3.1 Introduction

We are given the values of a function $\boldsymbol{f(x)}$ at different points $\boldsymbol{x_0, x_1, \ldots\ldots x_n}$, **we want to** find approximate values of the function $\boldsymbol{f(x)}$ for "new" x's that lie between these points for which the function values are given, and this process is called **interpolation**.

Continuing our discussion, we write these given values of a function (f) in the form

$$f_0 = f(x_0), \quad f_1 = f(x_1), \quad \ldots\ldots f_n = f(x_n)$$

Or as ordered pairs

$$(x_0, f_0), \quad (x_1, f_1), \quad \ldots\ldots (x_n, f_n)$$

These function values come from a "mathematical" function, such as a logarithm or a Bessel function. More frequently, they may be measured or automatically recorded values of an "empirical" function, such as air resistance of a car or an airplane at different speeds. Other examples of functions that are "empirical" are the yield of a chemical process at different temperatures or the size of the Iraq population as it appears from censuses taken at 10-year intervals.

3.2 Types of interpolation

A- Linear Interpolation, the simplest form of interpolations to connect data points with straight line.

B- Quadratic Interpolation, a strategy for improving the estimate is to introduce some curvature into the line connecting the points. This can be accomplished with a second order polynomial.

3.3 Newton's Divided-Difference Interpolating Polynomial

There are a variety of alternative forms for expressing an interpolating polynomial. **Newton's divided-difference interpolating polynomial** is among the most popular and useful forms. Before presenting the general equation, we will introduce the first and second-order versions because of their simple visual interpretation.

3.3.1 Linear Interpolation

The simplest form of interpolation is to connect two data points with a straight line. This technique, called **linear interpolation**, is depicted graphically in Figure (3.1) using similar triangles

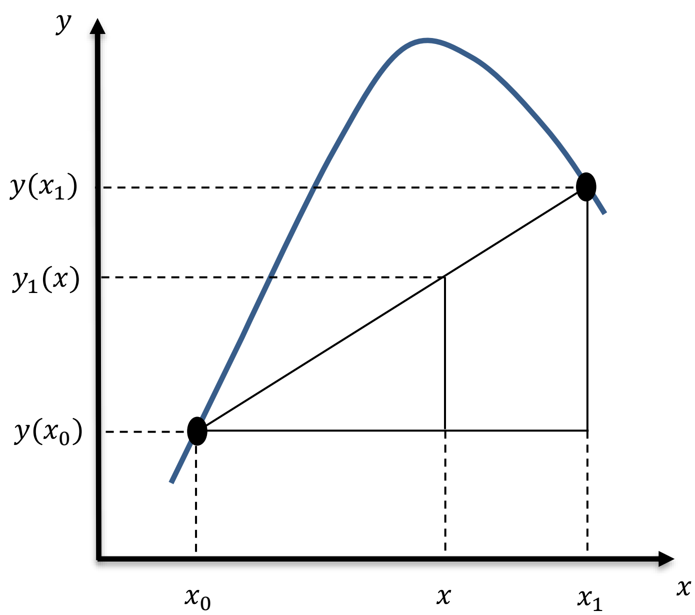

Figure (3.1) Graphical depiction of linear interpolation

$$\frac{y_1(x) - y(x_0)}{x - x_0} = \frac{y(x_1) - y(x_0)}{x_1 - x_0}$$

Which can be rearranged to yield

$$y_1(x) = y(x_0) + \frac{y(x_1) - y(x_0)}{x_1 - x_0}\,(x - x_0)$$

Which is a **linear-interpolation formula**. The notation $y_1(x)$ designates that this is a first-order interpolating polynomial.

Example (3.1)

Estimate the natural logarithm of 2 using linear interpolation. First, perform the computation by interpolating between $ln\,(1) = 0$ and $\ln(6) = 1.791759$. Then, repeat the procedure, but use a smaller interval from $ln\,(1)$ to $ln\,(4) = 1.386294$. Note that the true value of $ln\,(2)$ is 0.6931472.

Solution

We use a linear interpolation for $ln(2)$ from $x_0 = 1$ to $x_1 = 6$ to give

$$y_1(x) = y(x_0) + \frac{y(x_1) - y(x_0)}{x_1 - x_0}\,(x - x_0)$$

$$y_1(2) = 0 + \frac{1.791759 - 0}{6 - 1}\,(2 - 1) = 0.3583519$$

This means the percentage error is 48.3%. Using a smaller interval from $x_0 = 1$ to $x_1 = 4$

$$y_1(2) = 0 + \frac{1.386294 - 0}{4 - 1}\,(2 - 1) = 0.4620981$$

Thus, using the shorter interval reduces the percent relative error to 33.3%.

