

Constrained Cubic Spline Interpolation for Chemical Engineering Applications

by CJC Kruger

Summary

Cubic spline interpolation is a useful technique to interpolate between known data points due to its stable and smooth characteristics. Unfortunately it does not prevent overshoot at intermediate points, which is essential for many chemical engineering applications. This article presents a new interpolation method that combines the smooth curve characteristics of spline interpolation, with the non-overshooting behaviour of linear interpolation.

Introduction

Interpolation is used to estimate the value of a function between known data points without knowing the actual function. Interpolation methods can be divided into two main categories [1,2]:

- Global interpolation. These methods rely on a constructing single equation that fits all the data points. This equation is usually a high degree polynomial equation. Although these methods result in smooth curves, they are usually not well suited for engineering applications, as they are prone to severe oscillation and overshoot at intermediate points.
- Piecewise interpolation. These methods rely on constructing a polynomial of low degree between each pair of known data points. If a first degree polynomial is used, it is called linear interpolation. For second and third degree polynomials, it is called quadratic and cubic splines respectively. The higher the degree of the spline, the smoother the curve. Splines of degree m , will have continuous derivatives up to degree $m-1$ at the data points.

Linear interpolation result in straight line between each pair of points and all derivatives are discontinuous at the data points. As it never overshoots or oscillates, it is frequently used in chemical engineering despite the fact that the curves are not smooth.

To obtain a smoother curve, cubic splines are frequently recommended. They are generally well behaved and continuous up to the second order derivative at the data points. Even though cubic splines are less prone to oscillation or overshoot than global polynomial equations, they do not prevent it. Thus, the use of cubic splines in chemical engineering is limited to applications where oscillation and overshoot are acceptable or desirable.

Traditional Cubic Splines

Consider a collection of known points $(x_0, y_0), (x_1, y_1), \dots (x_{i-1}, y_{i-1}), (x_i, y_i), (x_{i+1}, y_{i+1}), \dots (x_n, y_n)$. To interpolate between these data points using traditional cubic splines, a third degree polynomial is constructed between each point. The equation to the left of point (x_i, y_i) is indicated as f_i with a y value of $f_i(x_i)$ at point x_i . Similarly, the equation to the right of point (x_i, y_i) is indicated as f_{i+1} with a y value of $f_{i+1}(x_i)$ at point x_i .

Traditionally the cubic spline function, f_i , is constructed based on the following criteria:

- Curves are third order polynomials,

$$f_i(x) = a_i + b_i x + c_i x^2 + d_i x^3 \quad - (1)$$

- Curves pass through all the known points,

$$f_i(x_i) = f_{i+1}(x_i) = y_i \quad - (2)$$

- The slope, or first order derivative, is the same for both functions on either side of a point,

$$f'_i(x_i) = f'_{i+1}(x_i) \quad - (3)$$

- The second order derivative is the same for both functions on either side of a point,

$$f''_i(x_i) = f''_{i+1}(x_i) \quad - (4)$$

This results in a matrix of n-1 equations and n+1 unknowns. The two remaining equations are based on the border conditions for the starting point, $f_1(x_0)$, and end point, $f_n(x_n)$. Historically one of the following border conditions have been used [1,2,3]:

- Natural splines. The second order derivative of the splines at the end points are zero.

$$f''_1(x_0) = f''_n(x_n) = 0 \quad - (5a)$$

- Parabolic runout splines. The second order derivative of the splines at the end points are the same as at the adjacent points. The result is that the curve becomes a parabolic curve at the end points.

$$\begin{aligned} f''_1(x_0) &= f''_1(x_1) \\ f''_n(x_n) &= f''_n(x_{n-1}) \end{aligned} \quad - (5b)$$

- Cubic runout splines. The curve degrades to a single cubic curve over the last two intervals by setting the second order derivative of the splines at the end points to:

$$\begin{aligned} f''_1(x_0) &= 2f''_1(x_1) - f''_2(x_2) \\ f''_n(x_n) &= 2f''_n(x_{n-1}) - f''_{n-1}(x_{n-2}) \end{aligned} \quad - (5c)$$

- Clamped spline. The first order derivative of the splines at the end points are set to known values.

$$\begin{aligned} f'_1(x_0) &= f'(x_0) \\ f'_n(x_n) &= f'(x_n) \end{aligned} \quad - (5d)$$

In traditional cubic splines equations 2 to 5 are combined and the n+1 by n+1 tridiagonal matrix is solved to yield the cubic spline equations for each segment [1,3]. As both the first and second order derivative for connecting functions are the same at every point, the result is a very smooth curve.

