LECTURE 11: *Matrix Determination of Splines; Energy Minimization*

1.11.4 *General case, $k \geq 1$*

Now consider the general spline interpolant of degree $k \geq 1$,

$$S_k(x) = \sum_{j=-k}^{n-1} c_{j,k} B_{j,k}(x),$$

with constants $c_{-k,k}, \ldots, c_{n-1,k}$ determined to satisfy the interpolation conditions $S_k(\ell) = f_\ell$, i.e.,

$$\sum_{j=-k}^{n-1} c_{j,k} B_{j,k}(x_\ell) = f_\ell, \quad \ell = 0, \ldots, n.$$

By now we are accustomed to transforming constraints like this into matrix equations. Each value $\ell = 0, \ldots, n$ gives a row of the equation

$$\begin{bmatrix} B_{-k,k}(x_0) & B_{-k+1,k}(x_0) & \cdots & B_{n-1,k}(x_0) \\ B_{-k,k}(x_1) & B_{-k+1,k}(x_1) & \cdots & B_{n-1,k}(x_1) \\ \vdots & \vdots & \ddots & \vdots \\ B_{-k,k}(x_n) & B_{-k+1,k}(x_n) & \cdots & B_{n-1,k}(x_n) \end{bmatrix} \begin{bmatrix} c_{-k,k} \\ c_{-k+1,k} \\ \vdots \\ c_{n-1,k} \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix}. \tag{1.35}$$

Let us consider the matrix in this equation. The matrix will have $n+1$ rows and $n+k$ columns, so when $k > 1$ the system of equations will be *underdetermined*. Since B-splines have 'small support' (i.e., $B_{j,k}(x) = 0$ for most $x \in [x_0, x_n]$), this matrix will be *sparse*: most entries will be zero.

One could obtain an $(n+1) \times (n+1)$ matrix by arbitrarily setting $k-1$ certain values of $c_{j,k}$ to zero, but this would miss a great opportunity: we can constructively include all $n+k$ B-splines and impose k extra properties on S_k to pick out a unique spline interpolant from the infinitely many options that satisfy the interpolation conditions.

The following subsections will describe the particular form the system (1.35) takes for $k = 1, 2, 3$. In each case we will illustrate the resulting spline interpolant through the following data set.

(1.36)

j	0	1	2	3	4
x_j	0	1	2	3	4
f_j	1	3	2	-1	1

1.11.5 *Linear splines, $k = 1$*

Linear splines are simple to construct: in this case $n + k = n + 1$, so the matrix in (1.35) is square. Let us evaluate it: since

$$B_{j,1}(x_\ell) = \begin{cases} 1, & j = \ell; \\ 0, & j \neq \ell, \end{cases}$$

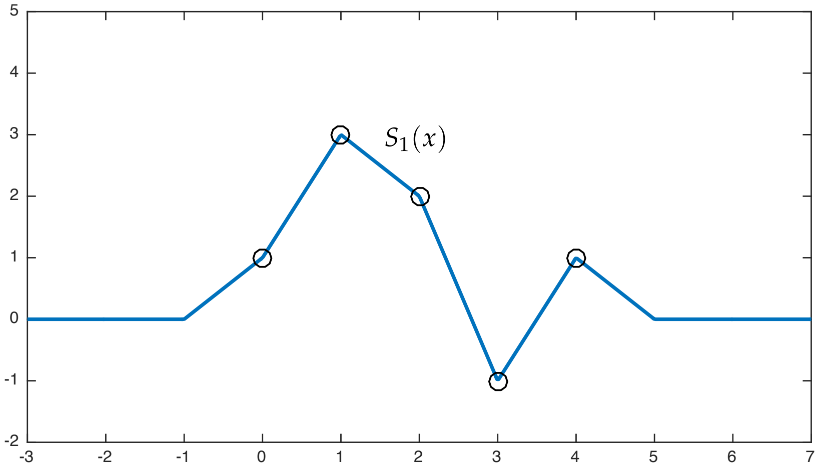

Figure 1.23: Linear spline S_1 interpolating 5 data points $\{(x_j, f_j)\}_{j=0}^4$.

