Cubic Spline Tutorial

Cubic splines are a popular choice for curve fitting because of ease of data interpolation, integration, differentiation, and they are normally very smooth. This tutorial will describe a computationally efficient method of constructing joined cubic splines through known data points. As background and motivation, we will begin with a discussion of the basic cubic polynomial:

$y = ax^3 + bx^2 + cx + d$ where x is the independent variable and a, b, c, d are real numbers

OK, what's there to know? It was most likely studied in pre-University education, has three roots (real and/or imaginary), the 1st derivative is a quadratic function, the 2nd derivative is a linear function, and it can be fit to four points. Let's dig a little deeper. The general cubic polynomial has four parameters or coefficients that completely define the shape of the curve. Alternatively, we can consider a cubic polynomial as having four constraints or degrees of freedom in physical terms (i.e. points and derivatives) that determine the four parameters. The constraints can be four points, which means the equation can be represented in the matrix form on the left. The solution to the matrix equation is the four polynomial coefficients. However, there are other possibilities. The four constraints can be three points plus a 1st or 2nd derivative at any arbitrary point. Or, the constraints can be two points plus two derivatives, which is actually the basis for cubic spline interpolation.

$$\begin{bmatrix} x_1^3 & x_1^2 & x_1 & 1 \\ x_2^3 & x_2^2 & x_2 & 1 \\ x_3^3 & x_3^2 & x_3 & 1 \\ x_4^3 & x_4^2 & x_4 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

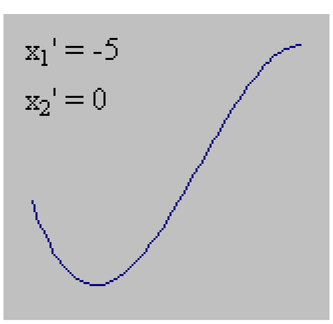
Let's say we wish to represent a cubic equation as two points (x_1, y_1), (x_2, y_2) plus first derivatives at those points (y_1', y_2'). How would that be done? Well, let's first take the derivative of the cubic equation, which is $y' = 3ax^2 + 2bx + c$. Thus, $y_1' = 3ax_1^2 + 2bx_1 + c$ and $y_2' = 3ax_2^2 + 2bx_2 + c$. The matrix equation defining the associated cubic polynomial is on the right. OK. How about two points plus 2nd derivatives at the points? In that case, the equation of the 2nd derivative is needed, which is $y'' = 6ax + 2b$. Thus, $y_1'' = 6ax_1 + 2b$ and $y_2'' = 6ax_2 + 2b$. The matrix equation defining the associated cubic polynomial is on the left. How about two points plus 1st and 2nd derivatives at the first point? The matrix equation for that is on the right. As can be seen, there are multiple ways to formulate a cubic polynomial by specifying points and derivatives.

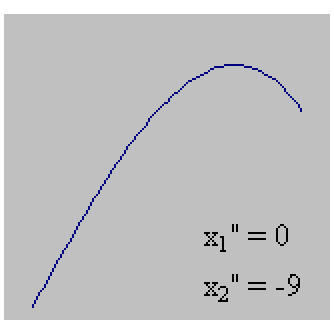
$$\begin{bmatrix} 3x_1^2 & 2x_1 & 1 & 0 \\ x_1^3 & x_1^2 & x_1 & 1 \\ x_2^3 & x_2^2 & x_2 & 1 \\ 3x_2^2 & 2x_2 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} y_1' \\ y_1 \\ y_2 \\ y_2' \end{bmatrix}$$

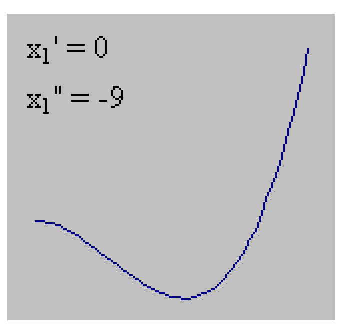
$$\begin{bmatrix} 6x_1 & 2 & 0 & 0 \\ x_1^3 & x_1^2 & x_1 & 1 \\ x_2^3 & x_2^2 & x_2 & 1 \\ 6x_2 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} y_1'' \\ y_1 \\ y_2 \\ y_2'' \end{bmatrix}$$

