TechNote TN-05-001

Natural Cubic Splines

Jim Armstrong
Singularity
October 2005

This is the first in a series of TechNotes on the subject of applied curve mathematics in Adobe Flash™. Each TechNote provides a mathematical foundation for a set of Actionscript examples.

Splines

Following is an informal derivation of the equations governing the generation of a natural cubic spline. A cubic spline interpolates n data points (sometimes called knots),

$$(t_1, y_1), \quad (t_2, y_2), \quad \ldots \quad (t_n, y_n)$$

with a piecewise cubic polynomial. To provide a 'smooth' fit between knots, the spline function, $y = S(x)$ enforces interpolation conditions $S(t_i) = y_i$ as well as continuous first and second derivatives at interior knots. The value of the second derivative at the endpoints is arbitrary. A *natural* cubic spline sets second derivatives to zero at the endpoints. The convention followed in this derivation (and the associated Actionscript code) is that the knots represent non-overlapping intervals.

The general form of the cubic spline is

$$S(x) = \begin{cases} S_1(x) & t_1 \le x \le t_2 \\ S_2(x) & t_2 \le x \le t_3 \\ \vdots & \vdots \\ S_{n-1}(x) & t_{n-1} \le x \le t_n \end{cases}$$

where each $S_i(x)$ is a cubic polynomial. Evaluating the spline at any point, x, in $[t_1, t_n]$ requires the coefficients of the cubic polynomial in that interval. Define $z_i = S''(t_i)$. Because the second derivatives are continuous, these values are uniquely determined. Given the z_i, polynomial coefficients are generated by enforcing interpolation conditions, continuity of first derivative, and endpoint conditions for the second derivative.

If each polynomial is cubic, the second derivative is a linear function between t_i and t_{i+1}. The second derivative passes through the points z_i and z_{i+1} with slope, $h_i^{-1}(z_{i+1} - z_i)$ where $h_i = t_{i+1} - t_i$. For x in $[t_i, t_{i+1}]$, the equation of the line is

$$S_i''(x) = z_i + h_i^{-1}(z_{i+1} - z_i)(x - t_i) = h_i^{-1} z_{i+1}(x - t_i) - h_i^{-1} z_i (x - t_i) + z_i$$

Since $h_i = (x - t_i) + (t_{i+1} - x)$, the quantity $(x - t_i)$ can be expressed as $h_i - (t_{i+1} - x)$ and used to simplify

$$-h_i^{-1} z_i (x - t_i) + z_i = -h_i^{-1} z_i (h_i - (t_{i+1} - x)) + z_i = -z_i + h_i^{-1} z_i (t_{i+1} - x) + z_i$$

which yields

$$S_i^{"}(x) = \frac{z_{i+1}}{h_i}(x_i - t_i) + \frac{z_i}{h_i}(t_{i+1} - x) \qquad [1]$$

This simplification makes the follow-on integration easier. Integrate equation [1] twice to obtain

$$S_i(x) = \frac{z_{i+1}}{6h_i}(x - t_i)^3 + \frac{z_i}{6h_i}(t_{i+1} - x)^3 + ax + b \qquad [2]$$

Where a and b are constants of integration that can be rearranged to the more convenient form

$$S_i(x) = \frac{z_{i+1}}{6h_i}(x - t_i)^3 + \frac{z_i}{6h_i}(t_{i+1} - x)^3 + A(x - t_i) + B(t_{i+1} - x)$$

To compute A and B, apply the interpolation conditions

$$S_i(t_i) = y_i \quad , \quad S_i(t_{i+1}) = y_{i+1}$$

and a lot of algebra to yield

$$S_i(x) = \frac{z_{i+1}}{6h_i}(x - t_i)^3 + \frac{z_i}{6h_i}(t_{i+1} - x)^3 + \left(h_i^{-1} y_{i+1} - \frac{h_i}{6} z_{i+1}\right)(x - t_i) + \left(h_i^{-1} y_i - \frac{h_i}{6} z_i\right)(t_{i+1} - x)$$

This gives each cubic polynomial in terms of already known information and the z_i.

The z_i can be computed by enforcing continuity of the first derivative at interior knots, $t_i, i \le 2 \le n-1$.

