Basics of Cubic Spline Fitting

1 Preliminaries

A spline is a piecewise polynomial approximation of a set of points $\{(x_0, f(x_0), (x_1, f(x_1), ..., (x_{n-1}, f_{n-1})\}$. The simplest approximation is just doing a polynomial regression on successive pairs of elements, treating the range between each pair of x-coordinates as a part of the piecewise function. The most common spline approximation is cubic spline fitting. The cubic spline approximation guarantees to correctly fit all y-coordinates to the given x-coordinates, but does not guarantee that the derivative of the fitted curve equals the derivative of the original function.

Given a function $f : [a, b]$ and a set of nodes $a = x_0 < x_1 < x_2 < ... < x_{n-1} = b$, a CUBIC SPINE INTERPOLANT S of function f satisfies the following properties:

1. $S(x)$ is a cubic piecewise polynomial; each part of the piecewise function is denoted as S_j for interval $[x_j, x_{j+1}]$ and $0 \leq j < n - 1$.

2. $S_j(x_j) = f(x_j); S_j(x_{j+1}) = f(x_{j+1})$ for $0 \leq j < n - 1$.

3. Following from (2), $S_j(x_{j+1}) = S_{j+1}(x_{j+1})$ (Continuous property)

4. $S'_j(x_{j+1}) = S'_{j+1}(x_{j+1})$ (Differentiable property)

5. $S''_j(x_{j+1}) = S''_{j+1}(x_{j+1})$ (Also twice differentiable)

6. One of the following sets of boundary conditions is satisfied:

   (a) Natural/free boundary: $S(x_0) = S(x_n) = 0$

   (b) Clamped boundary: $S(x_0) = f(x_0)$; $S(x_n) = f(x_n)$.

2 Constructing $S_j(x)$

Given the definition of $S(x)$ above, the general form of $S_j(x)$ is:

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

For all $0 \leq j < n - 1$. Now since

$$\forall 0 \leq j < n - 1, S_j(x) = a_j = f(x_j)$$

We know

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

For $0 \leq j < n - 2$. We thus establishe a correlation between the successive terms of S_j.

Now let $h_j = x_{j+1} - x_j$, then we can rewrite the formula above into a more compact form:

$$a_{j+1} = a_j + b_j h_j + c_j h_j^2 + d_j h_j^3$$

This compact form holds for $0 \leq j < n - 1$.

To proceed further, $S'_j(x)$ is defined as:

$$S'_j(x) := \frac{dS_j(x)}{dx} = b_j + 2c_j(x - x_j) + 3d_j(x - x_j)^2$$

The above holds for $0 \leq j < n-2$. Based on property 4 (the differentiability property) of S as defined in section 1:

$$S'_{j+1}(x_{j+1}) = S'_j(x_{j+1}) = b_j + 2c_j(x_{j+1} - x_j) + 3d_j(x_{j+1} - x_j)^2 = b_j + 2c_j h_j + 3d_j h_j^2$$

Now define $b_j = S'(x_j)$. Then:

$$b_{j+1} = b_j + 2c_j h_j + 3d_j h_j^2$$

Finally, based on the twice-differentiable property (property 5) defined in section 1:

$$S_j''(x) := \frac{dS_j'(x)}{dx} = 2c_j + 6d_j(x - x_j)$$

(Which also holds for $0 \leq j < n-1$), and:

$$S_{j+1}''(x_{j+1}) = S_j''(x_{j+1}) = 2c_j + 6d_j(x_{j+1} - x_j) = 2c_j + 6d_j h_j$$

Define $c_j = S''(x_j)/2$. Further simplifying:

$$c_{j+1} = c_j + 3d_j h_j$$

The following is a table to summarize all relationships:

Function	Definition	Simplifying Variable	Simplified Relationship
S_j	$a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$	$h_j = x_{j+1} - x_j$	$a_{j+1} = a_j + b_j h_j + c_j h_j^2 + d_j h_j^3$
S_j'	$S_j'(x) := \frac{dS_j(x)}{dx} = b_j + 2c_j(x - x_j) + 3d_j(x - x_j)^2$	$b_j = S'(x_j)$	$b_{j+1} = b_j + 2c_j h_j + 3d_j h_j^2$
S_j''	$S_j''(x) := \frac{dS_j'(x)}{dx} = 2c_j + 6d_j(x - x_j)$	$c_j = S''(x_j)/2$	$c_{j+1} = c_j + 3d_j h_j$

2.1 Further derivation

If we solve for d_j using the simplified relationship from the second derivative of S_j, we get:

$$d_j = \frac{1}{3h_j}(c_{j+1} - c_j)$$

If we substitute this into the simplified relationships for S_j and S_j'':

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

$$b_{j+1} = b_j + 2c_j h_j + (c_{j+1} - c_j)h_j = b_j + h_j(c_{j+1} + c_j)$$

Solve the a_{j+1} equation for b_j yields:

$$
\begin{aligned}
a_{j+1} - a_j - h_j^2(c_j + c_{j+1}/3 - c_j/3) &= b_j h_j \\
a_{j+1} - a_j - \frac{1}{3}(2c_j + c_{j+1})h_j^2 &= b_j h_j \\
\frac{1}{h_j}(...) &= b_j \\
b_j &= \frac{1}{h_j}(a_{j+1} - a_j) - \frac{h_j}{3}(2c_j + c_{j+1})
\end{aligned}
$$

Which means:

$$
b_{j-1} = \frac{1}{h_{j-1}}(a_j - a_{j-1}) - \frac{h_{j-1}}{3}(3c_{j-1} + c_j)
$$

Now substitute this representation of b_j into the second equation above phrased in terms of b_{j+1} yields:

$$
b_{j+1} = b_j + h_j(c_{j+1} + c_j)
$$

$$
b_j = b_{j-1} + h_{j-1}(c_{j-2} + c_{j-1})
$$

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

Simplifying we get:

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

Note how we eliminated b completely, and a, h values are given as inputs. Therefore the solution towards cubic spline approximation is now reduced to solving the linear system of equations involving $\{c\}_{j=1}^n$ as unknowns.

3 Where to go further from here?

(Burden & Faires[1]) provides a few theorems that formally proves that $S(x)$ can be generated from the system of linear equations, and a pseudocode for spline generation. Mind the gaps!

[1] Num. Analysis, 9th edition: https://fac.ksu.edu.sa/sites/default/files/numerical_analysis_9th.pdf