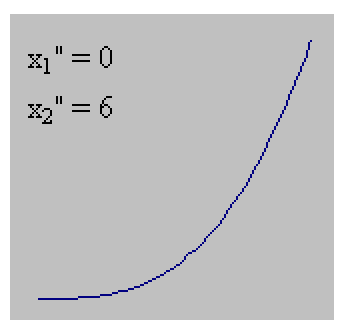
$$\begin{bmatrix} 3x_1^2 & 2x_1 & 1 & 0 \\ x_1^3 & x_1^2 & x_1 & 1 \\ x_2^3 & x_2^2 & x_2 & 1 \\ 6x_1 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} y_1' \\ y_1 \\ y_2 \\ y_1'' \end{bmatrix}$$

To illustrate, choose $p_1 = (0, 0)$ and $p_2 = (1, 1)$. The plots below indicate the result of using various boundary conditions at the end points to define the cubic polynomial.









Note the last plot (on the right). It represents the equation $y = x^3$. We know this because for this equation, $y'' = 6x$. Therefore $y(0)'' = 6(0) = 0$ and $y(1)'' = 6(1) = 6$, which match the constraints used to generate the plot.

Alternate Cubic Polynomial formulation - before moving on, it is useful to consider the following (more general) form of the cubic polynomial:

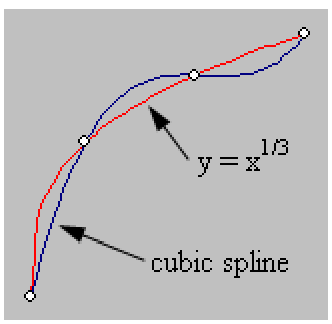
$y = a'(x-x_0)^3 + b'(x-x_0)^2 + c'(x-x_0) + d'$

where x_0 is an arbitrary reference point for the independent variable. The superscript (i.e. ') on a, b, c, d is used to distinguish the coefficients from the previous case where x^n is used instead of $(x-x_0)^n$. This form of a cubic polynomial is just as valid as the previous version and is convenient in cubic spline interpolation because if x_0 is chosen as the value of the independent variable at the beginning of the spline segment, the quantity $(x-x_0)$ begins at zero for each segment and the result is a consistent mathematical treatment for all segments. Also, if the above expression for y is expanded and terms collected, it is easy to show that:

$a = a'$ $\qquad$ $b = b' – 3a'x_0$ $\qquad$ $c = c' + 3a'x_0^2 – 2b'x_0$ $\qquad$ $d = d' – c'x_0 + b'x_0^2 – a'x_0^3$

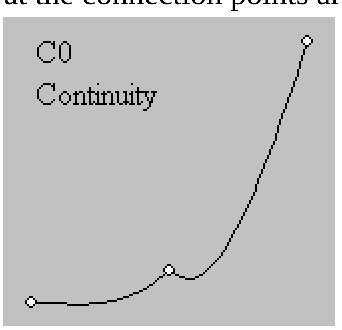
Clearly, if x_0 is chosen as 0, the coefficients (and equations) are identical. As a simple example, consider the equation $y = 2x^3 + 3x^2 + 4x + 5$. If x_0 is chosen as 1, the following equation produces the exact same result: $y = 2(x-1)^3 + 9(x-1)^2 + 16(x-1) + 14$. Checking the result for consistency:

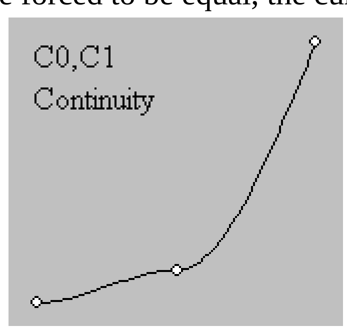
$a' = 2$	$a = a' = 2$
$b' = 9$	$b = b' – 3a'x_0 = 9 – 3(2)(1) = 3$
$c' = 16$	$c = c' + 3a'x_0^2 – 2b'x_0 = 16 + 3(2)(1)^2 – 2(9)(1) = 16 + 6 – 18 = 4$
$d' = 14$	$d = d' – c'x_0 + b'x_0^2 – a'x_0^3 = 14 – (16)(1) + (9)(1)^2 – (2)(1)^3 = 14 – 16 + 9 – 2 = 5$

