IMPROVED INTERPOLATION

Recall the examples of higher degree polynomial interpolation of the function $f(x) = \left(1 + x^2\right)^{-1}$ on $[-5, 5]$. The interpolants $P_n(x)$ oscillated a great deal, whereas the function $f(x)$ was nonoscillatory. To obtain interpolants that are better behaved, we look at other forms of interpolating functions.

Consider the data

x	0	1	2	2.5	3	3.5	4
y	2.5	0.5	0.5	1.5	1.5	1.125	0

What are methods of interpolating this data, other than using a degree 6 polynomial. Shown in the text (pages 130,131) are the graphs of the degree 6 polynomial interpolant, along with those of piecewise linear and a piecewise quadratic interpolating functions.

Since we only have the data to consider, we would generally want to use an interpolant that had somewhat the shape of that of the piecewise linear interpolant.

PIECEWISE POLYNOMIAL FUNCTIONS

Consider being given a set of data points $(x_1, y_1), ..., (x_n, y_n)$, with

$$x_1 < x_2 < \cdots < x_n$$

Then the simplest way to connect the points (x_j, y_j) is by straight line segments. This is called a piecewise linear interpolant of the data $\left\{(x_j, y_j)\right\}$. This graph has "corners", and often we expect the interpolant to have a smooth graph.

To obtain a somewhat smoother graph, consider using piecewise quadratic interpolation. Begin by constructing the quadratic polynomial that interpolates

$$\{(x_1, y_1), (x_2, y_2), (x_3, y_3)\}$$

Then construct the quadratic polynomial that interpolates

$$\{(x_3, y_3), (x_4, y_4), (x_5, y_5)\}$$

Continue this process of constructing quadratic interpolants on the subintervals

$$[x_1, x_3], [x_3, x_5], [x_5, x_7], ...$$

If the number of subintervals is even (and therefore n is odd), then this process comes out fine, with the last interval being $[x_{n-2}, x_n]$. This was illustrated on the graph for the preceding data. If, however, n is even, then the approximation on the last interval must be handled by some modification of this procedure. Suggest such!

With piecewise quadratic interpolants, however, there are "corners" on the graph of the interpolating function. With our preceding example, they are at x_3 and x_5. How do we avoid this?

Piecewise polynomial interpolants are used in many applications. We will consider them later, to obtain numerical integration formulas.

SMOOTH NON-OSCILLATORY INTERPOLATION

Let data points $(x_1, y_1), ..., (x_n, y_n)$ be given, as let

$$x_1 < x_2 < \cdots < x_n$$

Consider finding functions $s(x)$ for which the following properties hold:

(1) $s(x_i) = y_i, \qquad i = 1, ..., n$

(2) $s(x)$, $s'(x)$, $s''(x)$ are continuous on $[x_1, x_n]$.

Then among such functions $s(x)$ satisfying these properties, find the one which minimizes the integral

$$\int_{x_1}^{x_n} \left| s''(x) \right|^2 \, dx$$

The idea of minimizing the integral is to obtain an interpolating function for which the first derivative does not change rapidly. It turns out there is a unique solution to this problem, and it is called a natural cubic spline function.

SPLINE FUNCTIONS

Let a set of node points $\{x_i\}$ be given, satisfying

$$a \leq x_1 < x_2 < \cdots < x_n \leq b$$

for some numbers a and b. Often we use $[a, b] = [x_1, x_n]$. A cubic spline function $s(x)$ on $[a, b]$ with "breakpoints" $\{x_i\}$ has the following properties:
1. On each of the intervals

$$[a, x_1], [x_1, x_2], ..., [x_{n-1}, x_n], [x_n, b]$$

$s(x)$ is a polynomial of degree ≤ 3.
2. $s(x)$, $s'(x)$, $s''(x)$ are continuous on $[a, b]$.

In the case that we have given data points (x_1, y_1),...,(x_n, y_n) we say $s(x)$ is a cubic interpolating spline function for this data if
3. $s(x_i) = y_i, \quad i = 1, ..., n$.

EXAMPLE

Define

$$(x-\alpha)_+^3 = \begin{cases} (x-\alpha)^3, & x \geq \alpha \\ 0, & x \leq \alpha \end{cases}$$

This is a cubic spline function on $(-\infty, \infty)$ with the single breakpoint $x_1 = \alpha$.

Combinations of these form more complicated cubic spline functions. For example,

$$s(x) = 3\,(x-1)_+^3 - 2\,(x-3)_+^3$$

is a cubic spline function on $(-\infty, \infty)$ with the breakpoints $x_1 = 1$, $x_2 = 3$.

