

# Cubic Spline Interpolation

Sky McKinley and Megan Levine

Math 45: Linear Algebra

**Abstract.** An introduction into the theory and application of cubic splines with accompanying Matlab m-file cspline.m

## Introduction

Real world numerical data is usually difficult to analyze. Any function which would effectively correlate the data would be difficult to obtain and highly unwieldy. To this end, the idea of the cubic spline was developed. Using this process, a series of unique cubic polynomials are fitted between each of the data points, with the stipulation that the curve obtained be continuous and appear smooth. These cubic splines can then be used to determine rates of change and cumulative change over an interval. In this brief introduction, we will only discuss splines which interpolate equally spaced data points, although a more robust form could encompass unequally spaced points.

## Theory

The fundamental idea behind cubic spline interpolation is based on the engineer's tool used to draw smooth curves through a number of points. This spline consists of weights attached to a flat surface at the points to be connected. A flexible strip is then bent across each of these weights, resulting in a pleasingly smooth curve.

The mathematical spline is similar in principle. The points, in this case, are numerical data. The weights are the coefficients on the cubic polynomials used to interpolate the data. These coefficients 'bend' the line so that it passes through each of the data points without any erratic behavior or breaks in continuity.

## Process

The essential idea is to fit a piecewise function of the form

$$S(x) = \begin{cases} s_1(x) & \text{if } x_1 \leq x < x_2 \\ s_2(x) & \text{if } x_2 \leq x < x_3 \\ \vdots & \\ s_{n-1}(x) & \text{if } x_{n-1} \leq x < x_n \end{cases} \quad (1)$$

where  $s_i$  is a third degree polynomial defined by

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

for  $i = 1, 2, \dots, n - 1$ .

The first and second derivatives of these  $n - 1$  equations are fundamental to this process, and they are

$$s'_i(x) = 3a_i(x - x_i)^2 + 2b_i(x - x_i) + c_i \quad (3)$$

$$s''_i(x) = 6a_i(x - x_i) + 2b_i \quad (4)$$

for  $i = 1, 2, \dots, n - 1$ .

## The Four Properties of Cubic Splines

Our spline will need to conform to the following stipulations.

1. The piecewise function  $S(x)$  will interpolate all data points.

2.  $S(x)$  will be continuous on the interval  $[x_1, x_n]$

3.  $S'(x)$  will be continuous on the interval  $[x_1, x_n]$

4.  $S''(x)$  will be continuous on the interval  $[x_1, x_n]$

Since the piecewise function  $S(x)$  will interpolate all of the data points, we can conclude that

$$S(x_i) = y_i \quad (5)$$

for  $i = 1, 2, \dots, n - 1$ . Since  $x_i \in [x_i, x_{i+1}]$ ,  $S(x_i) = s_i(x_i)$  and we can use equation (2) to produce

$$y_i = s_i(x_i) \quad (6)$$

$$y_i = a_i(x_i - x_i)^3 + b_i(x_i - x_i)^2 + c_i(x_i - x_i) + d_i$$

$$y_i = d_i$$

for each  $i = 1, 2, \dots, n - 1$ .

Since the curve  $S(x)$  must be continuous across its entire interval, it can be concluded that each sub-function must join at the data points, so

$$s_i(x_i) = s_{i-1}(x_i) \quad (7)$$

for  $i = 2, 3, \dots, n$ .

From equation (2),

$$s_i(x_i) = d_i$$

$$\text{and} \quad (8)$$

$$s_{i-1}(x_i) = a_{i-1}(x_i - x_{i-1})^3 + b_{i-1}(x_i - x_{i-1})^2 + c_{i-1}(x_i - x_{i-1}) + d_{i-1}$$

so

$$d_i = a_{i-1}(x_i - x_{i-1})^3 + b_{i-1}(x_i - x_{i-1})^2 + c_{i-1}(x_i - x_{i-1}) + d_{i-1} \quad (9)$$

for  $i = 2, 3, \dots, n - 1$ . Letting  $h = x_i - x_{i-1}$  in equation (8), we have

$$d_i = a_{i-1}h^3 + b_{i-1}h^2 + c_{i-1}h + d_{i-1} \quad (10)$$

for  $i = 2, 3, \dots, n - 1$ .

Also, to make the curve smooth across the interval, the derivatives must be equal at the data points; that is,

$$s'_i(x_i) = s'_{i-1}(x_i) \quad (11)$$

However, by equation (3),

$$s'_i(x_i) = c_i$$

and

$$s'_{i-1}(x_i) = 3a_{i-1}(x_i - x_{i-1})^2 + 2b_{i-1}(x_i - x_{i-1}) + c_{i-1}$$

so

$$c_i = 3a_{i-1}(x_i - x_{i-1})^2 + 2b_{i-1}(x_i - x_{i-1}) + c_{i-1}. \quad (12)$$

Again, letting  $h = x_i - x_{i-1}$ , we arrive at

$$c_i = 3a_{i-1}h^2 + 2b_{i-1}h + c_{i-1} \quad (13)$$

for  $i = 2, 3, \dots, n-1$ .