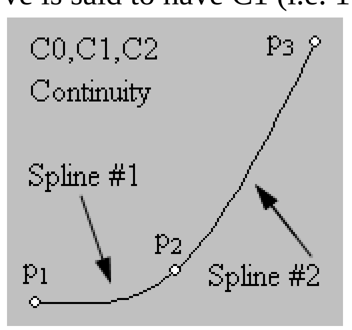


Now that we can think of cubic polynomials in terms of points and derivatives, let's connect two together. Why do that? The main reason is to have a smooth curve that goes through (i.e. interpolates) a given set of points. Of course, it is possible to derive a single polynomial of degree (n-1) to fit n points, but the end result is often not satisfactory. A good example is the function $y = x^{1/3}$ for x = 0 to 1. Picking four points x = (0, 0.2, 0.6, 1), the resulting cubic polynomial can be seen to wiggle too much to represent the desired function very well. This is often the case. Wait - can't the 1st or 2nd derivative be specified to force a better fit? Unfortunately, the answer is no. The reason is that all four constraints were used up by requiring the curve to interpolate four points. Remember, only four things (points, derivatives) can be specified.

Continuity – the concept of continuity is important in spline interpolation. If two polynomials are connected together (end point of the 1st is the beginning point of the 2nd) but no other conditions are specified, the two equations (or splines) are said to have C0 continuity, or zero derivative continuity. Though that can make some interesting plots, it doesn't result in a very smooth curve through the points. Additionally, if the 1st derivatives at the connection points are forced to be equal, the curve is said to have C1 (i.e. 1st derivative) continuity. That makes a smoother curve, but an even smoother one can be obtained if C2 continuity (i.e. 2nd derivative) is also enforced. The traditional implementation of cubic splines for the purpose of interpolating points uses C0, C1, and C2 continuity.







Development of Spline Equations

Consider the problem of constructing two cubic splines to fit three data points $p_1 = (x_1, y_1)$, $p_2 = (x_2, y_2)$, $p_3 = (x_3, y_3)$. Refer to the last plot on the right, above. This is the simplest case of cubic spline interpolation that will illustrate the methods used in more normal cases where several points are present. The key characteristics of cubic spline interpolation are:

1. Curve passes through all specified data points (C0 continuity)
2. 1st derivative continuity (C1) at interior points
3. 2nd derivative continuity (C2) at interior points
4. Boundary conditions are specified at the free ends

Begin with the equations of the two splines (using the spline starting point for x_0):

spline #1 $(x_1 \le x \le x_2)$

$y = a_1(x-x_1)^3 + b_1(x-x_1)^2 + c_1(x-x_1) + d_1$

$y' = 3a_1(x-x_1)^2 + 2b_1(x-x_1) + c_1$

$y'' = 6a_1(x-x_1) + 2b_1$

spline #2 $(x_2 \le x \le x_3)$

$y = a_2(x-x_2)^3 + b_2(x-x_2)^2 + c_2(x-x_2) + d_2$

$y' = 3a_2(x-x_2)^2 + 2b_2(x-x_2) + c_2$

$y'' = 6a_2(x-x_2) + 2b_2$

For now, the focus will be on spline #1. Starting with the 2nd derivative, impose the compatibility constraints that $y'' = y_1''$ at $x = x_1$ and $y'' = y_2''$ at $x = x_2$. Defining x_2-x_1 as h_1:

$y_1'' = 6a_1(x_1-x_1) + 2b_1 = 0 + 2b_1 = 2b_1$ $\qquad b_1 = y_1''/2$

$y_2'' = 6a_1(x_2-x_1) + 2b_1 = 6a_1h_1 + y_1''$ $\qquad a_1 = (y_2''-y_1'')/6h_1$

This results in the following equation for the 2nd derivative:

$y'' = (x-x_1)(y_2''-y_1'')/(x_2-x_1) + y_1''$

which can be verified to be correct (i.e. $y'' = y_1''$ at $x = x_1$ and $y'' = y_2''$ at $x = x_2$). Next, apply the conditions that the spline must pass though the points, in other words $y_1 = f(x_1)$ and $y_2 = f(x_2)$:

$y_1 = a_1(x_1-x_1)^3 + b_1(x_1-x_1)^2 + c_1(x_1-x_1) + d_1 = 0 + 0 + 0 + d_1$ $\qquad d_1 = y_1$