the matrix is simply

$$\begin{bmatrix} B_{-1,1}(x_0) & B_{0,1}(x_0) & \cdots & B_{n-1,1}(x_0) \\ B_{-1,1}(x_1) & B_{0,1}(x_1) & \cdots & B_{n-1,1}(x_1) \\ \vdots & \vdots & \ddots & \vdots \\ B_{-1,1}(x_n) & B_{0,1}(x_n) & \cdots & B_{n-1,1}(x_n) \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix} = \mathbf{I}.$$

The system (1.35) is thus trivial to solve, reducing to

$$\begin{bmatrix} c_{-1,1} \\ c_{-0,k} \\ \vdots \\ c_{n-1,k} \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix},$$

which gives the unique linear spline

$$S_1(x) = \sum_{j=-1}^{n-1} f_{j+1} B_{j,1}(x).$$

This above discussion is a pedantic way to arrive at an obvious solution: Since the jth 'hat function' B-spline equals one at x_{j+1} and zero at all other knots, just write the unique formula for the interpolant immediately.

Figure 1.23 shows the unique piecewise linear spline interpolant to the data in (1.36), which is a linear combination of the five linear splines shown in Figure 1.22. Explicitly,

$$\begin{aligned} S_1(x) &= f_0 B_{-1,1}(x) + f_1 B_{0,1}(x) + f_2 B_{1,1}(x) + f_3 B_{2,1}(x) + f_4 B_{3,1}(x) \\ &= B_{-1,1}(x) + 3\,B_{0,1}(x) + 2\,B_{1,1}(x) - B_{2,1}(x) + B_{3,1}(x). \end{aligned}$$

Note that linear splines are simply C^0 functions that interpolate a given data set—between the knots, they are identical to the piecewise linear functions constructed in Section 1.10.1. Note that $S_1(x)$ is supported on (x_{-1}, x_{n+1}) with $S_1(x) = 0$ for all $x \notin (x_{-1}, x_{n+1})$. This is a general feature of splines: Outside the range of interpolation, $S_k(x)$ goes to zero as quickly as possible for a given set of knots while still maintaining the specified continuity.

1.11.6 *Quadratic splines, $k = 2$*

The construction of quadratic B-splines from the linear splines via the recurrence (1.32) forces the functions $B_{j,2}$ to have a continuous derivative, and also to be supported over three intervals per spline, as seen in the middle plot in Figure 1.22. The interpolant takes the form

$$S_2(x) = \sum_{j=-2}^{n-1} c_{j,2} B_{j,2}(x),$$

with coefficients specified by $n+1$ equations in $n+2$ unknowns:

$$\begin{bmatrix} B_{-2,2}(x_0) & B_{-1,2}(x_0) & \cdots & B_{n-1,2}(x_0) \\ B_{-2,2}(x_1) & B_{-1,2}(x_1) & \cdots & B_{n-1,2}(x_1) \\ \vdots & \vdots & \ddots & \vdots \\ B_{-2,2}(x_n) & B_{-1,2}(x_n) & \cdots & B_{n-1,2}(x_n) \end{bmatrix} \begin{bmatrix} c_{-2,2} \\ c_{-1,2} \\ \vdots \\ c_{n-1,2} \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix}. \tag{1.37}$$

Since there are more variables than constraints, we expect infinitely many quadratic splines that interpolate the data.

Evaluate the entries of the matrix in (1.37). First note that

$$B_{j,2}(x_\ell) = 0, \quad \ell \notin \{j+1, j+2\},$$

so the matrix is zero in all entries except the main diagonal ($B_{j,2}(x_{j+2})$) and the first superdiagonal ($B_{j,2}(x_{j+1})$). To evaluate these nonzero entries, recall that the recursion (1.32) for B-splines gives

$$B_{j,2}(x) = \Big(\frac{x - x_j}{x_{j+2} - x_j}\Big) B_{j,1}(x) + \Big(\frac{x_{j+3} - x}{x_{j+3} - x_{j+1}}\Big) B_{j+1,1}(x).$$