Differentiate equation [2] to obtain

$$S_i^{'}(x) = \frac{z_{i+1}}{2h_i}(x - t_i)^2 - \frac{z_i}{2h_i}(t_{i+1} - x)^2 + h_i^{-1} y_{i+1} - \frac{h_i}{6} z_{i+1} - h_i^{-1} y_i + \frac{h_i}{6} z_i \qquad [3]$$

Continuity of the first derivative requires

$S'_{i-1}(t_i) = S'_i(t_i)$

Set $x = t_i$ in equation [3] and recall that $h_i = t_{i+1} - t_i$ to obtain

$$S'_i(t_i) = -\frac{z_i}{2h_i}(t_{i+1} - t_i)^2 + h_i^{-1} y_{i+1} - \frac{h_i}{6} z_{i+1} - h_i^{-1} y_i + \frac{h_i}{6} z_i = \frac{-h_i z_i}{2} + \frac{h_i}{6} z_i - \frac{h_i}{6} z_{i+1} + h_i^{-1}(y_{i+1} - y_i)$$

or

$$\begin{aligned} b_i &= h_i^{-1}(y_{i+1} - y_i) \\ S'_i(t_i) &= -\frac{h_i}{6} z_{i+1} - \frac{h_i}{3} z_i + b_i \end{aligned} \qquad [4]$$

Similarly,

$$S'_{i-1}(t_i) = \frac{h_{i-1}}{6} z_{i-1} + \frac{h_{i-1}}{3} z_i + b_{i-1} \qquad [5]$$

Set equation [4] equal to equation [5] to enforce continuity of first derivative at interior points.

$$-h_i z_{i+1} - 2h_i z_i + 6b_i = h_{i-1} z_{i-1} + 2h_{i-1} z_i + 6b_{i-1}$$

$$-h_i z_{i+1} - (2h_i + 2h_{i-1}) z_i - h_{i-1} z_{i-1} = 6(b_{i-1} - b_i)$$

$$h_{i-1} z_{i-1} + 2(h_i + h_{i-1}) z_i + h_i z_{i+1} = 6(b_i - b_{i-1}) \qquad [6]$$

Equation [6] can be written as a tridiagonal system of equations,

$$\begin{aligned} u_i &= 2(h_{i-1} + h_i) \\ v_i &= 6(b_i - b_{i-1}) \end{aligned}$$

$$\begin{aligned} z_1 &= 0 \\ h_{i-1} z_{i-1} + u_i z_i + h_i z_{i+1} &= v_i \quad , i \le 2 \le n-1 \\ z_n &= 0 \end{aligned}$$

The first and last equations are trivial, so the equations for the unknown z_i can be written in the form,

$$\begin{bmatrix} u_2 & h_2 & & \\ h_2 & u_3 & \ddots & \\ & \ddots & \ddots & h_{n-2} \\ & & h_{n-2} & u_{n-1} \end{bmatrix} \begin{bmatrix} z_2 \\ z_3 \\ \vdots \\ z_{n-1} \end{bmatrix} = \begin{bmatrix} v_2 \\ v_3 \\ \vdots \\ v_{n-1} \end{bmatrix} \qquad [7]$$

which is symmetric and tridiagonal. As it happens, the main diagonal elements in [7] are greater than zero, so the system of equations can be solved using Gaussian elimination without pivoting. This yields simple and fast recurrence relations for both the forward elimination and back-substitution phases.

There are computational advantages to writing the cubic spline equation in the form

$$S_i(x) = A_i + B_i(x - t_i) + C_i(x - t_i)^2 + D_i(x - t_i)^3 \qquad [8]$$

If $\delta = x - t_i$, [8] can be rewritten as

$$S_i(t_i + \delta) = A_i + B_i\delta + C_i\delta^2 + D_i\delta^3 \qquad [9]$$

Eq. [9] is the third-orderTaylor-series expansion of S_i about the point, t_i, meaning

$$\begin{aligned} A_i &= S_i(t_i) \\ B_i &= S_i^{'}(t_i) \\ C_i &= \frac{1}{2} S_i^{''}(t_i) \\ D_i &= \frac{1}{6} S_i^{'''}(t_i) \end{aligned}$$

A_i follows easily from the interpolation conditions. From eq. [1], $S_i^{'}(t_i) = z_i \Rightarrow C_i = \dfrac{z_i}{2}$

An easier way to determine D_i than computing a third derivative is to examine the coefficient of the x^3 term in the original equation for $S_i(x)$,

$$S_i(x) = \frac{z_{i+1}}{6h_i}(x - t_i)^3 + \frac{z_i}{6h_i}(t_{i+1} - x)^3 + \left(h_i^{-1} y_{i+1} - \frac{h_i}{6} z_{i+1}\right)(x - t_i) + \left(h_i^{-1} y_i - \frac{h_i}{6} z_i\right)(t_{i+1} - x)$$

Since this equation is supposed to equal [9],

$$D_i = \frac{1}{6h_i}(z_{i+1} - z_i)$$

From eq. [4], $S_i^{'}(t_i) = -\dfrac{h_i}{6} z_{i+1} - \dfrac{h_i}{3} z_i + b_i$, leaving the final set of equations

$$A_i = y_i$$
$$B_i = \frac{-h_i}{6} z_{i+1} - \frac{h_i}{3} z_i + b_i$$
$$C_i = \frac{z_i}{2}$$
$$D_i = \frac{1}{6}(z_{i+1} - z_i)$$

