Computer Science/Mathematics 455
Lecture Notes
Lecture # 25

Cubic Splines

The cubic spline interpolation problem takes a set of points (called *knots* or *nodes*) such that

$$a = x_0 < x_1 < \ldots < x_n = b.$$

Find

$$s(x) = \begin{cases} s_0(x) & x \in [x_0, x_1] \\ s_1(x) & x \in [x_1, x_2] \\ \vdots & \vdots \\ s_{n-1}(x) & x \in [x_k, x_{k+1}] \end{cases}$$

where $s_k(x)$ is a cubic polynomial and

$$s(x_k) = f(x_k) = f_k, \quad k = 0, \ldots, n$$

and s,s', and s'' are continous.

We take $s_k(x)$ to have the form

$$s_k(x) = a_k + b_k(x - x_k) + c_k(x - x_k)^2 + d_k(x - x_k)^3.$$

Thus we must solve for $4n$ coefficients.

The first two derivatives of s_k are

$$\begin{aligned} s_k'(x) &= b_k + 2c_k(x - x_k) + 3d_k(x - x_k)^2 \\ s_k''(x) &= 2c_k + 6d_k(x - x_k). \end{aligned}$$

We have that

$$s_k''(x_k) = 2c_k.$$

By continuity of s''

$$s_k''(x_{k+1}) = 2c_k + 6d_k h_k = s_{k+1}''(x_{k+1}) = 2c_{k+1}.$$

where $h_k = x_{k+1} - x_k$ is the spacing between points.

Solving for d_k yields

$$d_k = (c_{k+1} - c_k)/(3h_k). \tag{1}$$

We have that

$$s_k(x_k) = a_k = f_k$$

and

$$s_k(x_{k+1}) = f_{k+1} = f_k + b_k h_k + c_k h_k^2 + (c_{k+1} - c_k)h_k^3/3$$

so

$$b_k = \delta_k - (2c_k + c_{k+1})h_k/3, \quad \delta_k = (f_{k+1} - f_k)/h_k. \tag{2}$$

Thus we need only solve for the c_k. For that, we use first derivative continuity!

We have that

$$s_k'(x_{k+1}) = b_k + 2c_k h_k + 3d_k h_k^2 = s_{k+1}(x_{k+1}) = b_{k+1}.$$

Using the expressions (2) for b_k and (1) for d_k in terms of c_k and c_{k+1} and the definition of δ_k in (2) we obtain

$$\delta_k - \frac{h_k}{3}(c_{k+1} + 2c_k) + 2c_k h_k + (c_{k+1} - c_k)h_k = \delta_{k+1} - \frac{h_{k+1}}{3}(c_{k+2} + c_{k+1}).$$

After some algebra we get

$$h_k c_k + 2(h_k + h_{k+1})c_{k+1} + h_{k+1}c_{k+2} = 3(\delta_{k+1} - \delta_k), \quad k = 0, \ldots, n-2 \tag{3}$$

This is $n-1$ linear equations in the $n+1$ variables $c_0, c_1, \ldots, c_n$ necessary to construct the spline.

We need two more conditions, called *endpoint* conditions.

The simplest are for *natural* or *variational* splines. Here we set

$$s''(x_0) = 2c_0 = s''(x_n) = 2c_n = 0.$$

This leads to a tridiagonal system of equations. The following is the version for 7 knots. It is 5×5.

$$T\mathbf{c} = \mathbf{b} \tag{4}$$

where

$$T = \begin{pmatrix} 2(h_0+h_1) & h_1 & 0 & 0 & 0 \\ h_1 & 2(h_1+h_2) & h_2 & 0 & 0 \\ 0 & h_2 & 2(h_2+h_3) & h_3 & 0 \\ 0 & 0 & h_3 & 2(h_3+h_4) & h_4 \\ 0 & 0 & 0 & h_4 & 2(h_4+h_5) \end{pmatrix}$$

$$\mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{pmatrix}, \quad \mathbf{b} = 3 \begin{pmatrix} \delta_1 - \delta_0 \\ \delta_2 - \delta_1 \\ \delta_3 - \delta_2 \\ \delta_4 - \delta_3 \\ \delta_5 - \delta_4 \end{pmatrix}$$

This assumes that $c_0 = c_6 = 0$. This is a particularly nice linear system. It is diagonally dominant and tridiagonal (zero except for the main diagonal, the superdiagonal, and the subdiagonal). It does require pivoting in Gaussian elimination and can be solved in $O(n)$ operations where n is the number of knots in the spline.

Other endpoint conditions lead to slightly different systems.