Evaluate this function at x_{j+1} and x_{j+2}, using our knowledge of the

$B_{j,1}$ linear B-splines ('hat functions'):

$$\begin{aligned}
B_{j,2}(x_{j+1}) &= \Big(\frac{x_{j+1}-x_j}{x_{j+2}-x_j}\Big)B_{j,1}(x_{j+1}) + \Big(\frac{x_{j+3}-x_{j+1}}{x_{j+3}-x_{j+1}}\Big)B_{j+1,1}(x_{j+1})\\
&= \Big(\frac{x_{j+1}-x_j}{x_{j+2}-x_j}\Big)\cdot 1 + \Big(\frac{x_{j+3}-x_{j+1}}{x_{j+3}-x_{j+1}}\Big)\cdot 0 = \frac{x_{j+1}-x_j}{x_{j+2}-x_j};\\
B_{j,2}(x_{j+2}) &= \Big(\frac{x_{j+2}-x_j}{x_{j+2}-x_j}\Big)B_{j,1}(x_{j+2}) + \Big(\frac{x_{j+3}-x_{j+2}}{x_{j+3}-x_{j+1}}\Big)B_{j+1,1}(x_{j+2})\\
&= \Big(\frac{x_{j+2}-x_j}{x_{j+2}-x_j}\Big)\cdot 0 + \Big(\frac{x_{j+3}-x_{j+2}}{x_{j+3}-x_{j+1}}\Big)\cdot 1 = \frac{x_{j+3}-x_{j+2}}{x_{j+3}-x_{j+1}}.
\end{aligned}$$

Use these formulas to populate the superdiagonal and subdiagonal of the matrix in (1.37). In the (important) special case of uniformly spaced knots

$$x_j = x_0 + jh, \quad \text{for fixed } h > 0,$$

gives the particularly simple formulas

$$B_{j,2}(x_{j+1}) = B_{j,2}(x_{j+2}) = \frac{1}{2},$$

hence the system (1.37) becomes

$$\begin{bmatrix} 1/2 & 1/2 & & & \\ & 1/2 & 1/2 & & \\ & & \ddots & \ddots & \\ & & & 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} c_{-2,2} \\ c_{-1,2} \\ c_{0,2} \\ \vdots \\ c_{n-1,2} \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix},$$

where the blank entries are zero. This $(n+1)\times(n+2)$ system will have infinitely many solutions, i.e., infinitely many splines that satisfy the interpolation conditions. How to choose among them? Impose *one* extra condition, such as $S_2'(x_0) = 0$ or $S_2'(x_n) = 0$.

As an example, let us work through the condition $S_2'(x_0) = 0$; it raises an interesting issue. Refer to the middle plot in Figure 1.22. Due to the small support of the quadratic B-splines, $B_{j,2}'(x_0) = 0$ for $j > 0$, so

$$S_2'(x_0) = c_{-2,2}B_{-2,2}'(x_0) + c_{-1,2}B_{-1,2}'(x_0) + c_{0,2}B_{0,2}'(x_0). \tag{1.38}$$

The derivatives of the B-splines at knots are tricky to compute. Differentiating the recurrence (1.32) with $k = 2$, we can formally write

$$B_{j,2}'(x) = \Big(\frac{1}{x_{j+2}-x_j}\Big)B_{j,1}(x) + \Big(\frac{x-x_j}{x_{j+2}-x_j}\Big)B_{j,1}'(x) - \Big(\frac{1}{x_{j+3}-x_{j+1}}\Big)B_{j+1,1}(x) + \Big(\frac{x_{j+3}-x}{x_{j+3}-x_{j+1}}\Big)B_{j+1,1}'(x).$$

Try to evaluate this expression at x_j, x_{j+1}, or x_{j+2}: you must face that fact that the linear B-splines $B_{j,1}$ and $B_{j+1,1}$ *are not differentiable at the knots*! You must instead check that the one-sided derivatives match, e.g.,

$$\lim_{\substack{h\to 0\\ h<0}} \frac{B_{j,2}(x_{j+1}+h) - B_{j,2}(x_{j+1})}{h} = \lim_{\substack{h\to 0\\ h>0}} \frac{B_{j,2}(x_{j+1}+h) - B_{j,2}(x_{j+1})}{h}.$$

A mildly tedious calculation verifies that indeed these one-sided first derivatives do match, and that is the point of splines: each time you increase the degree k, you increase the smoothness, so $B_{j,2} \in C^1(\mathbb{R})$.