Even though traditional cubic splines are well behaved for many applications, it does not prevent overshoot at intermediate points. This is illustrated in Figures 1 and 2, where a natural cubic spline is fitted to hypothetical and somewhat unusual distillation and pump curves. Clearly this behaviour is unacceptable for chemical engineering applications, and the engineer has little choice but to revert back to linear interpolation.

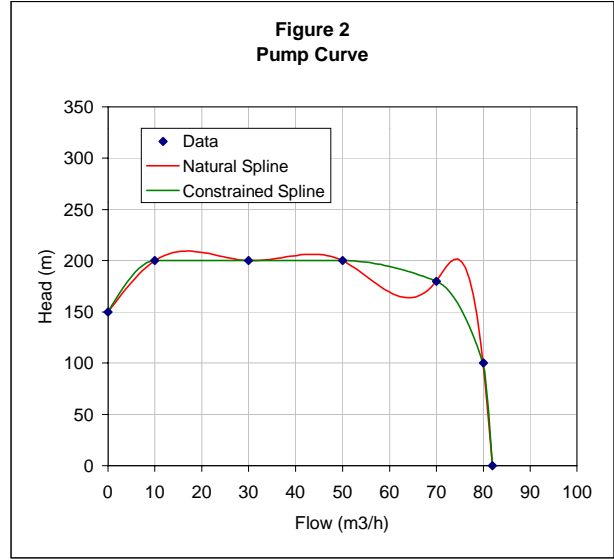
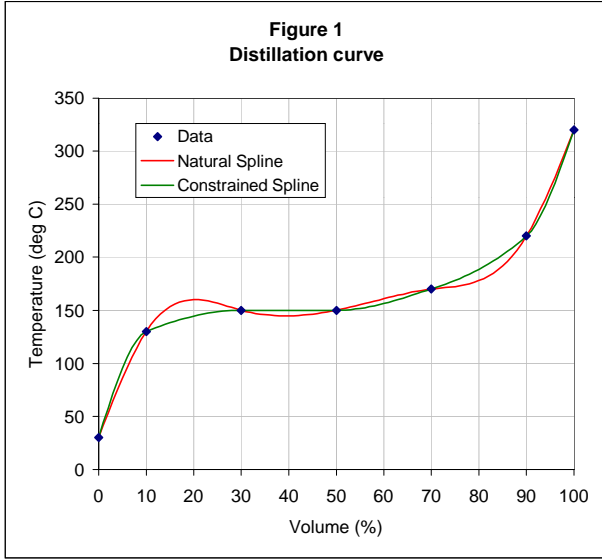
Proposed Constrained Cubic Splines

The principle behind the proposed constrained cubic spline is to prevent overshooting by sacrificing smoothness. This is achieved by eliminating the requirement for equal second order derivatives at every point (equation 4) and replacing it with specified first order derivatives.

Thus, similar to traditional cubic splines, the proposed constrained cubic splines are constructed according to equations (2), (3) and (5a). Equation (4) is replaced by,

- A specified first order derivative, or slope, at every point,

$$f'_i(x_i) = f'_{i+1}(x_i) = f'(x_i) \quad - (6)$$



The key step becomes the calculation of the slope at each point. Intuitively we know the slope will be between the slopes of the adjacent straight lines, and should approach zero if the slope of either line approaches zero.

A relatively simple equation that works well and satisfies these requirements, is:

$$f'(x_i) = \frac{2}{\frac{x_{i+1} - x_i}{y_{i+1} - y_i} + \frac{x_i - x_{i-1}}{y_i - y_{i-1}}} \quad - (7a)$$

= 0 if slope changes sign at point

Equation (7a) is only valid for intermediate points. The slope at the end points is based on rewriting equation (5a) to yield,

$$f'_1(x_0) = \frac{3(y_1 - y_0)}{2(x_1 - x_0)} - \frac{f'(x_1)}{2} \quad - (7b)$$

$$f'_n(x_n) = \frac{3(y_n - y_{n-1})}{2(x_n - x_{n-1})} - \frac{f'(x_{n-1})}{2} \quad - (7c)$$

