

# Interpolation Analytics in Curve Construction

This article focuses on interpolation subject, in particular, the interpolation of curve bootstrapping. Both linear spline and cubic spline are studied. Although there are a number of advantages to using piecewise cubic splines, there is one major drawback which leads us to go in favour of linear splines. This drawback stems from the fact that the perturbation of one point will affect another point.

We will show a graphical illustration later of how the perturbation of one point will affect other points. As a result, we recommend that linear splines be used to construct the various curves rather than higher order polynomial splines.

## Linear Spline

Given a set of discrete data points  $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ , one can use linear interpolation between successive points,

$$s_k(x) = y_k + d_k(x - x_k) \quad \text{where} \quad d_k = \frac{y_{k+1} - y_k}{x_{k+1} - x_k}$$

to build a piecewise linear interpolant of the data points  $(x_k, y_k)$ :

$$s(x) = \begin{cases} y_0 + d_0(x - x_0) & x \in [x_0, x_1] \\ y_1 + d_1(x - x_1) & x \in [x_1, x_2] \\ \vdots & \vdots \\ y_{n-1} + d_{n-1}(x - x_{n-1}) & x \in [x_{n-1}, x_n] \end{cases}$$

One can then use this to approximate other points on the curve. The advantage of linear interpolation is its simplicity and, in many cases, it provides an adequate approximation. A disadvantage is that the approximating curve is not smooth (since the derivative is in general discontinuous at given data points) even though the real curve may in fact be smooth.

## Cubic Spline

Another method of interpolation is to fit a series of linked cubic polynomials. Consider an interpolant that allows curvature between the data points and requires the first and second derivatives to be continuous throughout the interval  $[x_0, x_n]$ . Furthermore, assume that the interpolant linking the data points  $(x_k, y_k)$  and  $(x_{k+1}, y_{k+1})$  is a cubic polynomial of the form:

$$s_k(x) = s_{k,0} + s_{k,1}(x - x_k) + s_{k,2}(x - x_k)^2 + s_{k,3}(x - x_k)^3 \quad x \in [x_k, x_{k+1}]$$

subject to the following constraints:

$$\begin{aligned} s_k(x_k) &= y_k \\ s_k(x_{k+1}) &= s_{k+1}(x_{k+1}) \\ s'_k(x_{k+1}) &= s'_{k+1}(x_{k+1}) \\ s''_k(x_{k+1}) &= s''_{k+1}(x_{k+1}) \end{aligned}$$

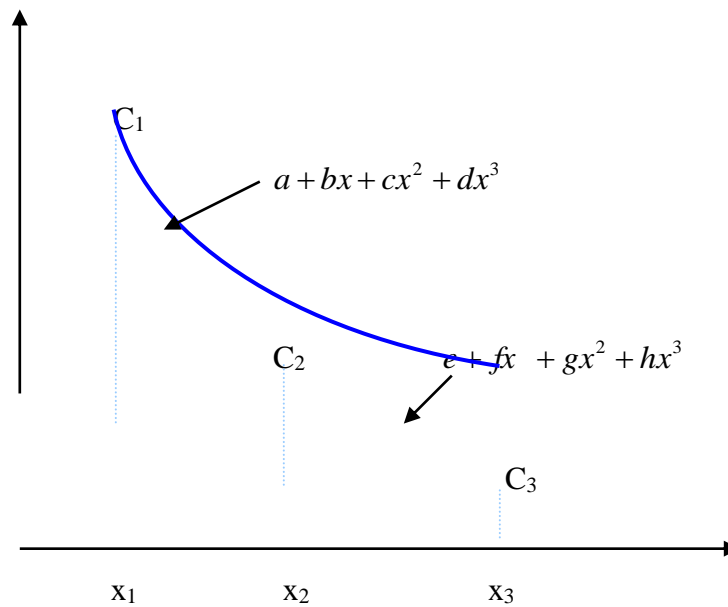
The first constraint is the interpolation of  $(x_k, y_k)$ ; the second constraint is the continuity of the interpolant and the third and fourth constraints are the continuity of the first and second derivatives respectively. A final condition is to set the change in slope at the ends of the curve equal to zero (for a natural cubic spline).

Hence, with cubic splines, the curve has to go through each data point but neither the curve nor its slope can be kinked at any point. The gap between each data point is represented by a unique polynomial that relates to the next “piece” in a chain by having

its slope and rate of change of the slope be equal to that of its neighbour at the point at which they join.

A simple example below shows how three data points are linked (at  $x_1, x_2, x_3$ ) with two cubic splines.

### Cubic Spline Interpolation



**Each section of the curve is a cubic polynomial**

$$C_1 = a + bx_1 + cx_1^2 + dx_1^3$$

$$C_2 = a + bx_2 + cx_2^2 + dx_2^3 \quad \text{and} \quad C_2 = e + fx_2 + gx_2^2 + hx_2^3$$

$$C_3 = e + fx_3 + gx_3^2 + hx_3^3$$

**First and second derivatives are equal where curves join:**

$$b + 2cx_2 + 3dx_2^2 = f + 2gx_2 + 3hx_2^2$$

$$2c + 6dx_2 = 2g + 6hx_2$$

**Two extra equations – change in slope at the ends = 0:**

$$2c + 6dx_2 = 0$$

$$2g + 6hx_2 = 0$$

Reference:

<https://finpricing.com/lib/FxTarn.html>