Now regarding formula (1.38), one can compute

$$B'_{-2,2}(x_0) = -\frac{2}{x_1 - x_{-1}}, \quad B'_{-1,2}(x_0) = \frac{2}{x_1 - x_{-1}}, \quad B'_{0,2}(x_0) = 0,$$

and so, in the special case of a uniformly spaced grid ($x_j = x_0 + jh$), the condition $S'(x_0) = 0$ becomes

$$-\frac{1}{h}c_{-2,2} + \frac{1}{h}c_{-1,2} = 0.$$

Insert this equation as the first row in the linear system for the coefficients,

$$\begin{bmatrix} -1/h & 1/h & & & \\ 1/2 & 1/2 & & & \\ & 1/2 & 1/2 & & \\ & & \ddots & \ddots & \\ & & & 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} c_{-2,2} \\ c_{-1,2} \\ c_{0,2} \\ \vdots \\ c_{n-1,2} \end{bmatrix} = \begin{bmatrix} 0 \\ f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix},$$

and solve this for $c_{-2,2}, \ldots, c_{n-1,2}$ to determine the unique interpolating quadratic spline with $S_2'(x_0) = 0$.

Figure **??** shows quadratic spline interpolants to the data in (1.36). One of these splines is determined with the extra condition $S_2'(x_0) = 0$ described above; the other satisfies $S_2'(x_n) = 0$. In any case, the quadratic spline S_2 is supported on (x_{-2}, x_{n+2}).

1.11.7 *Cubic splines, $k = 3$*

Cubic splines are the most famous of all splines. We began this section by discussing cubic splines as an alternative to piecewise cubic Hermite interpolation. Now we will show how to derive the same cubic splines from the cubic B-splines.

Begin by reviewing the bottom plot in Figure 1.22. The cubic B-splines $B_{-3,3}, \ldots, B_{n-1,3}$ take nonzero values on the interval $[x_0, x_n]$, and hence we write the cubic spline as

$$S_3(x) = \sum_{j=-3}^{n-1} c_{j,3} B_{j,3}(x). \tag{1.39}$$

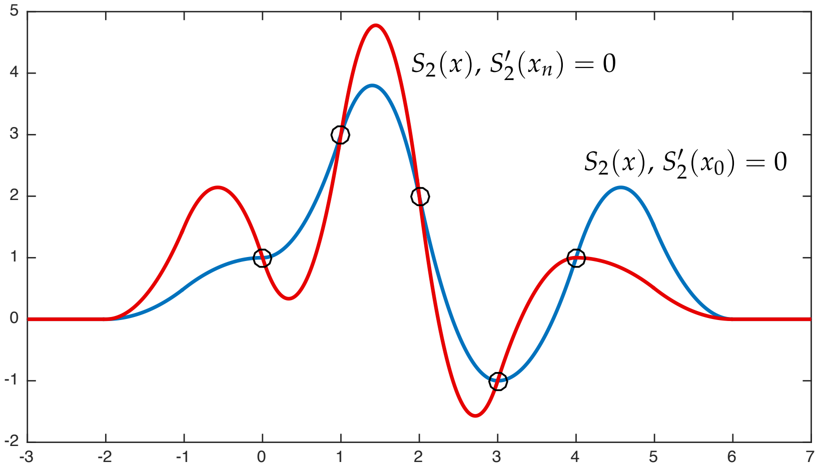

Figure 1.24: Two choices for the quadratic spline S_2 that interpolates the 5 data points $\{(x_j, f_j)\}_{j=0}^{4}$ in (1.36). The blue spline satisfies the extra condition that $S_2'(x_0) = 0$, while the red spline satisfies $S_2'(x_n) = 0$. Check to see that these conditions are consistent with the splines in the plot.