From equation (4),  $s''_i(x) = 6a_i(x - x_i) + 2b_i$ , so

$$s''_i(x) = 6a_i(x - x_i) + 2b_i \quad (14)$$

$$s''_i(x_i) = 6a_i(x_i - x_i) + 2b_i$$

$$s''_i(x_i) = 2b_i$$

for  $i = 2, 3, \dots, n-2$ .

Lastly, since  $s''_i(x)$  has to be continuous across the interval,  $s''_i(x_i) = s''_{i+1}(x_i)$  for  $i = 1, 2, 3, \dots, n-1$ . This and equation (14) lead us to the equation

$$s''_i(x_{i+1}) = 6a_i(x_{i+1} - x_i) + 2b_i \quad (15)$$

$$s''_{i+1}(x_{i+1}) = 6a_i(x_{i+1} - x_i) + 2b_i \quad (16)$$

and, letting  $h = x_{i+1} - x_i$  and using the conclusion from equations (14) and (16),

$$s''_{i+1}(x_{i+1}) = 6a_i(x_{i+1} - x_i) + 2b_i \quad (17)$$

$$2b_{i+1} = 6a_i h + 2b_i \quad (18)$$

These equations can be much simplified by substituting  $M_i$  for  $s''_i(x_i)$  and expressing the above equations in terms of  $M_i$  and  $y_i$ . This makes the determination of the weights  $a_i$ ,  $b_i$ ,  $c_i$ , and  $d_i$  a much easier task. Each  $b_i$  can be represented by

$$s''_i(x_i) = 2b_i \quad (19)$$

$$M_i = 2b_i$$

$$b_i = \frac{M_i}{2}$$

and  $d_i$  has already been determined to be

$$d_i = y_i. \quad (20)$$

Similarly, using equation  $a_i$  can be re-written as

$$\begin{aligned}
2b_{i+1} &= 6a_i h + 2b_i \\
6a_i h &= 2b_{i+1} - 2b_i \\
a_i &= \frac{2b_{i+1} - 2b_i}{6h} \\
a_i &= \frac{2(\frac{M_{i+1}}{2}) - 2(\frac{M_i}{2})}{6h} \\
a_i &= \frac{M_{i+1} - M_i}{6h}
\end{aligned} \tag{21}$$

and  $c_i$  can be re-written as

$$\begin{aligned}
d_{i+1} &= a_i h^3 + b_i h^2 + c_i h + d_i \\
c_i h &= -a_i h^3 - b_i h^2 - d_i + d_{i+1} \\
c_i &= \frac{-a_i h^3 - b_i h^2 - d_i + d_{i+1}}{h} \\
c_i &= \frac{-a_i h^3 - b_i h^2}{h} + \frac{-d_i + d_{i+1}}{h} \\
c_i &= (-a_i h^2 - b_i h) - \frac{d_i - d_{i+1}}{h} \\
c_i &= -(\frac{M_{i+1} - M_i}{6h} h^2 + \frac{M_i}{2} h) - \frac{y_i - y_{i+1}}{h} \\
c_i &= \frac{y_{i+1} - y_i}{h} - (\frac{M_{i+1} - M_i}{6} h + \frac{3M_i}{6} h) \\
c_i &= \frac{y_{i+1} - y_i}{h} - (\frac{M_{i+1} - M_i + 3M_i}{6}) h \\
c_i &= \frac{y_{i+1} - y_i}{h} - (\frac{M_{i+1} + 2M_i}{6}) h.
\end{aligned} \tag{22}$$

We now have our equations for determining the weights for our  $n - 1$  equations

$$\begin{aligned}
a_i &= \frac{M_{i+1} - M_i}{6h} \\
b_i &= \frac{M_i}{2} \\
c_i &= \frac{y_{i+1} - y_i}{h} - (\frac{M_{i+1} + 2M_i}{6}) h \\
d_i &= y_i
\end{aligned} \tag{23}$$