3.3.2 Quadratic Interpolation

The error in Example (3.1) resulted from our approximating a curve with a straight line is large. Consequently, a strategy for improving the estimate is to introduce some curvature into the line connecting the points. If three data points are available, this can be accomplished with a second-order polynomial (also called a quadratic polynomial or a *parabola*). A particularly convenient form for this purpose is

$$y_2(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1)$$

Or

$$y_2(x) = a_0 + a_1 x + a_2 x^2$$

Where

$$a_0 = b_0 - b_1 x_0 - b_2 x_0 x_1$$

$$a_1 = b_1 - b_2 x_0 - b_2 x_1$$

$$a_2 = b_2$$

A simple procedure can be used to determine the values of the coefficients

$$b_0 = f(x_0) = y(x_0)$$

$$b_1 = \frac{y(x_1) - y(x_0)}{x_1 - x_0}$$

$$b_2 = \frac{\dfrac{y(x_2) - y(x_1)}{x_2 - x_1} - \dfrac{y(x_1) - y(x_0)}{x_1 - x_0}}{x_2 - x_0}$$

3.3.3 General Form of Newton’s Interpolating Polynomials

The preceding analysis can be generalized to fit an *n*th-order polynomial to $(n + 1)$ data points. The nth-order polynomial is

$$y_n(x) = b_0 + b_1(x - x_0) + \cdots + b_n(x - x_0)(x - x_1)\ldots(x - x_{n-1})$$

As was done previously with the linear and quadratic interpolations, data points can be used to evaluate the coefficients b_0, b_1, . . . , b_n. For an nth order polynomial, *n* + 1 data points are required: $(x_0, y_0), (x_1, y_1), (x_n, y_n)$. We use these data points and the following equations to evaluate the coefficients:

$$b_0 = f(x_0)$$

$$b_1 = f(x_1, x_0)$$

$$b_2 = f(x_2, x_1, x_0)$$

.

.

$$b_n = f(x_n, x_{n-1}, \ldots, x_1, x_0)$$

The bracketed function evaluations are finite divided differences. For example, the first finite divided difference is represented generally as

$$f[x_i, x_j] = \frac{f(x_i) - f(x_j)}{x_i - x_j}$$

The **second finite divided difference**, which represents the difference of two first divided differences, is expressed generally as

$$f[x_i, x_j, x_k] = \frac{f[x_i - x_j] - f[x_j - x_k]}{x_i - x_k}$$

Similarly, the nth finite divided difference is

$$f[x_n, x_{n-1}, \dots, x_1, x_0] = \frac{f[x_n, x_{n-1}, \dots, x_1] - f[x_{n-1}, x_{n-2}, \dots, x_0]}{x_n - x_0}$$

Example (3.2)

Fit a second-order polynomial to the three points used in Example (3.1)

Solution

$$x_0 = 1 \quad y_0 = 0$$

$$x_1 = 4 \quad y_1 = 1.386294$$

$$x_2 = 6 \quad y_2 = 1.791759$$

$$b_0 = 0$$

$$b_1 = \frac{y(x_1) - y(x_0)}{x_1 - x_0}$$

$$b_1 = \frac{1.386294 - 0}{4 - 1} = 0.4620981$$

$$b_2 = \frac{\frac{y(x_2) - y(x_1)}{x_2 - x_1} - \frac{y(x_1) - y(x_0)}{x_1 - x_0}}{x_2 - x_0}$$

$$b_2 = \frac{\frac{1.791759 - 1.386294}{6 - 4} - \frac{1.386294 - 0}{4 - 1}}{6 - 1} = -0.0518731$$

$$y_2(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1)$$

$$y_2(2) = 0 + 0.4620981(2 - 1) - 0.0518731(2 - 1)(2 - 4)$$

$$y_2(2) = 0.5658444$$

Example (3.3)

In Example (3.2), data points at $x_0 = 1$, $x_1 = 4$, and $x_2 = 6$were used to estimate $ln\,(2)$ with a parabola. Now, adding a fourth point $[x_3 = 5;\ f\,(x3) = 1.609438]$, estimate $ln\,(2)$ with a third-order Newton's interpolating polynomial.

