3.5 Cubic spline interpolation

Rationale: High-degree polynomials can oscillate erratically.

Piecewise-polynomial approximation is intended to remedy that.

#11. A natural cubic spline S on $[0,2]$ is defined by

$$S(x) = \begin{Bmatrix} S_0(x) = 1 + 2x - x^3, & if\ 0 \le x < 1 \\ S_1(x) = 2 + b(x-1) + c(x-1)^2 + d(x-1)^3, & if\ 1 \le x \le 2 \end{Bmatrix}.$$

Find $b, c, and\ d$.

Calculate the derivatives: $S_0{}'(x) = 2 - 3x^2$; $S_0{}''(x) = -6x$

$S_1{}'(x) = b + 2c(x-1) + 3d(x-1)^2$; $S_1{}''(x) = 2c + 6d(x-1)$

(1) $S_0(1) = S_1(1) \quad 2 = 2.$
(2) $S_0{}'(1) = S_1{}'(1) \quad -1 = b$
(3) $S_0{}''(1) = S_1{}''(1) \quad -6 = 2c \qquad c = -3$
(4) $S_0{}''(0) = 0$; $S_1{}''(2) = -6 + 6d$; $-6 + 6d = 0$; $d = 1.$

Definition (3.10)

Given a function f defined on $[a, b]$ and a set of nodes $a = x_0 < x_1 < \cdots < x_n = b$, a **cubic spline interpolant** S for f is a function that satisfies the following conditions:

(a) $S(x)$ is a cubic polynomial, denoted $S_j(x)$, on the subinterval $[x_j, x_{j+1}]$ for each $j = 0, 1, \ldots, n-1$;

(b) $S_j(x_j) = f(x_j)$ and $S_j(x_{j+1}) = f(x_{j+1})$ for each $j = 0, 1, \ldots, n-1$;

(c) $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$ for each $j = 0, 1, \ldots, n-2$; (*Implied by* **(b)**.)

(d) $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1})$ for each $j = 0, 1, \ldots, n-2$;

(e) $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$ for each $j = 0, 1, \ldots, n-2$;

(f) One of the following sets of boundary conditions is satisfied:

(i) $S''(x_0) = S''(x_n) = 0$ **(natural** *(or free)* **boundary)**;

(ii) $S'(x_0) = f'(x_0)$ and $S'(x_n) = f'(x_n)$ **(clamped boundary)**.

Construction of a cubic spline.

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3,$$

$$j = 0,1,\cdots,n-1$$

We will use the conditions (a-f) to find the coefficients.

$S_j(x_j) = a_j = f(x_j)$ (from (b)).

$a_{j+1} = S_{j+1}(x_{j+1}) = S_j(x_{j+1}) = a_j + b_j(x_{j+1} - x_j) + c_j(x_{j+1} - x_j)^2 + d_j(x_{j+1} - x_j)^3,$

$j = 0,1,\cdots,n-2.$

Define $h_j = x_{j+1} - x_j$, $j = 0,1,\cdots,n-1$. Also, define $a_n = f(x_n)$.

Then

$$a_{j+1} = a_j + b_jh_j + c_jh_j^2 + d_jh_j^3 \;;\; j = 0,1,\cdots,n-1 \qquad (3.15)$$

Define $b_n = S'(x_n)$. Differentiate $S_j(x)$:

$S_j'(x) = b_j + 2c_j(x - x_j) + 3d_j(x - x_j)^2.$

Then $S_j'(x_j) = b_j$; $j = 0,1,\cdots,n-1$

Apply (d): $b_{j+1} = b_j + 2c_jh_j + 3d_jh_j^2$; $j = 0,1,\cdots,n-1$ (3.16)

Define $c_n = \frac{S''(x_n)}{2}$. Differentiate $S_j'(x)$.

$S_j''(x) = 2c_j + 3 * 2 * d_j(x - x_j).$

Then, with (e),

$$S_{j+1}''(x_{j+1}) = 2c_{j+1} = S_j''(x_{j+1}) = 2c_j + 6d_jh_j.$$

Thus,

$c_{j+1} = c_j + 3d_j h_j \; ; j = 0,1,\cdots,n-1$ (3.17).