As the slope at each point is known, it is no longer necessary to solve a system of equations. Each spline function, as given by equation (1), can be calculated based on the two adjacent points on each side. This is summarized in equations (8) to (13) below.

$$f''_i(x_{i-1}) = -\frac{2[f'_i(x_i) + 2f'_i(x_{i-1})]}{(x_i - x_{i-1})} + \frac{6(y_i - y_{i-1})}{(x_i - x_{i-1})^2} \quad - (8)$$

$$f''_i(x_i) = \frac{2[2f'_i(x_i) + f'_i(x_{i-1})]}{(x_i - x_{i-1})} - \frac{6(y_i - y_{i-1})}{(x_i - x_{i-1})^2} \quad - (9)$$

$$d_i = \frac{f''_i(x_i) - f''_i(x_{i-1})}{6(x_i - x_{i-1})} \quad - (10)$$

$$c_i = \frac{x_i f''_i(x_{i-1}) - x_{i-1} f''_i(x_i)}{2(x_i - x_{i-1})} \quad - (11)$$

$$b_i = \frac{(y_i - y_{i-1}) - c_i(x_i^2 - x_{i-1}^2) - d_i(x_i^3 - x_{i-1}^3)}{(x_i - x_{i-1})} \quad - (12)$$

$$a_i = y_{i-1} - b_i x_{i-1} - c_i x_{i-1}^2 - d_i x_{i-1}^3 \quad - (13)$$

The behaviour of the proposed constrained cubic spline is shown in Figures 1 and 2. In general it fits chemical engineering needs well in cases where oscillation or overshoot cannot be tolerated.

An Excel Visual Basic for Applications (VBA) example of this technique can be obtained from www.korf.co.uk. This interpolation method will also be used in the next release of Korf Hydraulics, which is a flexible and user-friendly piping network solver available from the same website.

Conclusions

A modified cubic spline interpolation method has been developed for chemical engineering application. The main benefits of the proposed constrained cubic spline are:

- It is a relatively smooth curve;
- It never overshoots intermediate values;
- Interpolated values can be calculated directly without solving a system of equations;
- The actual parameters (a_i , b_i , c_i and d_i) for each of the cubic spline equations can still be calculated. This permits analytical integration of the data.

Example

The hypothetical distillation curve in Figure 1 is represented by the following data points:

Point	0	1	2	3	4	5	6
(x,y)	(0,30)	(10,130)	(30,150)	(50,150)	(70,170)	(90,220)	(100,320)

Calculate the cubic equations for the first two segments.

First segment, $i=1$, for $0 \leq x \leq 10$:

$$\begin{aligned} f_1(x_1) &= 2/((x_2 - x_1)/(y_2 - y_1) + (x_1 - x_0)/(y_1 - y_0)) \\ &= 2/((30 - 10)/(150 - 130) + (10 - 0)/(130 - 30)) \\ &= 1.8181 \end{aligned} \quad - (7a)$$

$$\begin{aligned} f_1(x_0) &= 3/2*(y_1 - y_0)/(x_1 - x_0) - f_1(x_1)/2 \\ &= 3/2*(130 - 30)/(10 - 0) - 1.818/2 \\ &= 14.0909 \end{aligned} \quad - (7b)$$

$$\begin{aligned} f''_1(x_0) &= -2*(f_1(x_1) + 2*f_1(x_0))/(x_1 - x_0) + 6*(y_1 - y_0)/(x_1 - x_0)^2 \\ &= -2*(1.8181 + 2*14.0909)/(10 - 0) + 6*(130 - 30)/(10 - 0)^2 \\ &= 0 \end{aligned} \quad - (8)$$

$$\begin{aligned} f''_1(x_1) &= 2*(2*f_1(x_1) + f_1(x_0))/(x_1 - x_0) - 6*(y_1 - y_0)/(x_1 - x_0)^2 \\ &= 2*(2*1.8181 + 14.0909)/(10 - 0) - 6*(130 - 30)/(10 - 0)^2 \\ &= -2.4545 \end{aligned} \quad - (9)$$