Define

$$s(x) = p_3(x) + \sum_{j=1}^{n} a_j \left(x - x_j\right)_+^3$$

with $p_3(x)$ some cubic polynomial. Then $s(x)$ is a cubic spline function on $(-\infty, \infty)$ with breakpoints $\{x_1, ..., x_n\}$.

Return to the earlier problem of choosing an interpolating function $s(x)$ to minimize the integral

$$\int_{x_1}^{x_n} \left| s''(x) \right|^2 \, dx$$

There is a unique solution to problem. The solution $s(x)$ is a cubic interpolating spline function, and moreover, it satisfies

$$s''(x_1) = s''(x_n) = 0$$

Spline functions satisfying these boundary conditions are called "natural" cubic spline functions, and the solution to our minimization problem is a "natural cubic interpolatory spline function". We will show a method to construct this function from the interpolation data.

Motivation for these boundary conditions can be given by looking at the physics of bending thin beams of flexible materials to pass thru the given data. To the left of x_1 and to the right of x_n, the beam is straight and therefore the second derivatives are zero at the transition points x_1 and x_n.

CONSTRUCTION OF THE INTERPOLATING SPLINE FUNCTION

To make the presentation more specific, suppose we have data

$$(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$$

with $x_1 < x_2 < x_3 < x_4$. Then on each of the intervals

$$[x_1, x_2], [x_2, x_3], [x_3, x_4]$$

$s(x)$ is a cubic polynomial. Taking the first interval, $s(x)$ is a cubic polynomial and $s''(x)$ is a linear polynomial. Let

$$M_i = s''(x_i), \quad i = 1, 2, 3, 4$$

Then on $[x_1, x_2]$,

$$s''(x) = \frac{(x_2 - x)\, M_1 + (x - x_1)\, M_2}{x_2 - x_1}, \quad x_1 \le x \le x_2$$

We can find $s(x)$ by integrating twice:

$$s(x) = \frac{(x_2 - x)^3 M_1 + (x - x_1)^3 M_2}{6\,(x_2 - x_1)} + c_1 x + c_2$$

We determine the constants of integration by using

$$s(x_1) = y_1, \quad s(x_2) = y_2 \qquad (*)$$

Then

$$\begin{aligned} s(x) = {} & \frac{(x_2 - x)^3 M_1 + (x - x_1)^3 M_2}{6\,(x_2 - x_1)} \\ & + \frac{(x_2 - x)\, y_1 + (x - x_1)\, y_2}{x_2 - x_1} \\ & - \frac{x_2 - x_1}{6} \left[(x_2 - x)\, M_1 + (x - x_1)\, M_2\right] \end{aligned}$$

for $x_1 \leq x \leq x_2$. Check that this formula satisfies the given interpolation conditions($*$)!

We can repeat this on the intervals $[x_2, x_3]$ and $[x_3, x_4]$, obtaining similar formulas.

$$s(x) = \frac{(x_3 - x)^3 M_2 + (x - x_2)^3 M_3}{6(x_3 - x_2)} + \frac{(x_3 - x) y_2 + (x - x_2) y_3}{x_3 - x_2} - \frac{x_3 - x_2}{6} [(x_3 - x) M_2 + (x - x_2) M_3]$$

for $x_2 \le x \le x_3$.

$$s(x) = \frac{(x_4 - x)^3 M_3 + (x - x_3)^3 M_4}{6(x_4 - x_3)} + \frac{(x_4 - x) y_3 + (x - x_3) y_4}{x_4 - x_3} - \frac{x_4 - x_3}{6} [(x_4 - x) M_3 + (x - x_3) M_4]$$

for $x_3 \le x \le x_4$.

We still do not know the values of the second derivatives $\{M_1, M_2, M_3, M_4\}$. The above formulas guarantee that $s(x)$ and $s''(x)$ are continuous for $x_1 \leq x \leq x_4$. For example, the formula on $[x_1, x_2]$ yields

$$s(x_2) = y_2, \qquad s''(x_2) = M_2$$

The formula on $[x_2, x_3]$ also yields

$$s(x_2) = y_2, \qquad s''(x_2) = M_2$$

All that is lacking is to make $s'(x)$ continuous at x_2 and x_3. Thus we require

$$\begin{array}{l} s'(x_2 + 0) = s'(x_2 - 0) \\ s'(x_3 + 0) = s'(x_3 - 0) \end{array} \qquad (**)$$

This means

$$\lim_{x \searrow x_2} s'(x) = \lim_{x \nearrow x_2} s'(x)$$

and similarly for x_3.