Solve (3.17) for d_j :

$d_j = \frac{c_{j+1}-c_j}{3h_j}$. Substitute this into (3.15) and (3.16):

$$a_{j+1} = a_j + b_j h_j + c_j {h_j}^2 + \frac{c_{j+1}-c_j}{3h_j} {h_j}^3 =$$

$$a_j + b_j h_j + c_j {h_j}^2 + \frac{c_{j+1}-c_j}{3} {h_j}^2 =$$

$a_j + b_j h_j + \frac{{h_j}^2}{3}\left(2c_j + c_{j+1}\right)$ (3.18)

From (3.16):

$b_{j+1} = b_j + h_j\left(c_j + c_{j+1}\right)$ (3.19)

Solve (3.18) for b_j :

$b_j = \frac{1}{h_j}\left(a_{j+1} - a_j\right) - \frac{h_j}{3}\left(2c_j + c_{j+1}\right)$ (3.20)

Here $j = 0,1,\cdots,n-1$. Equivalently,

$b_{j-1} = \frac{1}{h_{j-1}}\left(a_j - a_{j-1}\right) - \frac{h_{j-1}}{3}\left(2c_{j-1} + c_j\right), j = 1,\cdots,n$ (3.20a)

From (3.19),

$b_j = b_{j-1} + h_{j-1}\left(c_{j-1} + c_j\right), j = 1,\cdots,n.$

$$\frac{1}{h_j}\left(a_{j+1} - a_j\right) - \frac{h_j}{3}\left(2c_j + c_{j+1}\right)$$
$$= \frac{1}{h_{j-1}}\left(a_j - a_{j-1}\right) - \frac{h_{j-1}}{3}\left(2c_{j-1} + c_j\right) + h_{j-1}\left(c_{j-1} + c_j\right).$$

Using some algebra, one obtains

$$h_{j-1}c_{j-1} + 2(h_{j-1} + h_j)c_j + h_j c_{j+1} = \frac{3}{h_j}(a_{j+1} - a_j) - \frac{3}{h_{j-1}}(a_j - a_{j-1}) \qquad (3.21)$$

$$j = 1, \cdots, n-1$$

(3.21) is a system of linear equations in $\{c_j\}_{j=0}^{n}$ (n+1 unknowns)

The values of $a_j = f(x_j)$ and $h_j = x_{j+1} - x_j$ are known.

Once the system (3.21) has been solved and the values of $\{c_j\}_{j=0}^{n}$ have been found, use (3.20) and (3.17) to find all coefficients b_j and d_j.

In order to find a unique solution, one needs two additional equations.

Present the system (3.21) in the matrix form. Add boundary conditions first: $c_n = \frac{S''(x_n)}{2} = 0$ and $0 = S''(x_0) = 2c_0 + 6d(x_0 - x_0)$.

Thus, $c_0 = c_n = 0$.

Clamped splines.

The two additional equations are $S'(a) = f'(a)$ and $S'(b) = f'(b)$. We will show that these two equations are linear in c_j s.

$f'(a) = S'(a) = S'(x_0) = b_0$.

Eq. (3.20) with $j = 0$ implies

$f'(a) = \frac{a_1 - a_0}{h_0} - \frac{h_0}{3}(2c_0 + c_1)$.

Rearrange and multiply by 3:

$$2h_0c_0 + h_0c_1 = \frac{3}{h_0}(a_1 - a_0) - 3\ f'(a) \qquad (*)$$

Similarly, from (3.19) with $j = n - 1$

$$f'(b) = b_n = b_{n-1} + h_{n-1}(c_{n-1} + c_n).$$

Then, from (3.20) and (3.20a),

$$\begin{aligned} f'(b) &= b_n \\ &= \frac{1}{h_{n-1}}(a_n - a_{n-1}) - \frac{h_{n-1}}{3}(2c_{n-1} + c_n) \\ &\quad + h_{n-1}(c_{n-1} + c_n) \\ &= \frac{1}{h_{n-1}}(a_n - a_{n-1}) + \frac{h_{n-1}}{3}(c_{n-1} + 2c_n) \end{aligned}$$

This leads to

$$h_{n-1}c_{n-1} + 2h_{n-1}c_n = 3\ f'(b) - \frac{3}{h_{n-1}}(a_n - a_{n-1}) \qquad (**)$$

Eqs. (*) and (**) will be added to the system (3.21).

Natural splines in a matrix form. Introduce n+1) x (n+1) matrix A as

$$\begin{array}{cccccccc}
1 & 0 & 0 & * & * & * & & 0 \\
h_0 & 2(h_0 + h_1) & h_1 & * & * & & & * \\
0 & h_1 & 2(h_1 + h_2) & h_2 & * & * & & * \\
* & * & * & * & * & * & & * \\
* & * & * & * & & * & 0 & 0 \\
* & & * & & & h_{n-2} & 2(h_{n-2} + h_{n-1}) & h_{n-1} \\
0 & * & * & 0 & & 0 & 0 & 1
\end{array}$$

$$
g = \begin{matrix} 0 \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ * \\ * \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 0 \end{matrix}
$$

$$
x = \begin{matrix} c_0 \\ c_1 \\ * \\ * \\ c_n \end{matrix}
$$

~~Solve $Ag = x$.~~ The solution $x = A^{-1}g$.

