mathcal{M}} = \{ s \in C^2(I) : s_j := s|_{[t_{j-1},t_j]} \in \mathcal{P}_3 \quad \forall j = 1,\dots,n \}$$

$$\uparrow$$

$$s_j'(t_j) = s_{j+1}'(t_j)$$

$$s_j''(t_j) = s_{j+1}''(t_j)$$

$$s_j = s|_{[t_{j-1},t_j]} \in \mathcal{P}_3 \qquad \forall j = 1,\dots,n$$

$$s_j(t) = a_j + b_j t + c_j t^2 + d_j t^3$$

$\rightsquigarrow 4n$ coefficients to determine

① Interpolating conditions

$$s_j(t_{j-1}) = y_{j-1}, \quad s_j(t_j) = y_j$$

$$\Rightarrow \left.\begin{aligned} a_j + b_j t_{j-1} + c_j t_{j-1}^2 + d_j t_{j-1}^3 &= y_{j-1} \\ a_j + b_j t_j + c_j t_j^2 + d_j t_j^3 &= y_j \end{aligned}\right\} \underline{2n \text{ conditions}}$$

② Smoothness conditions I:

$$s_j'(t_j) = s_{j+1}'(t_j) \qquad j = 1,\dots,n-1$$

$$b_j + 2c_j t_j + 3d_j t_j^2 = b_{j+1} + 2c_{j+1} t_j + 3d_{j+1} t_j^2$$

$\rightsquigarrow$ <u>$n-1$ conditions</u>

(3) Smoothness conditions II:

$$s_j''(t_j) = s_{j+1}''(t_j) \qquad j = 1, \dots, n-1$$

$$2c_j + 6d_j t_j = 2c_{j+1} + 6d_{j+1} t_j$$

$\rightsquigarrow$ <u>$n-1$ conditions</u>

$\Rightarrow$ overall: $2n + 2(n-1) = 4n-2$ conditions

to determine $4n$ coefficients

$\rightarrow$ Need 2 more constraints:

(4) E.g. natural/simple BCs:

$$s_1''(t_0) = 0, \quad s_n''(t_n) = 0 .$$

Altogether: LSE for the coefficients

$$\{a_j, b_j, c_j, d_j\}_{j=1,\dots,n} .$$

<u>Economical implementation of spline interpolation:</u>

$\forall\, j = 1, \dots, n$

$$s_j(t) = a_j + b_j(t - t_{j-1}) + c_j(t - t_{j-1})^2 + d_j(t - t_{j-1})^3$$

with $s_j(t_j) = y_j = s_{j+1}(t_j)$ (cf. (1))

$s_j''(t_j) = \sigma_j = s_{j+1}''(t_j)$ $\quad j = 1, \dots, n-1$ (cf. (3))

unknown

$\rightarrow \quad a_j = y_{j-1} \qquad\qquad h_j := t_j - t_{j-1}$

$$b_j = \frac{y_j - y_{j-1}}{h_j} - \frac{h_j(2\sigma_{j-1} + \sigma_j)}{6} \qquad (*)$$

$$c_j = \frac{\sigma_{j-1}}{2}$$

$$d_j = \frac{\sigma_j - \sigma_{j-1}}{6h_j}$$

Check: $\quad s_j(t_{j-1}) = a_j = y_{j-1}$ ✓

$$s_j(t_j) = a_j + b_j(t_j - t_{j-1}) + c_j(t_j - t_{j-1})^2 + d_j(t_j - t_{j-1})^3 \overset{(*)}{=} y_j .$$

$$s_j''(t_{j-1}) = 2c_j = \sigma_{j-1}$$

$$s_j''(t_j) = 2c_j + 6d_j \underbrace{(t_j - t_{j-1})}_{h_j} = \sigma_j .$$

What remains: matching first derivatives at nodes:

$$s_j'(t_j) = b_j + 2c_j h_j + 3d_j h_j^2$$

$$s_{j+1}'(t_j) = b_{j+1}$$

$$b_j + 2c_j h_j + 3d_j h_j^2 = b_{j+1}$$

$$\frac{y_j - y_{j-1}}{h_j} - \frac{h_j(2\sigma_{j-1} + \sigma_j)}{6} + \sigma_{j-1} h_j + \cancel{3}\,\frac{\sigma_j - \sigma_{j-1}}{\underset{2}{\cancel{6}}\cancel{h_j}}\, h_j^{\cancel{2}} = b_{j+1}$$

$$\sigma_{j-1}\left(-\frac{h_j}{3} - \frac{h_j}{2} + h_j\right) + \sigma_j\left(-\frac{h_j}{6} + \frac{h_j}{2} - \frac{h_{j+1}}{3}\right) + \sigma_{j+1}\frac{h_{j+1}}{6} = \underbrace{\frac{y_{j+1} - y_j}{h_{j+1}} - \frac{y_j - y_{j-1}}{h_j}}_{=: r_j}$$

$$\sigma_{j-1}\frac{h_j}{6}+\sigma_j\frac{h_j+h_{j+1}}{3}+\sigma_{j+1}\frac{h_{j+1}}{6}=r_j \qquad j=1,\dots,n-1$$

where $\sigma_0 = 0 = \sigma_n$

Tridiagonal system for $[\sigma_1,\dots,\sigma_{n-1}]^T$:

$$\begin{bmatrix} \frac{h_1+h_2}{3} & \frac{h_2}{6} & 0 & \cdots & 0 \\ \frac{h_2}{6} & \frac{h_2+h_3}{3} & \frac{h_3}{6} & & 0 \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \frac{h_{n-1}}{6} \\ 0 & & & \frac{h_{n-1}}{6} & \frac{h_{n-1}+h_n}{3} \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \vdots \\ \vdots \\ \sigma_{n-1} \end{bmatrix} = \begin{bmatrix} r_1 \\ \vdots \\ \vdots \\ r_{n-1} \end{bmatrix}$$

$(n-1)\times(n-1)$ system to solve for $[\sigma_1,\dots,\sigma_{n-1}]^T$

$\rightarrow$ coefficients $\{a_j, b_j, c_j, d_j\}_{j=1,\dots,n}$ determined by $(*)$

6. Approximation of Functions in 1D

Task: Given a function f, find a "simple" approximation $\tilde{f}$

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. 5.1.6)

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"

simple: $\tilde{f}$
- encoded by small amount of information
- easy to evaluate

e.g. polynomial or piecewise polynomial

small error: $\|f-\tilde{f}\|$ small, where $\|\cdot\|$ some norm on $C^0(\bar{D})$, e.g. $\|\cdot\|_2$, $\|\cdot\|_\infty$

$$\|f-\tilde{f}\|_2^2 = \int_D |f-\tilde{f}|^2(x)\,dx$$

$$\|f-\tilde{f}\|_\infty = \max_{x\in\bar{D}} |f(x)-\tilde{f}(x)|$$

Applications: model reduction in dynamic systems & control systems

For example: Dimension reduction in weather prediction model

Throughout this chapter: $n=d=1$:

$$f: D\subset\mathbb{R} \to \mathbb{R}$$

Interpolation scheme + sampling $\to$ approximation scheme

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

free choice of nodes $t_i \in I$

6.1 Approximation by Global Polynomials

Any function $f \in C^k(I)$ can be approximated by Taylor polynomial:

Given $t_0 \in I$, there exists a function $h_k : \mathbb{R} \to \mathbb{R}$ s.t.

$$f(t) = \underbrace{\sum_{j=0}^{k} \frac{f^{(j)}(t_0)}{j!} (t-t_0)^j}_{=: T_k(t)} + h_k(t) \cdot (t-t_0)^k$$

and $h_k(t) \xrightarrow[t \to t_0]{} 0$.

(here $f^{(j)}$ denotes j-th derivative of f)

→ T_k approximates f in a (possibly small) neighborhood $J \subset I$ of t_0.

