Cubic B-Splines

Recall from poly interpolation, we find:

- $p(x)$ st $p(x_i) = f_i$
- $p(x)$ is unique, but can have diff. bases
  - power series basis

    $\{1, x, x^2, \dots x^n\}$ $\quad p(x) = \sum_{i=1}^{n} a_i x^i$
  - Newton basis

    $\{1, (x-x_0), (x-x_0)(x-x_1), \dots (x-x_0)(x-x_1)\cdots(x-x_{n-1})\}$

    $p(x) = \sum_{i=0}^{n} \left[b_i \prod_{j=0}^{i-1} (x-x_j)\right]$
  - Lagrange basis:

    $\{L_{n,0}(x), L_{n,1}(x), \dots L_{n,n}(x)\}$ $\quad p(x) = \sum_{i=1}^{n} f_i L_{n,i}(x)$

Q: Can we find basis for cubic splines?

i.e. instead of writing

$$S_{3,n}(x) = \begin{cases} p_1(x) \\ p_2(x) \\ \vdots \\ p_n(x) \end{cases}$$

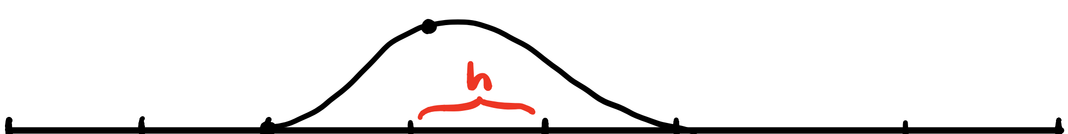
we want to find basis function

$\{B_{-1}(x), B_0(x), \dots B_n(x), B_{n+1}(x)\}$

such that $\underbrace{S_{3,n}(x) = \sum_{i=-1}^{n+1} a_i B_i(x)}_{\text{called B-spline}}$.

Basic Idea:

- use equally-spaced points



x_{-1} x_0 x_1 x_2 x_3 x_4 x_n x_{n+1}

← include two "new points" on each end →

- define cubic polynomials on

  $[x_{i-2}, x_{i+2}] = [x_i - 2h, x_i + 2h]$ (consider $x_i = x_3$)

  with properties:

  $B_i(x) = 0$ for $x \notin [x_i - 2h, x_i + 2h]$

  $B_i(x)$ is cubic spline interpolating

  $(x_i - 2h, 0)$, $(x_i, 1)$, $(x_i + 2h, 0)$

  After some algebra, we get:

$$
B_i(x) = \begin{cases}
\frac{1}{4h^3}(x - x_{i-2})^3 & x_{i-2} \le x < x_{i-1} \\
\frac{1}{4h^3}(x - x_{i-2})^3 - \frac{1}{h^3}(x - x_{i-1})^3 & x_{i-1} \le x < x_i \\
\frac{-1}{4h^3}(x - x_{i-2})^3 + \frac{1}{h^3}(x - x_{i-1})^3 & x_i \le x < x_{i+1} \\
\frac{-1}{4h^3}(x - x_{i-2})^3 & x_{i+1} \le x < x_{i+2} \\
0 & \text{else}
\end{cases}
$$

- Then, we find a_i such that

$$\sum_{i-1}^{i+1} a_i B_i(x_j) = f_j, \quad j = 0, 1, \dots n$$

Note B_i, $i = -1, 0, 1, \dots n+1$ are the set of all B-spline basis which are nonzero on $[x_i - 2h, x_i + 2h]$

Observe: $\vdots$

$$
\begin{aligned}
&B_{j-2}(x_j) = 0 \\
&B_{j-1}(x_j) = 1/4 \\
&B_j(x_j) = 1 \\
&B_{j+1}(x_j) = 1/4 \\
&B_{j+2}(x_j) = 0 \\
&\quad\vdots
\end{aligned}
\quad \Rightarrow \quad
\begin{aligned}
&a_{j-1} B_{j-1}(x_j) + a_j B_j(x_j) \\
&+ a_{j+1} B_{j+1}(x_j) = f_j \\
&j = 0, 1, \dots n
\end{aligned}
$$

Or $\frac{1}{4}a_{j-1} + a_j + \frac{1}{4}a_{j+1} = f_j, \quad j = 0, 1, \dots n$

This gives n+1 eqns, n+3 unknowns

$a_{-1}, a_0, \dots a_n, a_{n+1}$

→ need 2 more eqns.

In the case of natural end (free boundary) conditions:

$$S''(x_0) = S''(x_n) = 0$$

Note that $\frac{\partial}{\partial x}\left(\frac{1}{4h^3}(x - x_{i-2})^3\right) = \frac{3}{4h^3}(x - x_{i-2})^2$

$\frac{\partial^2}{\partial x^2}\left(\frac{1}{4h^3}(x - x_{i-2})^3\right) = \frac{3}{2h^3}(x - x_{i-2})$

$$\Rightarrow \begin{array}{ll} S''(x_0) = 0 & \frac{3}{2h^2}a_{-1} - \frac{3}{h^2}a_0 + \frac{3}{h^2}a_1 = 0 \\ S''(x_n) = 0 & \frac{3}{2h^2}a_{n-1} - \frac{3}{h^2}a_n + \frac{3}{h^2}a_{n+1} = 0 \end{array} \quad (*)$$

n+3 linear eqns can be solved using Gaussian Elim.

But we can simplify the eqns.

Adding first 2 eqns:

$$\frac{3}{2}a_{-1} - 3a_0 + \frac{3}{2}a_1 = 0$$
$$\frac{1}{4}a_{-1} + a_0 + \frac{1}{4}a_1 = f_0$$
$$\rightarrow a_0 = \frac{2}{3}f_0$$

Adding last 2 eqns: $a_n = \frac{2}{3}f_n$

So we have:

$$\begin{aligned} a_1 + \tfrac{1}{4}a_2 \qquad\quad &= f_1 - \tfrac{1}{6}f_0 \\ \tfrac{1}{4}a_1 + a_2 + \tfrac{1}{4}a_3 &= f_2 \\ \tfrac{1}{4}a_2 + a_3 + \tfrac{1}{4}a_4 &= f_3 \\ &\vdots \\ \tfrac{1}{4}a_{n-3} + a_{n-2} + \tfrac{1}{4}a_{n-1} &= f_{n-2} \\ \tfrac{1}{4}a_{n-2} + a_{n-1} \qquad\quad &= f_{n-1} - \tfrac{1}{6}f_n \end{aligned}$$

To find coeffs. a_i, solve:

$$\begin{bmatrix} 1 & 1/4 & \cdots & & \\ 1/4 & 1 & 1/4 & & \\ \vdots & 1/4 & 1 & 1/4 & \\ & & 1/4 & \ddots & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{n-2} \\ a_{n-1} \end{bmatrix} = \begin{bmatrix} f_1 - \frac{1}{6} f_0 \\ f_2 \\ \vdots \\ f_{n-2} \\ f_{n-1} - \frac{1}{6} f_n \end{bmatrix}$$

can be solved efficiently using e.g. Thomas Algorithm ($O(n)$ instead of $\underbrace{O(n^3)}_{\text{Gaussian Elim.}}$ FLOPs)

Finally, compute a_{-1} and a_{n+1} from $(*)$.