To simplify the presentation somewhat, I assume in the following that our node points are evenly spaced:

$$x_2 = x_1 + h, \quad x_3 = x_1 + 2h, \quad x_4 = x_1 + 3h$$

Then our earlier formulas simplify to

$$\begin{aligned} s(x) = {} & \frac{(x_2 - x)^3 M_1 + (x - x_1)^3 M_2}{6h} \\ & + \frac{(x_2 - x) y_1 + (x - x_1) y_2}{h} \\ & - \frac{h}{6} \left[(x_2 - x) M_1 + (x - x_1) M_2 \right] \end{aligned}$$

for $x_1 \leq x \leq x_2$, with similar formulas on $[x_2, x_3]$ and $[x_3, x_4]$.

Without going thru all of the algebra, the conditions $(**)$ leads to the following pair of equations.

$$\frac{h}{6}M_1 + \frac{2h}{3}M_2 + \frac{h}{6}M_3 = \frac{y_3 - y_2}{h} - \frac{y_2 - y_1}{h}$$

$$\frac{h}{6}M_2 + \frac{2h}{3}M_3 + \frac{h}{6}M_4 = \frac{y_4 - y_3}{h} - \frac{y_3 - y_2}{h}$$

$$\frac{h}{6}M_1 + \frac{2h}{3}M_2 + \frac{h}{6}M_3 = \frac{y_3 - y_2}{h} - \frac{y_2 - y_1}{h}$$

$$\frac{h}{6}M_2 + \frac{2h}{3}M_3 + \frac{h}{6}M_4 = \frac{y_4 - y_3}{h} - \frac{y_3 - y_2}{h}$$

This gives us two equations in four unknowns. The earlier boundary conditions on $s''(x)$ gives us immediately

$$M_1 = M_4 = 0$$

Then we can solve the linear system for M_2 and M_3.

EXAMPLE

Consider the interpolation data points

x	1	2	3	4
y	1	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$

In this case, $h = 1$, and linear system becomes

$$\begin{aligned}
\frac{2}{3}M_2 + \frac{1}{6}M_3 &= y_3 - 2y_2 + y_1 = \frac{1}{3} \\
\frac{1}{6}M_2 + \frac{2}{3}M_3 &= y_4 - 2y_3 + y_2 = \frac{1}{12}
\end{aligned}$$

This has the solution

$$M_2 = \frac{1}{2}, \qquad M_3 = 0$$

This leads to the spline function formula on each subinterval.

On $[1, 2]$,

$$s(x) = \frac{(x_2 - x)^3 M_1 + (x - x_1)^3 M_2}{6h} + \frac{(x_2 - x) y_1 + (x - x_1) y_2}{h} - \frac{h}{6} [(x_2 - x) M_1 + (x - x_1) M_2]$$

$$= \frac{(2 - x)^3 \cdot 0 + (x - 1)^3 \left(\frac{1}{2}\right)}{6} + \frac{(2 - x) \cdot 1 + (x - 1) \left(\frac{1}{2}\right)}{1} - \frac{1}{6} \left[(2 - x) \cdot 0 + (x - 1) \left(\frac{1}{2}\right)\right]$$

$$= \tfrac{1}{12} (x - 1)^3 - \tfrac{7}{12} (x - 1) + 1$$

Similarly, for $2 \leq x \leq 3$,

$$s(x) = \frac{-1}{12} (x - 2)^3 + \frac{1}{4} (x - 2)^2 - \frac{1}{3} (x - 1) + \frac{1}{2}$$

and for $3 \leq x \leq 4$,

$$s(x) = \frac{-1}{12} (x - 4) + \frac{1}{4}$$

Return to the equations

$$\frac{h}{6}M_1 + \frac{2h}{3}M_2 + \frac{h}{6}M_3$$
$$= \frac{y_3 - y_2}{h} - \frac{y_2 - y_1}{h}$$
$$\frac{h}{6}M_2 + \frac{2h}{3}M_3 + \frac{h}{6}M_4$$
$$= \frac{y_4 - y_3}{h} - \frac{y_3 - y_2}{h}$$

Sometimes other boundary conditions are imposed on $s(x)$ to help in determining the values of M_1 and M_4. For example, the data in our numerical example were generated from the function $f(x) = \frac{1}{x}$. With it, $f''(x) = \frac{2}{x^3}$, and thus we could use

$$M_1 = 2, \qquad M_4 = \frac{1}{32}$$

With this we are led to a new formula for $s(x)$, one that approximates $f(x) = \frac{1}{x}$ more closely.