If $f \in C^{k+1}(I)$: can quantify the error:

$$f(t) - T_k(t) = \frac{f^{(k+1)}(\xi)}{(k+1)!} (t-t_0)^{k+1}$$

for some point $\xi \in (\min(t, t_0), \max(t, t_0))$

Taylor approximation:

⊕ easy, direct

⊖ inefficient [same accuracy often reached with lower degree polynomials]

access to higher order derivatives required

⇝ can be difficult to obtain

6.1.1. Polynomial Approximation: Theory

Taylor polynomials yield <u>local</u> approximation of sufficiently <u>smooth</u> functions

uniform approximation on I without smoothness requirements (functions are merely continuous)?

YES (recall Weierstrass Approximation Theorem)

Theorem 6.1.6. Uniform approximation by polynomials

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

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

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 $0 \le k \le m$.

uniform approximation in $\|\cdot\|_\infty = \|\cdot\|_{L^\infty([0,1])}$

Bernstein polynomials

$$B_j^n(t) = \binom{n}{j} t^j (1-t)^{n-j}, \quad p_n \in \mathcal{P}_n. \tag{6.1.8}$$

$$\Rightarrow \quad p_n(t) = \sum_{j=0}^{n} f(j/n)\, B_j^n(t)$$

B_j^n satisfy

$$\sum_{j=0}^{n} B_j^n(t) \equiv 1, \tag{6.1.9}$$

$$0 \le B_j^n(t) \le 1 \quad \forall 0 \le t \le 1. \tag{6.1.10}$$

B_j^n for $n = 7$:

$j = 1, \dots, 7$

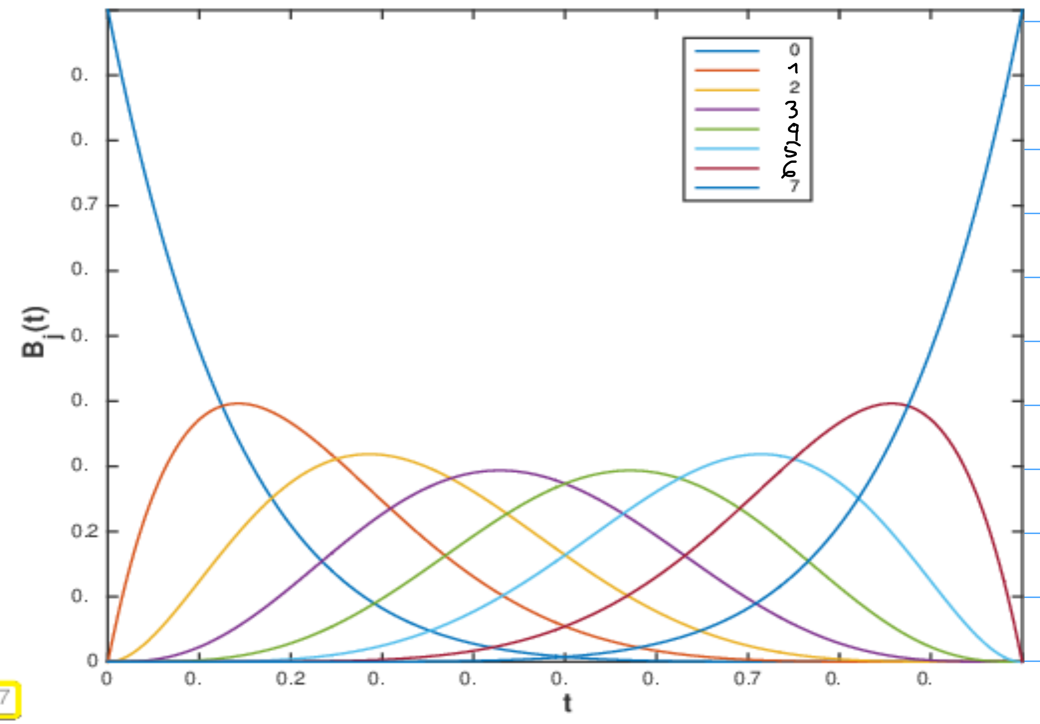

Fig. 207