These systems can be handled more conveniently by putting them into matrix form as follows

$$c_{i+1} = 3a_i h^2 + 2b_i h + c_i$$

(24)

$$\begin{aligned}
3\left(\frac{M_{i+1}-M_i}{6h}\right)h^2 + 2\left(\frac{M_i}{2}\right)h + \frac{y_{i+1}-y_i}{h} - \left(\frac{M_{i+1}+2M_i}{6}\right)h &= \frac{y_{i+2}-y_{i+1}}{h} - \left(\frac{M_{i+2}+2M_{i+1}}{6}\right)h \\
3\left(\frac{M_{i+1}-M_i}{6h}\right)h^2 + 2\left(\frac{M_i}{2}\right)h - \left(\frac{M_{i+1}+2M_i}{6}\right)h + \left(\frac{M_{i+2}+2M_{i+1}}{6}\right)h &= -\left(\frac{y_{i+1}-y_i}{h}\right) + \frac{y_{i+2}-y_{i+1}}{h} \\
h\left(\frac{3M_{i+1}-3M_i}{6} + \frac{6M_i}{6} - \left(\frac{M_{i+1}+2M_i}{6}\right) + \left(\frac{M_{i+2}+2M_{i+1}}{6}\right)\right) &= \frac{y_i - 2y_{i+1} + y_{i+2}}{h} \\
\frac{h}{6}(M_i + 4M_{i+1} + M_{i+2}) &= \frac{y_i - 2y_{i+1} + y_{i+2}}{h} \\
M_i + 4M_{i+1} + M_{i+2} &= 6\left(\frac{y_i - 2y_{i+1} + y_{i+2}}{h^2}\right)
\end{aligned}$$

for  $i = 1, 2, 3, \dots, n-1$

which leads to the matrix equation

$$\begin{bmatrix}
1 & 4 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 1 & 4 & 1 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 4 & \cdots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \cdots & 4 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & \cdots & 1 & 4 & 1 & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 & 1 & 4 & 1
\end{bmatrix}
\begin{bmatrix}
M_1 \\
M_2 \\
M_3 \\
M_4 \\
\vdots \\
M_{n-3} \\
M_{n-2} \\
M_{n-1} \\
M_n
\end{bmatrix}
= \frac{6}{h^2}
\begin{bmatrix}
y_1 - 2y_2 + y_3 \\
y_2 - 2y_3 + y_4 \\
y_3 - 2y_4 + y_5 \\
\vdots \\
y_{n-4} - 2y_{n-3} + y_{n-2} \\
y_{n-3} - 2y_{n-2} + y_{n-1} \\
y_{n-2} - 2y_{n-1} + y_n
\end{bmatrix}
\quad (25)$$

Note that this system has  $n-2$  rows and  $n$  columns, and is therefore under-determined. In order to generate a unique cubic spline, two other conditions must be imposed upon the system. This leads us to our next section.

## Three types of Splines

### Natural splines

This first spline type includes the stipulation that the second derivative be equal to zero at the endpoints.

$$M_1 = M_n = 0 \quad (26)$$

This results in the spline extending as a line outside the endpoints. The matrix for determining the  $M_1 - M_n$  values can be adapted accordingly.

$$\begin{bmatrix}
1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 1 & 4 & 1 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 4 & \cdots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \cdots & 4 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & \cdots & 1 & 4 & 1 & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
0 \\
M_2 \\
M_3 \\
M_4 \\
\vdots \\
M_{n-3} \\
M_{n-2} \\
M_{n-1} \\
0
\end{bmatrix}
= \frac{6}{h^2}
\begin{bmatrix}
y_1 - 2y_2 + y_3 \\
y_2 - 2y_3 + y_4 \\
y_3 - 2y_4 + y_5 \\
\vdots \\
y_{n-4} - 2y_{n-3} + y_{n-2} \\
y_{n-3} - 2y_{n-2} + y_{n-1} \\
y_{n-2} - 2y_{n-1} + y_n
\end{bmatrix}
\quad (27)$$

For reasons of convenience, the first and last columns of this matrix can be eliminated, as they correspond to the  $M_1$  and  $M_n$  values, which are both 0.

$$\begin{bmatrix}
4 & 1 & 0 & \cdots & 0 & 0 & 0 \\
1 & 4 & 1 & \cdots & 0 & 0 & 0 \\
0 & 1 & 4 & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 4 & 1 & 0 \\
0 & 0 & 0 & \cdots & 1 & 4 & 1 \\
0 & 0 & 0 & \cdots & 0 & 1 & 4
\end{bmatrix}
\begin{bmatrix}
M_2 \\
M_3 \\
M_4 \\
\vdots \\
M_{n-3} \\
M_{n-2} \\
M_{n-1}
\end{bmatrix}
= \frac{6}{h^2}
\begin{bmatrix}
y_1 - 2y_2 + y_3 \\
y_2 - 2y_3 + y_4 \\
y_3 - 2y_4 + y_5 \\
\vdots \\
y_{n-4} - 2y_{n-3} + y_{n-2} \\
y_{n-3} - 2y_{n-2} + y_{n-1} \\
y_{n-2} - 2y_{n-1} + y_n
\end{bmatrix}
\quad (28)$$

This results in an  $n - 2$  by  $n - 2$  matrix, which will determine the remaining solutions for  $M_2$  through  $M_{n-1}$ . The spline is now unique.