THE CLAMPED SPLINE

In this case, we augment the interpolation conditions

$$s(x_i) = y_i, \quad i = 1, 2, 3, 4$$

with the boundary conditions

$$s'(x_1) = y_1', \qquad s'(x_4) = y_4' \qquad (\#)$$

The conditions (#) lead to another pair of equations, augmenting the earlier ones. Combined these equations are

$$\frac{h}{3}M_1 + \frac{h}{6}M_2 = \frac{y_2 - y_1}{h} - y_1'$$

$$\frac{h}{6}M_1 + \frac{2h}{3}M_2 + \frac{h}{6}M_3$$

$$= \frac{y_3 - y_2}{h} - \frac{y_2 - y_1}{h}$$

$$\frac{h}{6}M_2 + \frac{2h}{3}M_3 + \frac{h}{6}M_4$$

$$= \frac{y_4 - y_3}{h} - \frac{y_3 - y_2}{h}$$

$$\frac{h}{6}M_3 + \frac{h}{3}M_4 = y_4' - \frac{y_4 - y_3}{h}$$

For our numerical example, it is natural to obtain these derivative values from $f'(x) = -\frac{1}{x^2}$:

$$y_1' = -1, \qquad y_4' = -\frac{1}{16}$$

When combined with your earlier equations, we have the system

$$\begin{aligned}
\frac{1}{3}M_1 + \frac{1}{6}M_2 \qquad &= \frac{1}{2} \\
\frac{1}{6}M_1 + \frac{2}{3}M_2 + \frac{1}{6}M_3 &= \frac{1}{3} \\
\frac{1}{6}M_2 + \frac{2}{3}M_3 + \frac{1}{6}M_4 &= \frac{1}{12} \\
\frac{1}{6}M_3 + \frac{1}{3}M_4 \qquad &= \frac{1}{48}
\end{aligned}$$

This has the solution

$$[M_1, M_2, M_3, M_4] = \left[\frac{173}{120}, \frac{7}{60}, \frac{11}{120}, \frac{1}{60}\right]$$

We can now write the functions $s(x)$ for each of the subintervals $[x_1, x_2]$, $[x_2, x_3]$, and $[x_3, x_4]$. Recall for $x_1 \leq x \leq x_2$,

$$s(x) = \frac{(x_2 - x)^3 M_1 + (x - x_1)^3 M_2}{6h} + \frac{(x_2 - x)\, y_1 + (x - x_1)\, y_2}{h} - \frac{h}{6}\left[(x_2 - x)\, M_1 + (x - x_1)\, M_2\right]$$

We can substitute in from the data

x	1	2	3	4
y	1	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$

and the solutions $\{M_i\}$. Doing so, consider the error $f(x) - s(x)$. As an example,

$$f(x) = \frac{1}{x}, \quad f\left(\frac{3}{2}\right) = \frac{2}{3}, \qquad s\left(\frac{3}{2}\right) = .65260$$

This is quite a decent approximation.

THE GENERAL PROBLEM

Consider the spline interpolation problem with n nodes

$$\left(x_1, y_1\right), \left(x_2, y_2\right), ..., \left(x_n, y_n\right)$$

and assume the node points $\{x_i\}$ are evenly spaced,

$$x_j = x_1 + (j-1)\,h, \qquad j = 1, ..., n$$

We have that the interpolating spline $s(x)$ on $x_j \leq x \leq x_{j+1}$ is given by

$$\begin{aligned} s(x) = & \frac{\left(x_{j+1}-x\right)^3 M_j + \left(x-x_j\right)^3 M_{j+1}}{6h} \\ & + \frac{\left(x_{j+1}-x\right) y_j + \left(x-x_j\right) y_{j+1}}{h} \\ & - \frac{h}{6}\left[\left(x_{j+1}-x\right) M_j + \left(x-x_j\right) M_{j+1}\right] \end{aligned}$$

for $j = 1, ..., n-1$.