The linear system (1.35) now involves $n+1$ equations in $n+3$ unknowns:

$$\begin{bmatrix} B_{-3,3}(x_0) & B_{-2,3}(x_0) & \cdots & B_{n-1,3}(x_0) \\ B_{-3,3}(x_1) & B_{-2,3}(x_1) & \cdots & B_{n-1,3}(x_1) \\ \vdots & \vdots & \ddots & \vdots \\ B_{-3,3}(x_n) & B_{-2,3}(x_n) & \cdots & B_{n-1,3}(x_n) \end{bmatrix} \begin{bmatrix} c_{-3,3} \\ c_{-2,3} \\ \vdots \\ c_{n-1,3} \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix}. \tag{1.40}$$

Given the support of cubic splines, note that

$$B_{j,3}(x_\ell) = 0, \quad \ell \notin \{j+1, j+2, j+3\},$$

which implies that only three diagonals of the matrix in (1.40) will be nonzero. We shall only work out the nonzero entries in the case of uniformly spaced knots, $x_j = x_0 + jh$ for fixed $h > 0$. In this case,

$$\begin{aligned} B_{j,3}(x_{j+1}) &= \Big(\frac{x_{j+1}-x_j}{x_{j+3}-x_j}\Big)B_{j,2}(x_{j+1}) + \Big(\frac{x_{j+4}-x_{j+1}}{x_{j+4}-x_{j+1}}\Big)B_{j+1,2}(x_{j+1}) = \Big(\frac{h}{3h}\Big)\cdot\frac{1}{2} + \Big(\frac{3h}{3h}\Big)\cdot 0 = \frac{1}{6} \\ B_{j,3}(x_{j+2}) &= \Big(\frac{x_{j+2}-x_j}{x_{j+3}-x_j}\Big)B_{j,2}(x_{j+2}) + \Big(\frac{x_{j+4}-x_{j+2}}{x_{j+4}-x_{j+1}}\Big)B_{j+1,2}(x_{j+2}) = \Big(\frac{2h}{3h}\Big)\cdot\frac{1}{2} + \Big(\frac{2h}{3h}\Big)\cdot\frac{1}{2} = \frac{2}{3} \\ B_{j,3}(x_{j+3}) &= \Big(\frac{x_{j+3}-x_j}{x_{j+3}-x_j}\Big)B_{j,2}(x_{j+3}) + \Big(\frac{x_{j+4}-x_{j+3}}{x_{j+4}-x_{j+1}}\Big)B_{j+1,2}(x_{j+3}) = \Big(\frac{3h}{3h}\Big)\cdot 0 + \Big(\frac{h}{3h}\Big)\cdot\frac{1}{2} = \frac{1}{6}, \end{aligned}$$

where we have used the fact that $B_{j,2}(x_{j+1}) = B_{j,2}(x_{j+2}) = 1/2$ and

$B_{j,2}(x_j) = B_{j,2}(x_{j+3}) = 0$. Substituting these values into (1.40) gives

$$\begin{bmatrix} 1/6 & 2/3 & 1/6 & & & \\ & 1/6 & 2/3 & 1/6 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1/6 & 2/3 & 1/6 \end{bmatrix} \begin{bmatrix} c_{-3,3} \\ c_{-2,3} \\ \vdots \\ c_{n-1,3} \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix} \tag{1.41}$$

involving a matrix with $n+1$ rows and $n+3$ columns. Again, infinitely many cubic splines satisfy these interpolation conditions; two independent requirements must be imposed to determine a unique spline. Recall the three alternatives discussed in Section 1.11.1: complete splines (specify a value for S_3' at x_0 and x_n), natural splines (force $S_3''(x_0) = S_3''(x_n) = 0$), or not-a-knot splines. One can show that imposing natural spline conditions on S_3 requires

$$\begin{aligned} (x_2 - x_{-1})c_{-3,3} - (x_2 + x_1 - x_{-1} - x_{-2})c_{-2,3} + (x_1 - x_{-2})c_{-1,3} &= 0 \\ (x_{n+2} - x_{n-1})c_{n-3,3} - (x_{n+2} + x_{n+1} - x_{n-1} - x_{n-2})c_{n-2,3} + (x_{n+1} - x_{n-2})c_{n-1,3} &= 0, \end{aligned}$$

which for equally spaced knots ($x_j = x_0 + jh$) simplify to

$$\begin{aligned} 3hc_{-3,3} - 6hc_{-2,3} + 3hc_{-1,3} &= 0 \\ 3hc_{n-3,3} - 6hc_{n-2,3} + 3hc_{n-1,3} &= 0. \end{aligned}$$

It is convenient to add these conditions (dividing out the h) as the first and last row of (1.40) to give

$$\begin{bmatrix} 3 & -6 & 3 & & & \\ 1/6 & 2/3 & 1/6 & & & \\ & 1/6 & 2/3 & 1/6 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1/6 & 2/3 & 1/6 \\ & & & 3 & -6 & 3 \end{bmatrix} \begin{bmatrix} c_{-3,3} \\ c_{-2,3} \\ c_{-1,3} \\ \vdots \\ c_{n-2,3} \\ c_{n-1,3} \end{bmatrix} = \begin{bmatrix} 0 \\ f_0 \\ f_1 \\ \vdots \\ f_n \\ 0 \end{bmatrix}. \tag{1.42}$$

This system of $n+3$ equations in $n+3$ variables has a unique solution, the natural cubic spline interpolant.