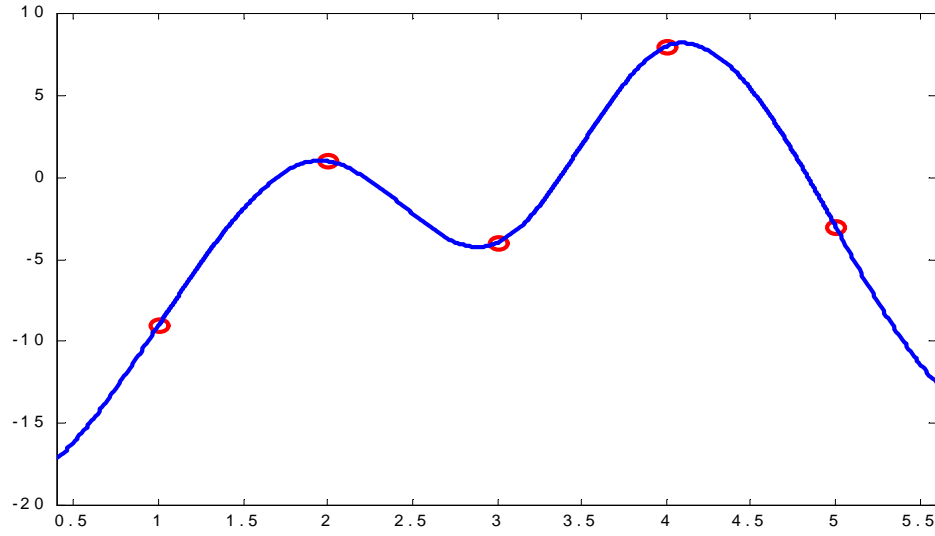


Figure 1. Natural interpolating curve

### Parabolic Runout Spline

The parabolic spline imposes the condition that the second derivative at the endpoints,  $M_1$  and  $M_n$ , be equal to  $M_2$  and  $M_{n-1}$  respectively.

$$M_1 = M_2 \quad (29)$$

$$M_n = M_{n-1}$$

The result of this condition is the curve becomes a parabolic curve at the endpoint. This type of cubic spline is useful for periodic and exponential data.

The matrix equation for this type of spline is

$$\begin{bmatrix} 5 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 4 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 4 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 4 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 5 \end{bmatrix} \begin{bmatrix} M_2 \\ M_3 \\ M_4 \\ \vdots \\ M_{n-3} \\ M_{n-2} \\ M_{n-1} \end{bmatrix} = \frac{6}{h^2} \begin{bmatrix} y_1 - 2y_2 + y_3 \\ y_2 - 2y_3 + y_4 \\ y_3 - 2y_4 + y_5 \\ \vdots \\ y_{n-4} - 2y_{n-3} + y_{n-2} \\ y_{n-3} - 2y_{n-2} + y_{n-1} \\ y_{n-2} - 2y_{n-1} + y_n \end{bmatrix} \quad (30)$$

We can now determine the values for  $M_2$  through  $M_{n-1}$ , with the values for  $M_1$  and  $M_n$  already determined.

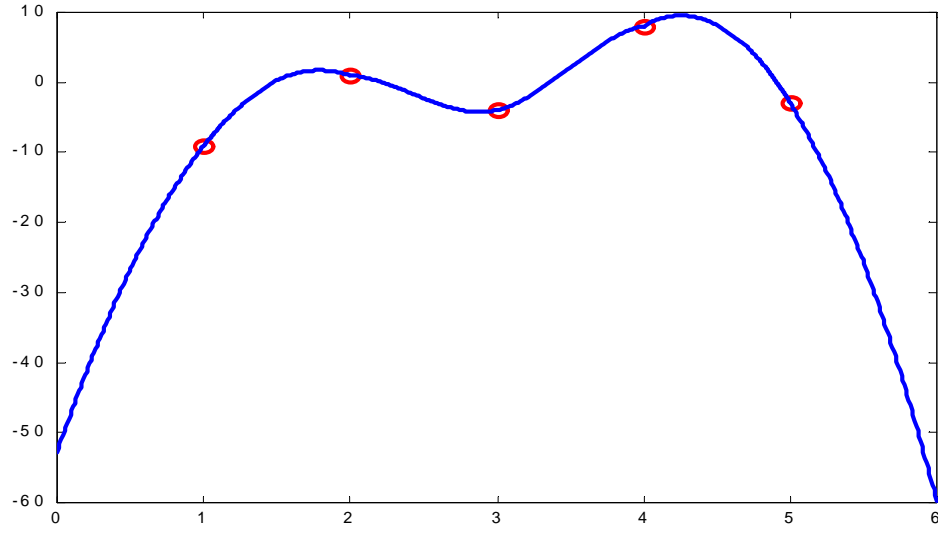


Figure 2. Parabolic Runout curve

Note that the endpoint behavior is a bit more extreme than with the natural spline option.