To enforce continuity of $s'(x)$ at the interior node points $x_2, ..., x_{n-1}$, the second derivatives $\left\{M_j\right\}$ must satisfy the linear equations

$$\frac{h}{6}M_{j-1} + \frac{2h}{3}M_j + \frac{h}{6}M_{j+1} = \frac{y_{j-1} - 2y_j + y_{j+1}}{h}$$

for $j = 2, ..., n-1$. Writing them out,

$$\begin{aligned}
\frac{h}{6}M_1 + \frac{2h}{3}M_2 + \frac{h}{6}M_3 &= \frac{y_1 - 2y_2 + y_3}{h} \\
\frac{h}{6}M_2 + \frac{2h}{3}M_3 + \frac{h}{6}M_4 &= \frac{y_2 - 2y_3 + y_4}{h} \\
&\vdots \\
\frac{h}{6}M_{n-2} + \frac{2h}{3}M_{n-1} + \frac{h}{6}M_n &= \frac{y_{n-2} - 2y_{n-1} + y_n}{h}
\end{aligned}$$

This is a system of $n-2$ equations in the n unknowns $\{M_1, ..., M_n\}$. Two more conditions must be imposed on $s(x)$ in order to have the number of equations equal the number of unknowns, namely n. With the added boundary conditions, this form of linear system can be solved very efficiently.

BOUNDARY CONDITIONS

<u>"Natural" boundary conditions</u>

$$s''(x_1) = s''(x_n) = 0$$

Spline functions satisfying these conditions are called "natural cubic splines". They arise out the minimization problem stated earlier. But generally they are not considered as good as some other cubic interpolating splines.

<u>"Clamped" boundary conditions</u> We add the conditions

$$s'(x_1) = y_1', \qquad s'(x_n) = y_n'$$

with y_1', y_n' given slopes for the endpoints of $s(x)$ on $[x_1, x_n]$. This has many quite good properties when compared with the natural cubic interpolating spline; but it does require knowing the derivatives at the endpoints.

<u>"Not a knot" boundary conditions</u> This is more complicated to explain, but it is the version of cubic spline interpolation that is implemented in Matlab.

THE "NOT A KNOT" CONDITIONS

As before, let the interpolation nodes be

$$(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$$

We separate these points into two categories. For constructing the interpolating cubic spline function, we use the points

$$(x_1, y_1), (x_3, y_3), ..., (x_{n-2}, y_{n-2}), (x_n, y_n)$$

Thus deleting two of the points. We now have $n-2$ points, and the interpolating spline $s(x)$ can be determined on the intervals

$$[x_1, x_3], [x_3, x_4], ..., [x_{n-3}, x_{n-2}], [x_{n-2}, x_n]$$

This leads to $n-4$ equations in the $n-2$ unknowns $M_1, M_3, ..., M_{n-2}, M_n$. The two additional boundary conditions are

$$s(x_2) = y_2, \qquad s(x_{n-1}) = y_{n-1}$$

These translate into two additional equations, and we obtain a system of $n-2$ linear simultaneous equations in the $n-2$ unknowns $M_1, M_3, ..., M_{n-2}, M_n$.

MATLAB SPLINE FUNCTION LIBRARY

Given data points

$$(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$$

type arrays containing the x and y coordinates:

$$x = [x_1 \; x_2 \; ...x_n]$$
$$y = [y_1 \; y_2 \; ...y_n]$$
$$plot\,(x, y, \text{'o'})$$

The last statement will draw a plot of the data points, marking them with the letter 'oh'. To find the interpolating cubic spline function and evaluate it at the points of another array xx, say

$$h = (x_n - x_1)\,/\,(4n); \qquad xx = x_1 : h : x_n$$

use

$$yy = spline\,(x, y, xx)$$
$$plot\,(x, y, \text{'o'}, xx, yy)$$

The last statement will plot the data points, as before, and it will plot the interpolating spline $s(x)$ as a continuous curve.

ERROR IN CUBIC SPLINE INTERPOLATION

Let an interval $[a, b]$ be given, and then define

$$h = \frac{b-a}{n-1}, \qquad x_j = a + (j-1)h, \quad j = 1, ..., n$$

Suppose we want to approximate a given function $f(x)$ on the interval $[a, b]$ using cubic spline interpolation. Define

$$y_i = f(x_i), \qquad j = 1, ..., n$$

Let $s_n(x)$ denote the cubic spline interpolating this data and satisfying the "not a knot" boundary conditions. Then it can be shown that for a suitable constant c,

$$E_n \equiv \max_{a \le x \le b} |f(x) - s_n(x)| \le ch^4$$

EXAMPLE

Take $f(x) = \arctan x$ on $[0, 5]$. The following table gives values of the maximum error E_n for various values of n. The values of h are being successively halved.

n	7	13	25	49	97
E_n	7.09E−3	3.24E−4	3.06E−5	1.48E−6	9.04E−8