Figure 1.25 shows the natural cubic spline interpolant to the data (1.36). Clearly this spline satisfies the interpolation conditions, but now there seems to be an artificial peak near $x = 5$ that you might not have anticipated from the data values. This is a feature

It is a useful exercise to work out the extra rows you would add to (1.40) to impose *complete* or *not-a-knot* boundary conditions.

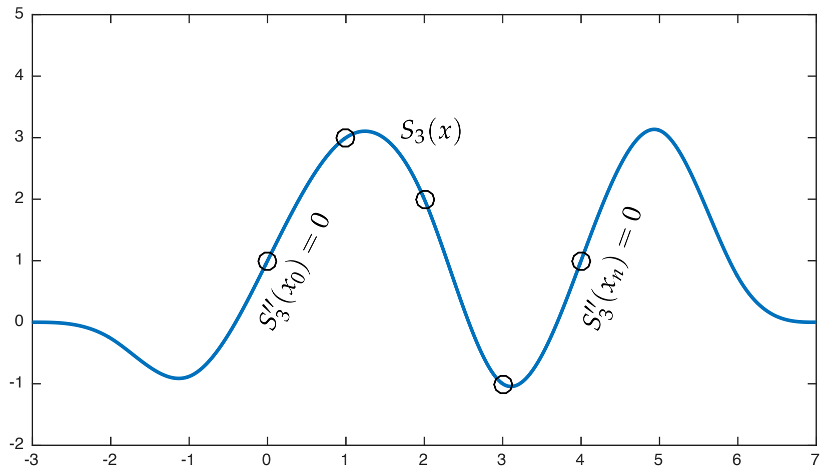

Figure 1.25: Cubic spline S_3 interpolant to 5 data points $\{(x_j, f_j)\}_{j=0}^4$, imposing the two extra *natural spline* conditions $S_3''(x_0) = S_3''(x_n) = 0$ to give a unique spline.

of the natural boundary conditions: by forcing S_3'' to be zero at x_0 and x_n, we ensure that the spline S_3 has constant slope at x_0 and x_n. Eventually this slope must be reversed, as our B-splines force $S_3(x)$ to be zero outside (x_{-3}, x_{n+3}), the support of the B-splines that contribute to the sum (1.39).

Of course, one can implement splines of higher degree, $k > 3$, if if greater continuity is required at the knots, or if there are more than two boundary conditions to impose (e.g., if one wants both first and second derivatives to be zero at the boundary). The procedure in that case follows the pattern detailed above: work out the entries in the matrix (1.35) and add in rows to encode the additional $k-1$ constraints needed to specify a unique degree-k spline interpolant.

1.11.8 Optimality properties of splines

Splines often enjoy a beautiful property: among all sufficiently smooth interpolants , certain splines minimize 'energy', quantified for a function $g \in C^2[x_0, x_n]$ as

$$\int_{x_0}^{x_n} g''(x)^2 \, \mathrm{d}x.$$

To give a flavor for such results, we present one example.

For a similar result involving complete cubic splines, see Theorem 2.3.1 of Gautschi's *Numerical Analysis* (2nd ed., Birkhäuser, 2012). The proof here is an easy adaptation of Gautchsi's.