#17. Construct a natural cubic spline to approximate $f(x) = \cos \pi x$ using its values at the nodes given in the left column.

Xi	Cos πx	a_i	b_i	c_i	d_i
0	1.0	1.0	-0.7574	0.0	-6.627
0.25	0.7071	0.7071	-2.0	-4.971	6.627
0.5	0	0	-3.243	0.0	6.627
0.75	-0.7071	-0.7071	-2.0	4.971	-6.627
1	-1.0	-1.0			

A=

1	0	0	0	0
0.25	1	0.25	0	0
0	0.25	1	0.25	0
0	0	0.25	1	0.25
0	0	0	0	1

$g =$

0
-4.971
0
4.971
0

$x = A^{-1}g =$

0
-4.971
0
4.971
0

(b) Integrate the spline over $[0,1]$ and compare the result to $\int_0^1 \cos \pi x \, dx = 0.$

$$\int_0^1 S(x)\, dx = 0.$$

Use the derivatives of the spline to approximate $f'(0.5)$ and $f''(0.5)$. Compare the approximations to the actual values.

$$f'(0.5) \approx S'(0.5) = -3.243$$

$$f'(0.5) = -\pi = -3.142$$

$$S''(0.5) = 2c_2 = 0.0$$

$f''(0.5) = 0.0.$

#34. Let f be defined on $[a, b]$ and let the nodes $a = x_0 < x_1 < x_2 = b$ be given. A quadratic interpolating spline S consists of the quadratic polynomial $S_0(x) = a_0 + b_0(x - x_0) + c_0(x - x_0)^2$ on $[x_0, x_1]$

and the quadratic polynomial

$S_1(x) = a_1 + b_1(x - x_1) + c_1(x - x_1)^2$ on $[x_1, x_2]$

Such that

(i) $S(x_0) = f(x_0), S(x_1) = f(x_1), S(x_2) = f(x_2)$

(ii) $S \in C^1[x_0, x_2]$.

Show that conditions (i) and (ii) lead to five equations in the six unknowns $a_0, b_0, c_0, a_1, b_1, and\ c_1$. What additional condition to impose to make the solution unique? Does the condition $S \in C^2[x_0, x_2]$ lead to a meaningful solution?

$a_0 = f(x_0);\ a_1 = f(x_1)$

$$S_1(x_2) = a_1 + b_1(x_2 - x_1) + c_1(x_2 - x_1)^2 = f(x_2)$$

$$S_0(x_1) = f(x_1);\ \ a_0 + b_0(x_1 - x_0) + c_0(x_1 - x_0)^2 = a_1$$

$S_0'(x_1) = S_0'(x_1);\ b_1 = b_0 + 2c_0(x_1 - x_0)$

$S \in C^1[x_0, x_2]$ implies $c_0 = c_1$. Then $S(x)$ is a quadratic function (solution is not a quadratic spline).

Theorem (3.11)

If f is defined at $a = x_0 < x_1 < \cdots < x_n = b$, then f has a unique natural spline interpolant S on the nodes $x_0, x_1, \ldots, x_n$; that is, a spline interpolant that satisfies the natural boundary conditions $S''(a) = 0$ and $S''(b) = 0$.

Algorithm 3.4: NATURAL CUBIC SPLINE

To construct the cubic spline interpolant S for the function f, defined at the numbers $x_0 < x_1 < \cdots < x_n$, satisfying $S''(x_0) = S''(x_n) = 0$:
INPUT n; $x_0, x_1, \ldots, x_n$; $a_0 = f(x_0), a_1 = f(x_1), \ldots, a_n = f(x_n)$.
OUTPUT a_j, b_j, c_j, d_j for $j = 0, 1, \ldots, n-1$.
(*Note:* $S(x) = S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$ *for* $x_j \le x \le x_{j+1}$.)

Step 1 For $i = 0, 1, \ldots, n-1$ set $h_i = x_{i+1} - x_i$.

Step 2 For $i = 1, 2, \ldots, n-1$ set $\alpha_i = \dfrac{3}{h_i}(a_{i+1} - a_i) - \dfrac{3}{h_{i-1}}(a_i - a_{i-1})$.