Solution

The third-order polynomial with $n = 3$, is

$$y_3(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1) + b_3(x - x_0)(x - x_1)(x - x_2)$$

$$f[x_1, x_0] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{1.386294 - 0}{4 - 1} = 0.4620981$$

$$f[x_2, x_1] = \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{1.791759 - 1.386294}{6 - 4} = 0.2027326$$

$$f[x_3, x_2] = \frac{f(x_3) - f(x_2)}{x_3 - x_2} = \frac{1.609438 - 1.791759}{5 - 6} = 0.1823216$$

The second divided differences are

$$f[x_2, x_1, x_0] = \frac{0.2027326 - 0.4620981}{6 - 1} = -0.05187311$$

$$f[x_3, x_2, x_1] = \frac{0.1823216 - 0.2027326}{5 - 4} = -0.02041100$$

$$f[x_3, x_2, x_1, x_0] = \frac{-0.02041100 - (-0.05187311)}{5 - 1} = 0.007865529$$

The result for $f[x_1, x_0]$, $f[x_2, x_1, x_0]$, $f[x_3, x_2, x_1, x_0]$ represent the coefficients b_1, b_2, b_3 respectively. $b_0 = 0$, so a third-order Newton's interpolating polynomial

$$y_3(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1) + b_3(x - x_0)(x - x_1)(x - x_2)$$

$$y_2(2) = 0 + 0.4620981(2-1) - 0.05187311(2-1)(2-4)$$
$$+ 0.007865529(2-1)(2-4)(2-6) = 0.6287686$$

This represents a relative error of 9.3%.

3.3.4 Errors of Newton's Interpolating Polynomials

Even though the Newton's Interpolating Polynomials decreases the percentage error between the calculated and exact value, there is still a percentage of error. The amount of error for an nth-order interpolating polynomial can be calculated from the following formula

$$R_n = \frac{f^{n+1}(\xi)}{(n+1)!}(x - x_0)(x - x_1) \dots (x - x_2)$$

Where ξ is somewhere in the interval containing the unknown and the data. For this formula to be of use, the function in question must be known and differentiable. This is not usually the case. Fortunately, an alternative formulation is available that does not require prior knowledge of the function. Rather, it uses a finite divided difference to approximate the $(n + 1)$th derivative

$$R_n = f[x, x_n, x_{n-1}, \dots, x_1, x_0](x - x_0)(x - x_1) \dots (x - x_n)$$

Where $f[x, x_n, x_{n-1}, \dots, x_1, x_0]$is the $(n + 1)$th finite divided difference. Because equation above contains the unknown $f(x)$, it cannot be solved for the error. However, if an additional data point $f(x_n + 1)$ is available, the equation above can be used to estimate the error, as in

$$R_n \cong f[x_{n+1}, x_n, x_{n-1}, \dots, x_1, x_0](x - x_0)(x - x_1) \dots (x - x_n)$$

Example (3.4)

Estimate the error for the second-order polynomial interpolation of Example (3.2). Use the additional data point $f(x_3) = f(5) = 1.609438$ to obtain the results.

Solution

The second-order interpolating polynomial provided an estimate of $f(2) = 0.5658444$, which represents an error of $0.6931472 - 0.5658444 = 0.1273028$.

$$R_2 = f[x_3, x_2, x_1, x_0](x - x_0)(x - x_1)(x - x_2)$$

$$R_2 = 0.007865529\ (2 - 1)(2 - 4)(2 - 6) = 0.0629242$$

This represents the estimated error at $x = 2$. it should be clear that the error estimate for the *n*th-order polynomial is equivalent to the difference between the $(n + 1)$th-order and the *n*th-order prediction. That is,

$$R_n = f_{n+1}(x) - f_n(x)$$

3.4 Lagrange Interpolation

The **Lagrange interpolating polynomial** is simply a reformulation of the Newton polynomial that avoids the computation of divided differences. It can be represented concisely as

$$f_n(x) = \sum_{i=0}^{n} L_i(x) f(x_i)$$

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_i}{x_j - x_i}$$

Where Π designates the “product of.” For example, the linear version (n = 1) is

$$f_1(x) = \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1)$$

And the second-order version is

$$f_2(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f(x_0) + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f(x_1) + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f(x_2)$$

Example (3-5)

Use Lagrange interpolating of the first order and second order to evaluate $f(x)$ at $x = 2$ due to the following data.

$x_0 = 1$	$f(x_0) = 0$
$x_1 = 4$	$f(x_1) = 1.3863$
$x_2 = 6$	$f(x_2) = 1.7917$

Solution:

The first order Lagrange interpolating is

$$f_1(x) = \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1)$$

At $x = 2$

$$f_1(2) = \frac{2 - 4}{1 - 4} * 0 + \frac{2 - 1}{4 - 1} * 1.3863$$

$$\therefore f_1(2) = 0.4621$$

The second order Lagrange interpolating is

$$f_2(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} f(x_0) + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} f(x_1) + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} f(x_2)$$

$$f_2(2) = \frac{(2-4)(2-6)}{(1-4)(1-6)} \times 0 + \frac{(2-1)(2-6)}{(4-1)(4-6)} \times 1.3863 + \frac{(2-1)(2-4)}{(6-1)(6-4)} \times 1.7917 = 0.5658444$$

H.W (1)

Compute the value of $ln\,9.2$ from $ln\,9 = 2.1972, ln\,9.5 = 2.2513$ by linear Lagrange interpolation and determine the error, using $ln\,9.2 = 2.2192$.