$y_2 = (x_2-x_1)^3(y_2''-y_1'')/6h_1 + y_1''(x_2-x_1)^2/2 + c_1(x_2-x_1) + y_1$

$y_2 = h_1^3(y_2''-y_1'')/6h_1 + y_1''h_1^2/2 + c_1h_1 + y_1$

$y_2 = h_1^2(y_2''-y_1'')/6 + y_1''h_1^2/2 + c_1h_1 + y_1$

$y_2-y_1 = y_2''h_1^2/6 - y_1''h_1^2/6 + y_1''h_1^2/2 + c_1h_1$

$(y_2-y_1)/h_1 = y_2''h_1/6 - y_1''h_1/6 + y_1''h_1/2 + c_1$

$(y_2-y_1)/h_1 = y_2''h_1/6 - y_1''h_1/6 + 3y_1''h_1/6 + c_1$

$(y_2-y_1)/h_1 = y_2''h_1/6 + y_1''h_1/3 + c_1$ $\qquad c_1 = (y_2-y_1)/h_1 - y_2''h_1/6 - y_1''h_1/3$

Finally, impose the compatibility condition that y_2' in spline #1 must equal y_2' in spline #2:

$3a_1(x_2-x_1)^2 + 2b_1(x_2-x_1) + c_1 = 3a_2(x_2-x_2)^2 + 2b_2(x_2-x_2) + c_2$

$3a_1h_1^2 + 2b_1h_1 + c_1 = c_2$

$h_1(y_2''-y_1'')/2 + y_1''h_1 + (y_2-y_1)/h_1 - y_2''h_1/6 - y_1''h_1/3 = (y_3-y_2)/h_2 - y_3''h_2/6 - y_2''h_2/3$

$h_1(y_2''-y_1'')/2 + y_1''h_1 - y_2''h_1/6 - y_1''h_1/3 + y_3''h_2/6 + y_2''h_2/3 = (y_3-y_2)/h_2 - (y_2-y_1)/h_1$

$3h_1(y_2''-y_1'') + 6y_1''h_1 - y_2''h_1 - 2y_1''h_1 + y_3''h_2 + 2y_2''h_2 = 6(y_3-y_2)/h_2 - 6(y_2-y_1)/h_1$

$3h_1y_2'' - 3h_1y_1'' + 6y_1''h_1 - y_2''h_1 - 2y_1''h_1 + y_3''h_2 + 2y_2''h_2 = 6(y_3-y_2)/h_2 - 6(y_2-y_1)/h_1$

$y_1''(6h_1 - 3h_1 - 2h_1) + y_2''(2h_1 + 2h_2) + y_3''h_2 = 6(y_3-y_2)/h_2 - 6(y_2-y_1)/h_1$

$h_1y_1'' + 2(h_1 + h_2)y_2'' + h_2y_3'' = 6[(y_3-y_2)/h_2 - (y_2-y_1)/h_1]$ $\qquad$ governing equation for cubic splines.

Generalizing, this equation results in a tri-diagonal set of linear equations (Ax = b), where x represents the unknowns (2nd derivatives of the points), and b is the right hand side. Tri-diagonal sets of linear equations are efficiently solved with specialized algorithms.

$$\begin{bmatrix} ? & & & & & & 0 \\ h_1 & 2(h_1+h_2) & h_2 & & & & \\ & h_2 & 2(h_2+h_3) & h_3 & \ddots & & \\ & & & \ddots & & & \\ & & & & \ddots & h_{n-2} & \\ & & & & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ 0 & & & & & & ? \end{bmatrix} \begin{bmatrix} y_1'' \\ y_2'' \\ \vdots \\ \vdots \\ \vdots \\ y_{n-1}'' \\ y_n'' \end{bmatrix} = 6 \begin{bmatrix} ? \\ (y_3-y_2)/h_2-(y_2-y_1)/h_1 \\ \vdots \\ \vdots \\ \vdots \\ (y_n-y_{n-1})/h_{n-1}-(y_{n-1}-y_{n-2})/h_{n-2} \\ ? \end{bmatrix}$$