### Cubic Runout Spline

This last type of spline has the most extreme endpoint behavior. It assigns  $M_1$  to be  $2M_2 - M_3$  and  $M_n$  to be  $2M_{n-1} - M_{n-2}$ . This causes the curve to degrade to a single cubic curve over the last two intervals, rather than two separate functions.

The matrix equation for this type is

$$\begin{bmatrix}
 6 & 0 & 0 & \cdots & 0 & 0 & 0 \\
 1 & 4 & 1 & \cdots & 0 & 0 & 0 \\
 0 & 1 & 4 & \cdots & 0 & 0 & 0 \\
 \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
 0 & 0 & 0 & \cdots & 4 & 1 & 0 \\
 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\
 0 & 0 & 0 & \cdots & 0 & 0 & 6
 \end{bmatrix}
 \begin{bmatrix}
 M_2 \\
 M_3 \\
 M_4 \\
 \vdots \\
 M_{n-3} \\
 M_{n-2} \\
 M_{n-1}
 \end{bmatrix}
 = \frac{6}{h^2}
 \begin{bmatrix}
 y_1 - 2y_2 + y_3 \\
 y_2 - 2y_3 + y_4 \\
 y_3 - 2y_4 + y_5 \\
 \vdots \\
 y_{n-4} - 2y_{n-3} + y_{n-2} \\
 y_{n-3} - 2y_{n-2} + y_{n-1} \\
 y_{n-2} - 2y_{n-1} + y_n
 \end{bmatrix}
 \quad (31)$$



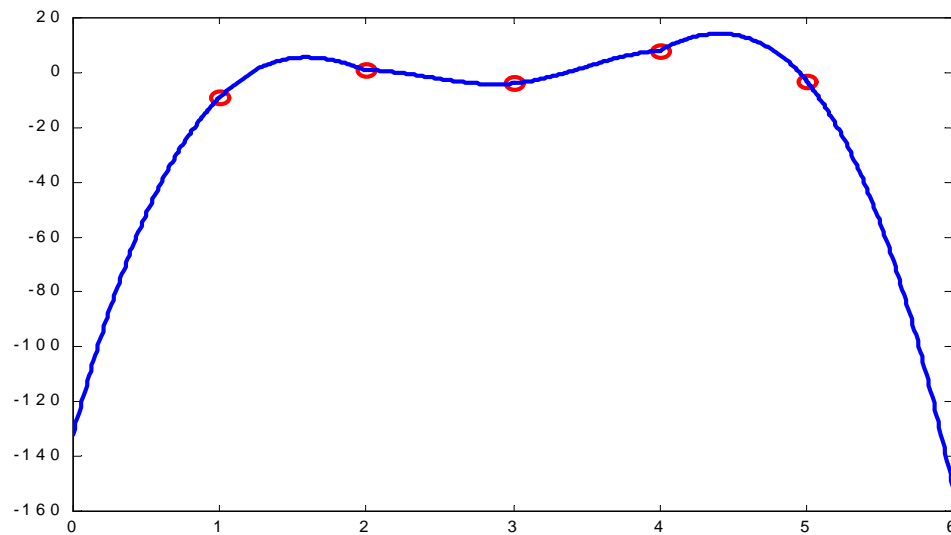


Figure 3. Cubic Runout curve

Note the pronounced curvature at the endpoints.

Keep in mind that there are many other types of interpolating spline curves, such as the Periodic Spline and the Clamped Spline. The three discussed in this work are simply the ones which we have chosen to examine; they are not intrinsically superior to, or more widely used than these other types of splines.

## Curve Fitting with Splines

The main application of cubic spline interpolation techniques is, of course, curve fitting. To this end, the consistency and efficiency of the spline as a data correlation tool will be demonstrated.

### Function Imitation

Cubic splines would not be necessary were it simple to determine a well-behaved function to fit any data set. This is, however, usually not the case. Thus, the cubic spline technique is used to generate a function to fit the data. Moreover, it can be shown that data generated by a particular function is interpolated by a spline which behaves more or less like the original function. This is testimony to the consistency of splines.

For example, the following figure was generated using the function  $y = \sin(x)$ .