$$\begin{aligned} d_1 &= 1/6 * (f''_1(x_1) - f''_1(x_0))/(x_1 - x_0) \\ &= 1/6 * (-2.4545 - 0)/(10 - 0) \\ &= -0.0409 \end{aligned} \quad - (10)$$

$$\begin{aligned} c_1 &= 1/2 * (x_1*f''_1(x_0) - x_0*f''_1(x_1))/(x_1 - x_0) \\ &= 1/2 * (10*0 - 0*1.8181)/(10 - 0) \\ &= 0 \end{aligned} \quad - (11)$$

$$\begin{aligned} b_1 &= ((y_1 - y_0) - c_1*(x_1^2 - x_0^2) - d_1*(x_1^3 - x_0^3))/(x_1 - x_0) \\ &= ((130 - 30) - 0*(10^2 - 0^2) + 0.0409*(10^3 - 0^3))/(10 - 0) \\ &= 14.09 \end{aligned} \quad - (12)$$

$$\begin{aligned} a_1 &= y_0 - b_1*x_0 - c_1*x_0^2 - d_1*x_0^3 \\ &= 30 \end{aligned} \quad - (13)$$

$$y_1 = 30 + 14.09x - 0.0409x^3 \quad \text{for } 0 \leq x \leq 10$$

Second segment, $i=2$, for $10 \leq x \leq 30$:

$$\begin{aligned} f_2(x_2) &= 2/((x_3 - x_2)/(y_3 - y_2) + (x_2 - x_1)/(y_2 - y_1)) \\ &= 2/((50 - 30)/(150 - 150) + (30 - 10)/(150 - 130)) \end{aligned} \quad - (7a)$$

$$\begin{aligned}
&= 0 \\
f_2(x_1) &= 2/((x_2 - x_1)/(y_2 - y_1) + (x_1 - x_0)/(y_1 - y_0)) & - (7a) \\
&= 1.8181 \\
f''_2(x_1) &= -2*(f_2(x_2) + 2*f_2(x_1))/(x_2 - x_1) + 6*(y_2 - y_1)/(x_2 - x_1)^2 & - (8) \\
&= -2*(0 + 2*1.8181)/(30 - 10) + 6*(150 - 130)/(30 - 10)^2 \\
&= -0.063636 \\
f''_2(x_2) &= 2*(2*f_2(x_2) + f_2(x_1))/(x_2 - x_1) - 6*(y_2 - y_1)/(x_2 - x_1)^2 & - (9) \\
&= 2*(2*0 + 1.8181)/(30 - 10) - 6*(150 - 130)/(30 - 10)^2 \\
&= -0.11818 \\
d_2 &= 1/6 * (f''_2(x_2) - f''_2(x_1))/(x_2 - x_1) & - (10) \\
&= 1/6 * (-0.11818 + 0.063636)/(30 - 10) \\
&= -0.0004545 \\
c_2 &= 1/2 * (x_2*f''_2(x_1) - x_1*f''_2(x_2))/(x_2 - x_1) & - (11) \\
&= 1/2 * (-30*0.063636 + 10*0.11818)/(30 - 10) \\
&= -0.01818 \\
b_2 &= ((y_2 - y_1) - c_2*(x_2^2 - x_1^2) - d_2*(x_2^3 - x_1^3))/(x_2 - x_1) & - (12) \\
&= ((150 - 130) + 0.01818*(30^2 - 10^2) + 0.0004545*(30^3 - 10^3))/(30 - 10) \\
&= 2.31818 \\
a_2 &= y_1 - b_2*x_1 - c_2*x_1^2 - d_2*x_1^3 & - (13) \\
&= 130 - 2.31818*10 + 0.01818*10^2 + 0.0004545*10^3 \\
&= 109.09 \\
y_2 &= 109.09 + 2.31818x - 0.01818x^2 - 0.0004545x^3 \quad \text{for } 10 \leq x \leq 30
\end{aligned}$$

References

1. Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P., *Numerical Recipes in Fortran, The Art of Scientific Computing*, Second Edition, Cambridge University Press, Cambridge, Reprinted 1995.
2. Henrici, P., *Essential of Numerical Analysis*, John Wiley & Sons, New York, 1982.
3. McKinley, S. and Levine, M., *Cubic Spline Interpolation*, <http://online.redwoods.cc.ca.us/instruct/darnold/laproj/Fall98/SkyMeg/Proj.PDF>.