If equal point spacing is used (i.e. $h_1 = h_2 = \ldots h_{n-1} = h$), even more simplification can be made:

$$\begin{bmatrix} ? & & & & & 0 \\ 1 & 4 & 1 & & & \\ & 1 & 4 & 1 & \ddots & \\ & & & \ddots & & \\ & & & & \ddots & 1 \\ & & & 1 & 4 & 1 \\ 0 & & & & & ? \end{bmatrix} \begin{bmatrix} y_1'' \\ y_2'' \\ \vdots \\ \vdots \\ \vdots \\ y_{n-1}'' \\ y_n'' \end{bmatrix} = 6/(h*h) \begin{bmatrix} ? \\ y_3-2y_2+y_1 \\ y_4-2y_3+y_2 \\ \vdots \\ \vdots \\ y_n-2y_{n-1}+y_{n-2} \\ ? \end{bmatrix}$$

The first and last equations (shown as ?) represent the boundary conditions of the free ends of the spline that must be chosen. Often, so called 'natural' boundary conditions are used, where the 2nd derivative is set to zero. Natural boundary conditions result in total minimum curvature. Other boundary conditions can be used. Some examples are:

1. Natural boundary conditions	$y_1'' = 0$	$y_n'' = 0$
2. Parabolic runout	$y_1'' = y_2''$	$y_{n-1}'' = y_n''$
3. Zero slope	see problem #2	see problem #2
4. Specified 1st derivative	see problem #2	see problem #2
5. Specified 2nd derivative	$y_1'' = s_1$	$y_n'' = s_n$

It is interesting to note that the equations for the spline segment coefficients can be represented in the following matrix form (example for spline segment #1):

$$\begin{bmatrix} a_1 \\ b_1 \\ c_1 \\ d_1 \end{bmatrix} = \begin{bmatrix} . & . & -\frac{1}{6h_1} & \frac{1}{6h_1} \\ . & . & \frac{1}{2} & . \\ -\frac{1}{h_1} & \frac{1}{h_1} & -\frac{h_1}{3} & -\frac{h_1}{6} \\ 1 & . & . & . \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_1'' \\ y_2'' \end{bmatrix} = \frac{1}{6h_1} \begin{bmatrix} . & . & -1 & 1 \\ . & . & 3h_1 & . \\ -6 & 6 & -2h_1^2 & -h_1^2 \\ 6h_1 & . & . & . \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_1'' \\ y_2'' \end{bmatrix}$$

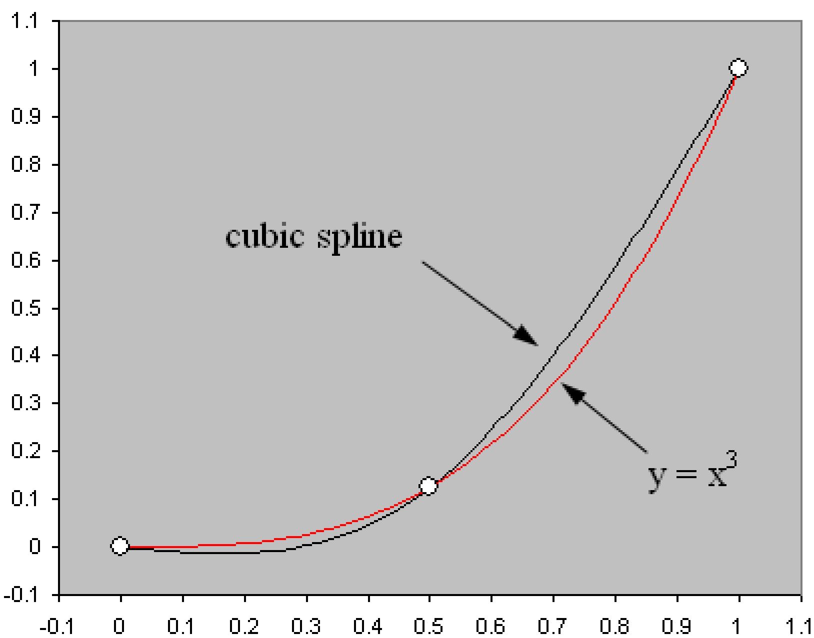
Example Problem #1 (function interpolation):