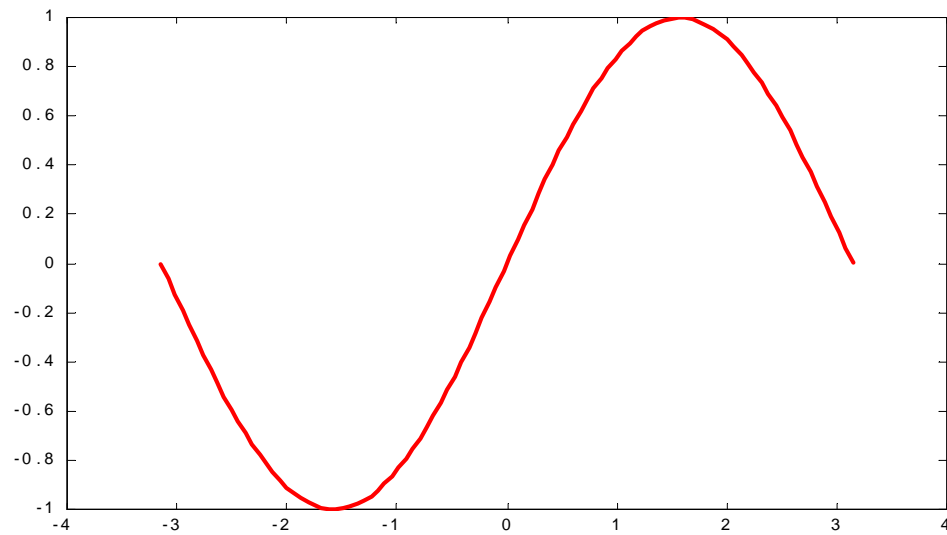


Figure 4. The graph of  $y = \sin(x)$ .

This next figure was generated by connecting six data points along the above line with a cubic spline function.

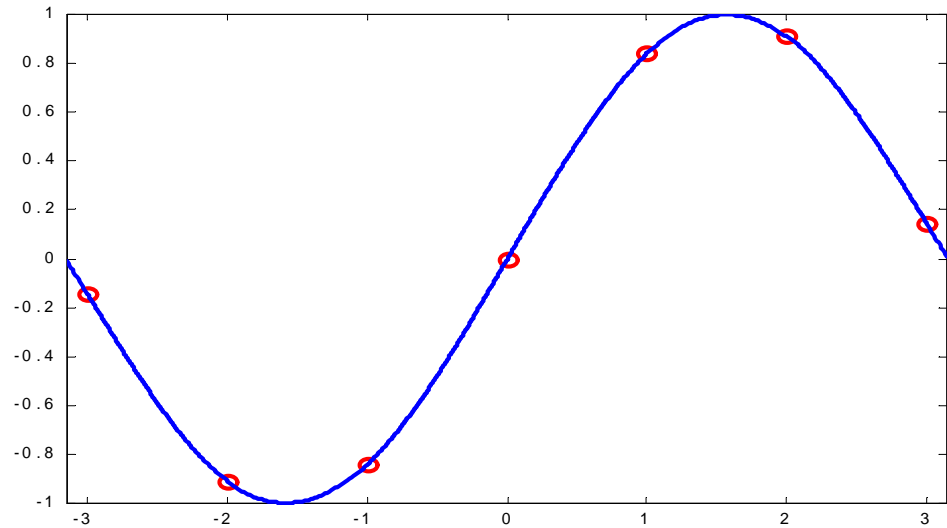


Figure 5. Spline through six points on a sine curve.

By superimposing Figure 5 on Figure 4, we can see the degree to which the cubic curve imitates the original function  $y = \sin(x)$ .

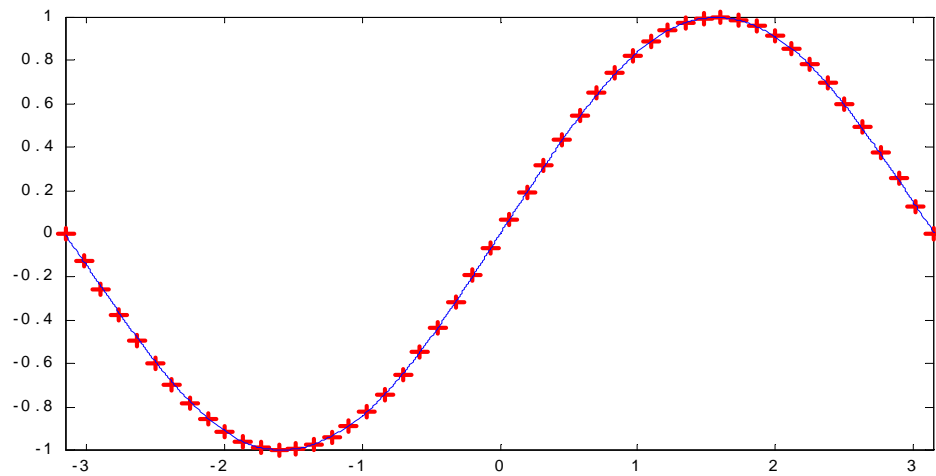


Figure 6. Superimposition of a spline curve on a sine function.