Step 3 Set $l_0 = 1$; $\mu_0 = 0$; $z_0 = 0$. (*Steps 3-5, & part of 6 solve tridiagonal linear system by method in Algorithm 6.7.*)

Step 4 For $i = 1, 2, \ldots, n-1$
set $l_i = 2(x_{i+1} - x_{i-1}) - h_{i-1}\mu_{i-1}$;
$\mu_i = h_i / l_i$; $z_i = (\alpha_i - h_{i-1} z_{i-1})/l_i$.

Step 5 Set $l_n = 1$; $z_n = 0$; $c_n = 0$.

Step 6 For $j = n-1, n-2, \ldots, 0$
set $c_j = z_j - \mu_j c_{j+1}$; $b_j = (a_{j+1} - a_j)/h_j - h_j(c_{j+1} + 2c_j)/3$;
$b_j = (a_{j+1} - a_j)/h_j - h_j(c_{j+1} + 2c_j)/3$;

Step 7 OUTPUT $(a_j, b_j, c_j, d_j$ for $j = 0, 1, \ldots, n-1)$;
STOP.

The YouTube video developed by Oscar Veliz can serve as a good illustration of the Cubic Spline. Cubic Spline Video

Theorem (3.12)

If f is defined at $a = x_0 < x_1 < \cdots < x_n = b$ and differentiable at a and b, then f has a unique clamped spline interpolant S on the nodes $x_0, x_1, \ldots, x_n$; that is, a spline interpolant that satisfies the clamped boundary conditions $S'(a) = f'(a)$ and $S'(b) = f'(b)$.

Algorithm 3.5: CLAMPED CUBIC SPLINE

To construct the cubic spline interpolant S for the function f defined at the numbers $x_0 < x_1 < \cdots < x_n$, satisfying $S'(x_0) = f'(x_0)$ and $S'(x_n) = f'(x_n)$:

INPUT n; $x_0, x_1, \ldots, x_n$; $a_0 = f(x_0)$, $a_1 = f(x_1), \ldots, a_n = f(x_n)$; $FPO = f'(x_0)$; $FPN = f'(x_n)$.

OUTPUT a_j, b_j, c_j, d_j for $j = 0, 1, \ldots, n-1$.

(*Note:* $S(x) = S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$ *for* $x_j \le x \le x_{j+1}$.)

Step 1 For $i = 0, 1, \ldots, n-1$ set $h_i = x_{i+1} - x_i$.

Step 2 Set $\alpha_0 = 3(a_1 - a_0)/h_0 - 3FPO$; $\alpha_n = 3FPN - 3(a_n - a_{n-1})/h_{n-1}$.

$\alpha_n = 3FPN - 3(a_n - a_{n-1})/h_{n-1}$.

Step 3 For $i = 1, 2, \ldots, n-1$

set $\alpha_i = \dfrac{3}{h_i}(a_{i+1} - a_i) - \dfrac{3}{h_{i-1}}(a_i - a_{i-1})$.

Step 4 Set $l_0 = 2h_0$; $\mu_0 = 0.5$; ${}_0 = \alpha_0/l_0$. (*Steps 4-6, & part of 7 solve a tridiagonal linear system by method in Algorithm 6.7.*)

Step 5 For $i = 1, 2, \ldots, n-1$

set $l_i = 2(x_{i+1} - x_{i-1}) - h_{i-1}\mu_{i-1}$; $\mu_i = h_i/l_i$; $z_i = (\alpha_i - h_{i-1}z_{i-1})/l_i$.

Step 6 Set $l_n = h_{n-1}(2 - \mu_{n-1})$; $z_n = (\alpha_n - h_{n-1}z_{n-1})/l_n$; $c_n = z_n$.

Step 7 For $j = n-1, n-2, \ldots, 0$

set $c_j = z_j - \mu_j c_{j+1}$; $b_j = (a_{j+1} - a_j)/h_j - h_j(c_{j+1} + 2c_j)/3$; $d_j = (c_{j+1} - c_j)/(3h_j)$.

Step 8 OUTPUT $(a_j, b_j, c_j, d_j$ for $j = 0, 1, \ldots, n-1)$;

STOP.

Theorem (3.13)

Let $f \in C^4[a,b]$ *with* $\max_{a \le x \le b} |f^{(4)}(x)| = M$. *If* S *is the unique clamped cubic spline interpolant to* f *with respect to the nodes* $a = x_0 < x_1 < \cdots < x_n = b$*, then for all* x *in* $[a,b]$*,*

$$|f(x) - S(x)| \le \frac{5M}{384} \max_{0 \le j \le n-1} (x_{j+1} - x_j)^4.$$