Theorem 1.10 (Natural cubic splines minimize energy)**.**
Suppose S_3 is the natural cubic spline interpolant to $\{(x_j, f_j)\}_{j=0}^n$, and g is any $C^2[x_0, x_n]$ function that also interpolates the same data. Then

$$\int_{x_0}^{x_n} S_3''(x)^2 \, \mathrm{d}x \le \int_{x_0}^{x_n} g''(x)^2 \, \mathrm{d}x.$$

Proof. The proof will actually quantify how much larger g'' is than S_3'' by showing that

$$\int_{x_0}^{x_n} g''(x)^2\,\mathrm{d}x = \int_{x_0}^{x_n} S_3''(x)^2\,\mathrm{d}x + \int_{x_0}^{x_n} \big(g''(x) - S_3''(x)\big)^2\,\mathrm{d}x. \tag{1.43}$$

Expanding the right-hand side, see that this claim is equivalent to

$$\int_{x_0}^{x_n} \big(g''(x) - S_3''(x)\big)S_3''(x)\,\mathrm{d}x = 0. \tag{1.44}$$

To prove this claim, break the integral on the left into segments $[x_j, x_{j+1}]$ between the knots. Write

$$\int_{x_0}^{x_n} \big(g''(x) - S_3''(x)\big)S_3''(x)\,\mathrm{d}x = \sum_{j=1}^{n}\int_{x_{j+1}}^{x_j} \big(g''(x) - S_3''(x)\big)S_3''(x)\,\mathrm{d}x.$$

This decomposition of $[x_0, x_n]$ will allow us to exploit the fact that S_3 is a standard cubic polynomial, and hence infinitely differentiable, on these subintervals.

On each subinterval, integrate by parts to obtain

$$\int_{x_{j+1}}^{x_j} \big(g''(x) - S_3''(x)\big)S_3''(x)\,\mathrm{d}x = \Big[(g'(x) - S_3'(x))S_3''(x)\Big]_{x=x_{j-1}}^{x_j} - \int_{x_{j+1}}^{x_j} \big(g'(x) - S_3'(x)\big)S_3'''(x)\,\mathrm{d}x. \tag{1.45}$$

Focus now on the integral on the right-hand side; we can show it is *zero* by integrating it by parts to get

$$\int_{x_{j+1}}^{x_j} \big(g'(x) - S_3'(x)\big)S_3'''(x)\,\mathrm{d}x = \Big[(g(x) - S_3(x))S_3'''(x)\Big]_{x=x_{j-1}}^{x_j} - \int_{x_{j+1}}^{x_j} \big(g(x) - S_3(x)\big)S_3''''(x)\,\mathrm{d}x. \tag{1.46}$$

The boundary term on the right is zero, since $g(x_\ell) - S_3(x_\ell) = 0$ for $\ell = 0, \ldots, n$ (both g and S_3 must interpolate the data). The integral on the right is also zero: since S_3 is a cubic polynomial on $[x_{j-1}, x_j]$, $S_3''''(x) = 0$. Thus (1.45) reduces to

$$\int_{x_{j+1}}^{x_j} \big(g''(x) - S_3''(x)\big)S_3''(x)\,\mathrm{d}x = \Big[(g'(x) - S_3'(x))S_3''(x)\Big]_{x=x_{j-1}}^{x_j}$$

Adding up these contributions over all the subintervals,

$$\int_{x_0}^{x_n} \big(g''(x) - S_3''(x)\big)S_3''(x)\,\mathrm{d}x = \sum_{j=1}^{n}\Big[(g'(x) - S_3'(x))S_3''(x)\Big]_{x=x_{j-1}}^{x_j}.$$

Most of the boundary terms on the right cancel one another out, leaving only

$$\int_{x_0}^{x_n} \big(g''(x) - S_3''(x)\big)S_3''(x)\,\mathrm{d}x = \Big((g'(x_n) - S_3'(x_n))S_3''(x_n)\Big) - \Big((g'(x_0) - S_3'(x_0))S_3''(x_0)\Big).$$

Each of the terms on the right is zero by virtue of the *natural* cubic spline condition $S_3''(x_0) = S_3''(x_n) = 0$. This confirms the formula (1.44), and hence the equivalent (1.43) that quantifies how much larger g'' can be than S_3''. ■

1.11.9 *Some omissions*

The great utility of B-splines in engineering has led to the development of the subject far beyond these basic notes. Among the omissions are: interpolation imposed at points distinct from the knots, convergence of splines to the function they are approximating as the number of knots increases, integration and differentiation of splines, tension splines, etc. Splines in higher dimensions ('thin-plate splines') are used, for example, to design the panels of a car body.