Clearly, the cubic curve closely imitates the sine curve. It has no extreme behavior between data points, and it effectively correlates the points.

This characteristic also works with more erratic functions. Take for instance the function below.

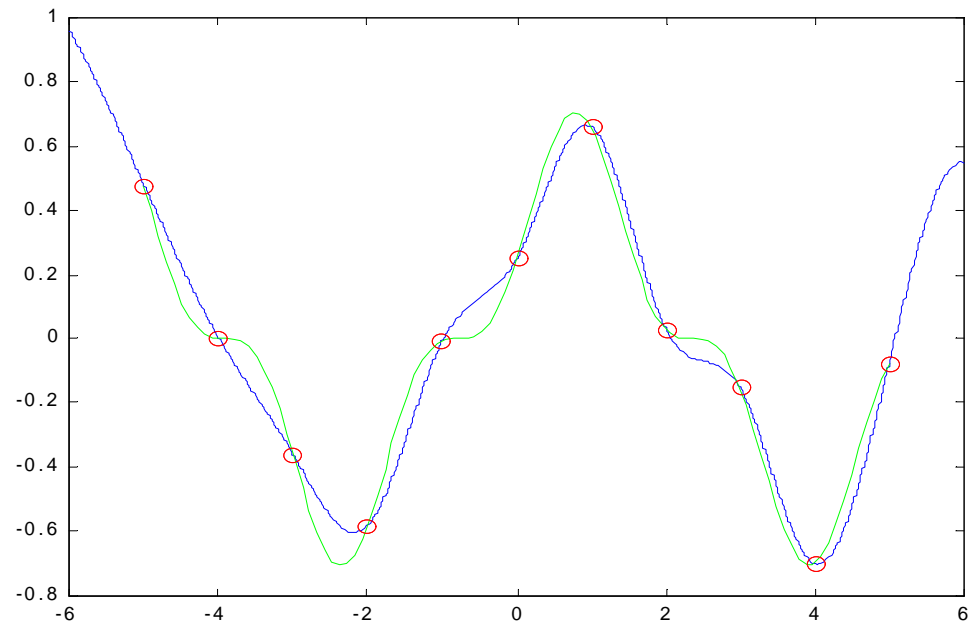


Figure 7. A cubic spline approximation of  $[\sin(x) + \cos(x)]^{3/4}$ .

While the fit is not perfect, it does closely approximate the function without a great degree of divergence.

## Data interpretation

The splines other strength lies in its ability to correlate data which doesn't follow any specific pattern without a single polynomial's extreme behavior. Take, for instance, the 100 random data points below.

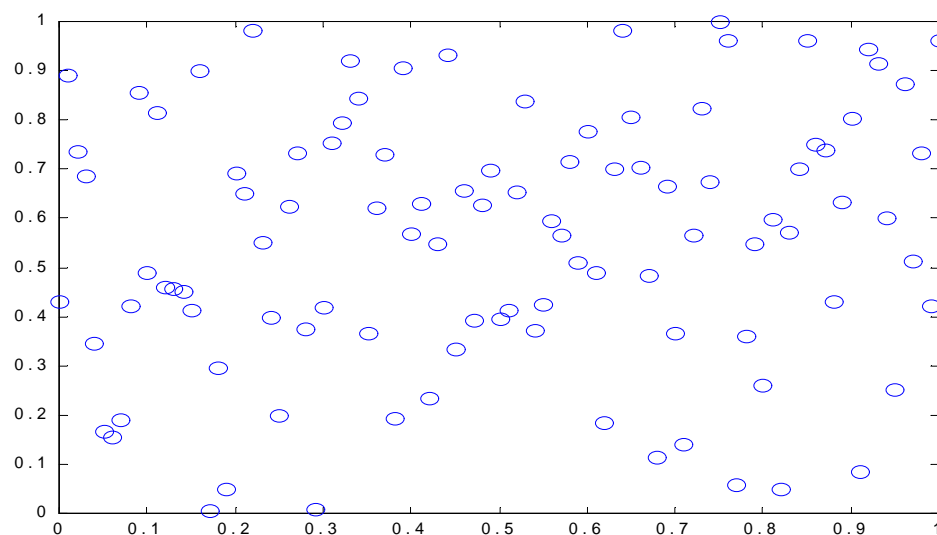


Figure 8. Random data points

Clearly there is no relation between these data points. A spline, however, can interpolate all 100 points without the drastic behavior that the necessary 99th degree polynomial would exhibit.