For instance, first derivative conditions lead to *complete* splines. These specify that

$$s'(x_0) = f'(x_0), \quad s'(x_n) = f'(x_n).$$

If we have first derivative information at the endpoints, these lead to better approximations than do the natural splines.

Since

$$\begin{aligned} s'(x_0) \quad &= b_0 = \delta_0 - (2c_0 + c_1)h_0/3 = f'(x_0), \\ s'(x_n) \quad &= b_{n-1} + 2c_{n-1}h_{n-1} + 3d_{n-1}h_{n-1}^2 = f'(x_n) \end{aligned}$$

Use of (2) and (1) and some algebra yields the two extra conditions

$$\begin{aligned} (2c_0 + c_1)h_0/3 &= \delta_0 - f'(x_0), \\ (2c_n + c_{n-1})h_0/3 &= f'(x_n) - \delta_{n-1} \end{aligned}$$

Thus for the 7 node example we obtain (4) with

$$T = \begin{pmatrix} 2h_0 & h_0 & 0 & 0 & 0 & 0 & \\ h_0 & 2(h_0+h_1) & h_1 & 0 & 0 & 0 & 0 \\ 0 & h_1 & 2(h_1+h_2) & h_2 & 0 & 0 & 0 \\ 0 & 0 & h_2 & 2(h_2+h_3) & h_3 & 0 & 0 \\ 0 & 0 & 0 & h_3 & 2(h_3+h_4) & h_4 & 0 \\ 0 & 0 & 0 & 0 & h_4 & 2(h_4+h_5) & h_5 \\ 0 & 0 & 0 & 0 & 0 & h_5 & 2h_5 \end{pmatrix}$$

$$\mathbf{c} = \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \end{pmatrix}, \quad \mathbf{b} = 3 \begin{pmatrix} \delta_0 - f'(x_0) \\ \delta_1 - \delta_0 \\ \delta_2 - \delta_1 \\ \delta_3 - \delta_2 \\ \delta_4 - \delta_3 \\ \delta_5 - \delta_4 \\ f'(x_n) - \delta_5 \end{pmatrix}.$$

Example 1

$$f(x) = \cos\, x, \quad x \in [0, \pi/2]$$

Let $x_k = \pi * k/12$ *giving us 7 knots as above. For the natural splines we get Then*

$$T = \pi/12 \begin{pmatrix} 4 & 1 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 1 & 4 \end{pmatrix}$$

Some computations yield

$$\mathbf{b} = \begin{pmatrix} -0.7543 \\ -0.6763 \\ -0.5522 \\ -0.3905 \\ -0.2021 \end{pmatrix}$$

and thus we get

$$\mathbf{c} = \begin{pmatrix} -0.6205 \\ -0.3994 \\ -0.3652 \\ -0.2489 \\ -0.1308 \end{pmatrix}$$

For the complete splines we get

$$T = \pi/12 \begin{pmatrix} 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 \end{pmatrix}$$

and

$$\mathbf{b} = \begin{pmatrix} -0.3905 \\ -0.7543 \\ -0.6763 \\ -0.5522 \\ -0.3905 \\ -0.2021 \\ -0.0342 \end{pmatrix}$$

The solution is

$$\mathbf{c} = \begin{pmatrix} -0.5029 \\ -0.4857 \\ -0.4355 \\ -0.3556 \\ -0.2514 \\ -0.1301 \\ -0.0002 \end{pmatrix}$$

Both the natural and complete splines have the following property.

Theorem 1 *Let f be any twice continuously differentiable function such that f satisfies the interpolation condtion and*

$$f''(x_0) = f''(x_n) = 0. \tag{5}$$

Then if s is the cubic spline satisfying the same conditions then

$$\int_a^b [f''(x)]^2 dx \geq \int_a^b [s''(x)]^2 dx.$$

The same conclusion holds if the condition

$$f'(x_0) = y'_0, \quad f'(x_0) = y'_n$$

is substituted for (5).

This theorem account for the "smooth" interpolation property of splines.

Other interesting endpoint conditions include *second derivative conditions.*

$$s''(x_0) = f''(x_0), \quad s''(x_n) = f''(x_n).$$

Again these assume some knowledge of the second derivative. That knowledge is often not available.

A useful spline if no endpoint information is available is the *not-a-knot* spline assumption. This assumes third derivative continuity near the boundary. This assumption is just

$$s_0'''(x_1) = s_1'''(x_1), \quad s_{n-2}'''(x_{n-1}) = s_{n-1}'''(x_{n-1}).$$