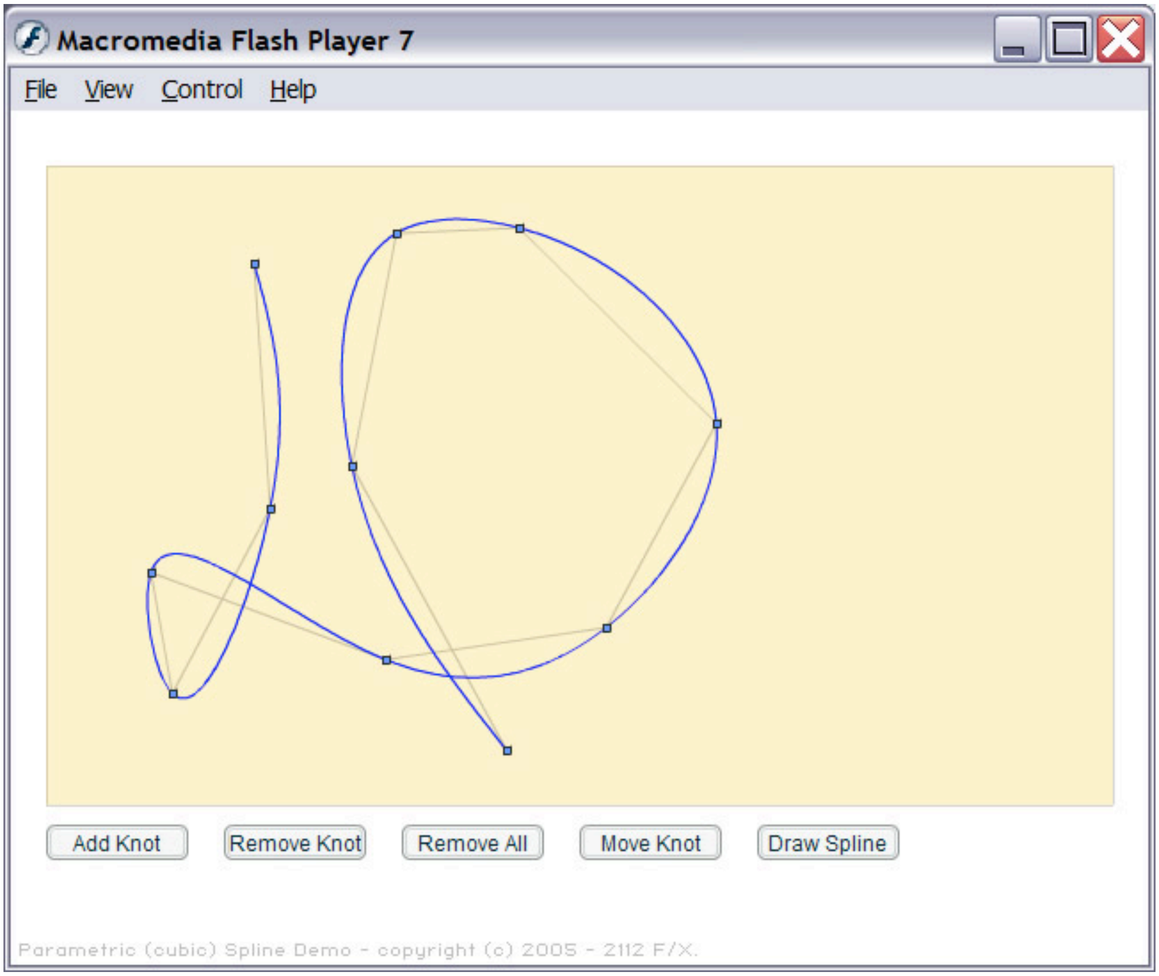
This allows the spline equation to be written in the nested form,

$$S_i(x) = y_i + \delta\left(B_i + \partial\left(\frac{z_i}{2} + \frac{\partial}{6h_i}(z_{i+1} - z_i)\right)\right)$$

It is possible to extrapolate with a cubic spline, using S_1 for $x\varepsilon[-\infty, t_1]$ and S_{n-1} for $x\varepsilon[t_n, \infty]$. Given the arbitrary conditions imposed on the second derivative at t_1 and t_n, results of extrapolation are unpredictable. The cubic spline function possesses several attractive smoothness properties when interpolating in $[t_1, t_n]$, so its primary (and best) use is for interpolation.

Parametric Splines

The cubic spline can be parameterized on t in [0,1], allowing interpolation with overlapping intervals. There are several applications for such splines, including mathematical typography. A parametric cubic spline was used to create the script letter D, as shown below.



This approach is useful when it is necessary to interpolate a specific set of points. Other types of splines (and Bezier curves) are used in mathematical typography. Knuth [2] is a useful starting reference.

References:

1) De Boor, C. “A Practical Guide to Splines”, Applied Mathematical Sciences 27, Springer.

2) Knuth, D.E., “Mathematical Typography,” Bulletin AMS 2, 337-372.