Let's illustrate with a specific problem: fit two cubic splines to the function $\boldsymbol{y = x^3}$ for x = 0 to 1. Thus, $x_1 = 0$, $y_1 = 0$, $x_3 = 1$, $y_3 = 1$. We'll pick $x_2 = 0.5$ (thus $y_2 = 0.125$) and use natural boundary conditions. Because the only unknowns are the 2nd derivative at each point, we have a 3 x 3 matrix to solve. Also, since $(x_2\text{-}x_1) = h_1 = (x_3\text{-}x_2) = h_2 = 0.5$, we can used the simplified version (Note: . means zero):

$$\begin{bmatrix} 1 & . & . \\ 1 & 4 & 1 \\ . & . & 1 \end{bmatrix} \begin{bmatrix} y_1'' \\ y_2'' \\ y_3'' \end{bmatrix} = 6/(0.25) \begin{bmatrix} 0 \\ y_3 - 2y_2 + y_1 \\ 0 \end{bmatrix} = 24 \begin{bmatrix} 0 \\ 1 - 2(0.125) + 0 \\ 0 \end{bmatrix} = 24 \begin{bmatrix} 0 \\ 0.75 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 18 \\ 0 \end{bmatrix}$$

The solution is $[y_1'', y_2'', y_3'']^T = [0, 4.5, 0]^T$. Note: in this case the solution is trivial, $y_2'' = 18/4$. From this we can calculate the coefficients of the cubic spline segments:

$a_1 = (y_2''-y_1'')/6h_1$	$b_1 = y_1''/2$	$c_1 = (y_2-y_1)/h_1 - y_2''h_1/6 - y_1''h_1/3$	$d_1 = y_1$
$a_2 = (y_3''-y_2'')/6h_2$	$b_2 = y_2''/2$	$c_2 = (y_3-y_2)/h_2 - y_3''h_2/6 - y_2''h_2/3$	$d_2 = y_2$
$a_1 = 1.5$	$b_1 = 0$	$c_1 = -0.125$	$d_1 = 0$
$a_2 = -1.5$	$b_2 = 2.25$	$c_2 = 1.0$	$d_2 = 0.125$



As can be seen in the plot, the cubic spline interpolation doesn't fit the function very well. Wait a minute. How can two cubic splines not fit a cubic polynomial very well? It should be a perfect fit, especially since it only takes one cubic spline to represent the cubic polynomial function $y = x^3$. The answer is that the 2nd derivative of the spline was forced to be zero at each free end. This works fine at x = 0 for $y = x^3$ because the 2nd derivative of this function is indeed 0 at x = 0. However, it isn't a good choice at x = 1 because the 2nd derivative of $y = x^3$ at x = 1 is 6x = 6. If y_3'' is set to 6 instead of 0, the fit is perfect. This illustrates the importance of choosing appropriate boundary conditions for the problem at hand.

Example Problem #2 (specify slope at end points):

As a final illustration, we will show how to enforce a slope at either end. Recall the equation of the 1st derivative:

At $x = x_1$:

$y_1' = 3a_1(x_1-x_1)^2 + 2b_1(x_1-x_1) + c_1 = 0 + 0 + c_1 = c_1 = (y_2-y_1)/h_1 - y_2''h_1/6 - y_1''h_1/3$

$(2h_1)y_1'' + (h_1)y_2'' = 6[(y_2-y_1)/h_1 - y_1']$

At $x = x_3$:

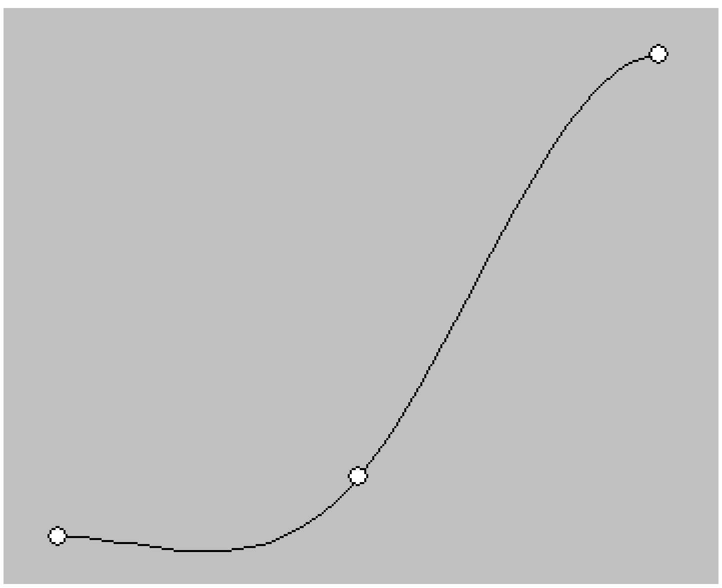
$y_3' = 3a_2(x_3-x_2)^2 + 2b_2(x_3-x_2) + c_2 = 3a_2h_2^2 + 2b_2h_2 + c_2$
$y_3' = 3h_2^2(y_3''-y_2'')/6h_2 + 2h_2y_2''/2 + (y_3-y_2)/h_2 - y_3''h_2/6 - y_2''h_2/3$
$y_3' = 3h_2(y_3''-y_2'')/6 + h_2y_2'' + (y_3-y_2)/h_2 - y_3''h_2/6 - y_2''h_2/3$
$h_2y_3''/2 - h_2y_2''/2 + h_2y_2'' - y_3''h_2/6 - y_2''h_2/3 = y_3' - (y_3-y_2)/h_2$
$3h_2y_3'' - 3h_2y_2'' + 6h_2y_2'' - y_3''h_2 - 2y_2''h_2 = 6(y_3' - (y_3-y_2)/h_2)$
$y_3''(3h_2 - h_2) - y_2''(6h_2 - 3h_2 - 2h_2) = 6(y_3' - (y_3-y_2)/h_2)$
$(h_2)y_2'' + (2h_2)y_3'' = 6[y_3' - (y_3-y_2)/h_2]$

The resulting matrix equation is:

$$\begin{bmatrix} 2h_1 & h_1 & . \\ h_1 & 2(h_1+h_2) & h_2 \\ . & h_2 & 2h_2 \end{bmatrix} \begin{bmatrix} y_1'' \\ y_2'' \\ y_3'' \end{bmatrix} = 6 \begin{bmatrix} (y_2-y_1)/h_1 - y_1' \\ (y_3-y_2)/h_2 - (y_2-y_1)/h_1 \\ y_3' - (y_3-y_2)/h_2 \end{bmatrix}$$

Using the simplified version (since $h_1 = h_2 = h = 0.5$), and setting the 1st derivatives to zero at both ends (i.e. horizontal slope), the result is:

$$\begin{bmatrix} 2 & 1 & . \\ 1 & 4 & 1 \\ . & 1 & 2 \end{bmatrix} \begin{bmatrix} y_1'' \\ y_2'' \\ y_3'' \end{bmatrix} = 6/(0.25) \begin{bmatrix} y_2-y_1-0 \\ y_3-2y_2+y_1 \\ 0-(y_3-y_2) \end{bmatrix} = 24 \begin{bmatrix} 0.125-0 \\ 1-2(0.125)+0 \\ 0-(1-0.125) \end{bmatrix} = 24 \begin{bmatrix} 0.125 \\ 0.75 \\ -0.875 \end{bmatrix} = \begin{bmatrix} 3 \\ 18 \\ -21 \end{bmatrix}$$



The solution is $[y_1'', y_2'', y_3'']^T = [-3, 9, -15]^T$, from which we can calculate the spline segment coefficients and plot the result.

Conclusion:

We have demonstrated a computationally efficient method of formulating cubic splines to interpolate a given set of points and have shown how to implement various free end boundary conditions.

Discussion:

The formulation described here is by no means the only one - there are other formulations of cubic splines. One possibility is to set up the matrix equations to directly calculate the spline segment coefficients, but it requires a matrix of dimension 4*(n-1), which is much more computationally intensive than the method shown here. If it is desired to not choose the free end boundary conditions, the splines on either end can be fit to the three points instead of two, or the method illustrated here can be used with the boundary conditions determined by fitting splines to four points on either end. These are just a few of the possible techniques for cubic spline interpolation. Lastly, it is worth noting that by defining $t = x-x_0$, the spline equations for example problems #1 and #2 can be expressed as:

Spline #1:	$y = a_1t^3 + b_1t^2 + c_1t + d_1$	$0 \le t \le (x_2-x_1)$	or	$0 \le t \le h_1$
Spline #2:	$y = a_2t^3 + b_2t^2 + c_2t + d_2$	$0 \le t \le (x_3-x_2)$	or	$0 \le t \le h_2$