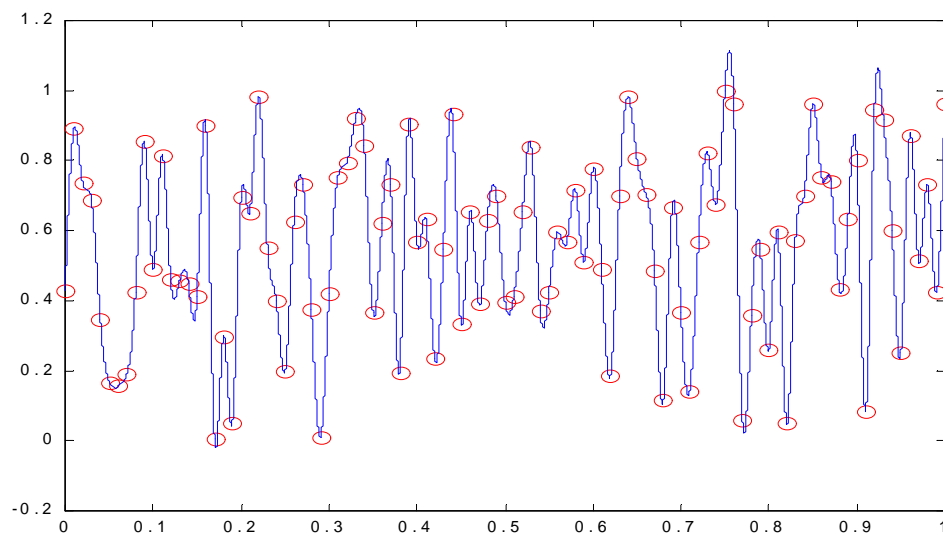


Figure 9. Cubic curve through the random data.

Note that, while the spline is not exactly pleasing to the eye, its range stays from 0 to 1.2; a much more reasonable range than an approximating polynomial. Also, it is not much of a task to generate an approximating spline which generally correlates the data while being much more well behaved.

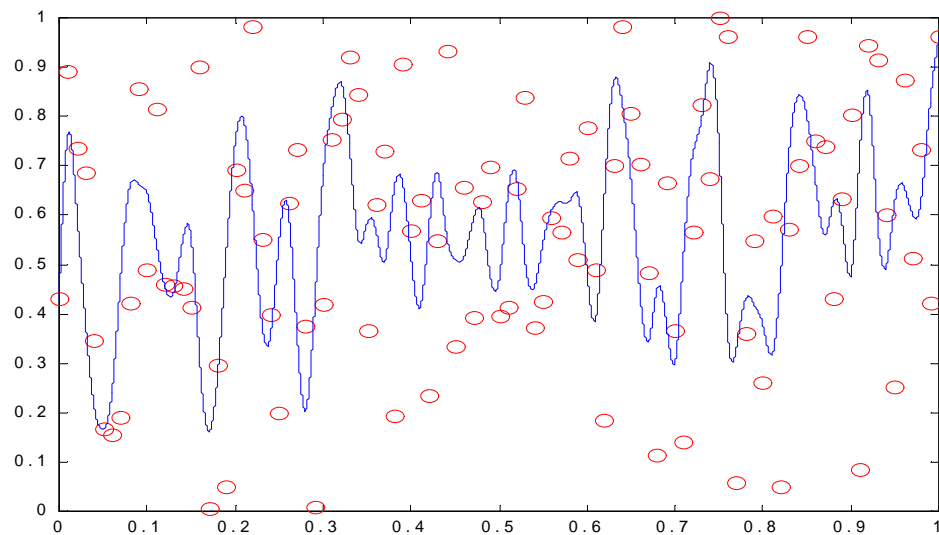


Figure 10. Approximating cubic curve

The above figure demonstrates the general trends in the data (of which there are none) without necessarily connecting all data points. Its curvature is much less severe than that of fig 9.

## Cubic Splines and Matlab

All of the graphics in this paper were generated using a cubic spline m-file and Matlab software. The m-file used is available for download at

<http://online.redwoods.cc.ca.us/instruct/darnold/laproj/fall98/Skymeg>

## Conclusion

Cubic spline interpolation is a powerful data analysis tool. Splines correlate data efficiently and effectively, no matter how random the data may seem. Once the algorithm for spline generation is produced, interpolating data with a spline becomes an easy task.

## Bibliography

Hanselman, Duane and Bruce Littlefield. *Mastering Matlab 5*. 1998 New Jersey: Prentice Hall

Nievergelt, Yves. *UMAP: Module 718; Splines in Single and Multivariable Calculus*. 1993. Lexington, MA: COMAP

Rorres, Chris and Howard Anton. *Applications of Linear Algebra 3ed.* 1984 New York: John Wiley and Sons.

Van Loan, Charles F. *Introduction to Scientific Computing.* 1997 New Jersey: Prentice Hall