H.W (2)

Calculate the Lagrange polynomial $P_2(x)$ for the values $\Gamma(1.00) = 1.000, \Gamma(1.02) = 0.988, \Gamma(1.04) = 0.9784$ of the Gamma function, from it approximations of $\Gamma(1.01) and\ \Gamma(1.03)$.

H.W (3)

From the principles of Newton's divided difference method, drive the linear Lagrange interpolation formula

$$f_1(x) = \frac{x-x_1}{x_0-x_1} f(x_0) + \frac{x-x_0}{x_1-x_0} f(x_1)$$

3.4 Spline Interpolation

In the previous sections, nth-order polynomials were used to interpolate between $n + 1$ data points. For example, for eight points, we can derive a perfect seventh-order polynomial. However, there are cases where these functions can lead to erroneous results because of round-off error and overshoot. An alternative approach is to apply lower-order polynomials to subsets of data points. Such connecting polynomials are called **spline functions**. For example, third-order curves employed to connect each pair of data points are called *cubic splines*. These functions can be constructed so that the connections between adjacent cubic equations are visually smooth. I.e. **not necessarily the exactness increase with increasing the points.**

A famous example by C. Runge for which the maximum error even approaches $\infty\ as\ n \to \infty$ with the nodes kept equidistant and their number increased

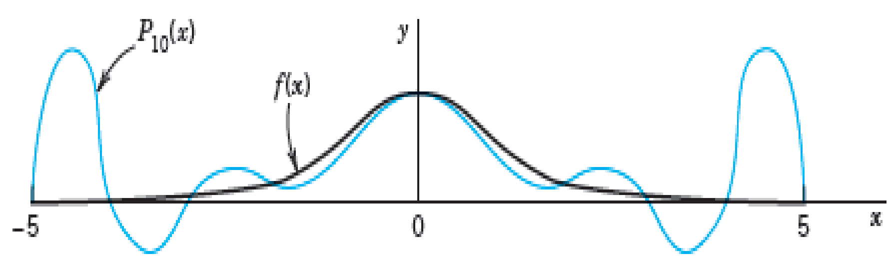

Figure (3-2) Runge's example $f(x) = 1/(1+x^2)$ and interpolating Polynomial $P_{10}(x)$

Those undesirable oscillations are avoided by the method of splines. This method is widely used in practice. It also laid the foundation for much of modern **CAD** (computer-aided design). Its name is borrowed from a draftsman's spline, which is an elastic rod bent to pass through given points and held in place by weights. The mathematical idea of this method divided into three types

1- Linear Spline

The simplest connection between two points is a straight line. The first-order splines for a group of ordered data points can be defined as a set of linear functions

$$f(x) = f(x_0) + m_0(x - x_0) \qquad x_0 \le x \le x_1$$

$$f(x) = f(x_1) + m_1(x - x_1) \qquad x_1 \le x \le x_2$$

.

.

.

$$f(x) = f(x_{n-1}) + m_{n-1}(x - x_{n-1}) \qquad x_{n-1} \le x \le x_n$$

Where m_i is the slope of the straight line connecting the points:

$$m_i = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$$

These equations can be used to evaluate the function at any point between x_0 and x_n by first locating the interval within which the point lies. Then the appropriate equation is used to determine the function value within the interval. The method is obviously identical to linear interpolation.

Example (3.6)

Fit the data in Table below with first-order splines. Evaluate the function at $x = 5$

X	$F(x)$
3.0	2.5
4.5	1.0
7.0	2.5
9.0	0.5

Solution

The data can be used to determine the slopes between points. For example, for the interval $x = 4.5$ to $x = 7$ the slope can be computed

$$m = \frac{2.5 - 1}{7 - 4.5} = 0.6$$

The slopes for the other intervals can be computed, and the resulting first-order splines are plotted in figure (3.3). The value at $x = 5$ is 1.3

Visual inspection of figure (3.3a) indicates that the primary disadvantage of first-order splines is that they are not smooth. In essence, the slope changes abruptly. In formal terms, the first derivative of the function is discontinuous at these points. This deficiency is overcome by using higher-order polynomial splines that ensure smoothness by equating derivatives at these points, as discussed in the next section.

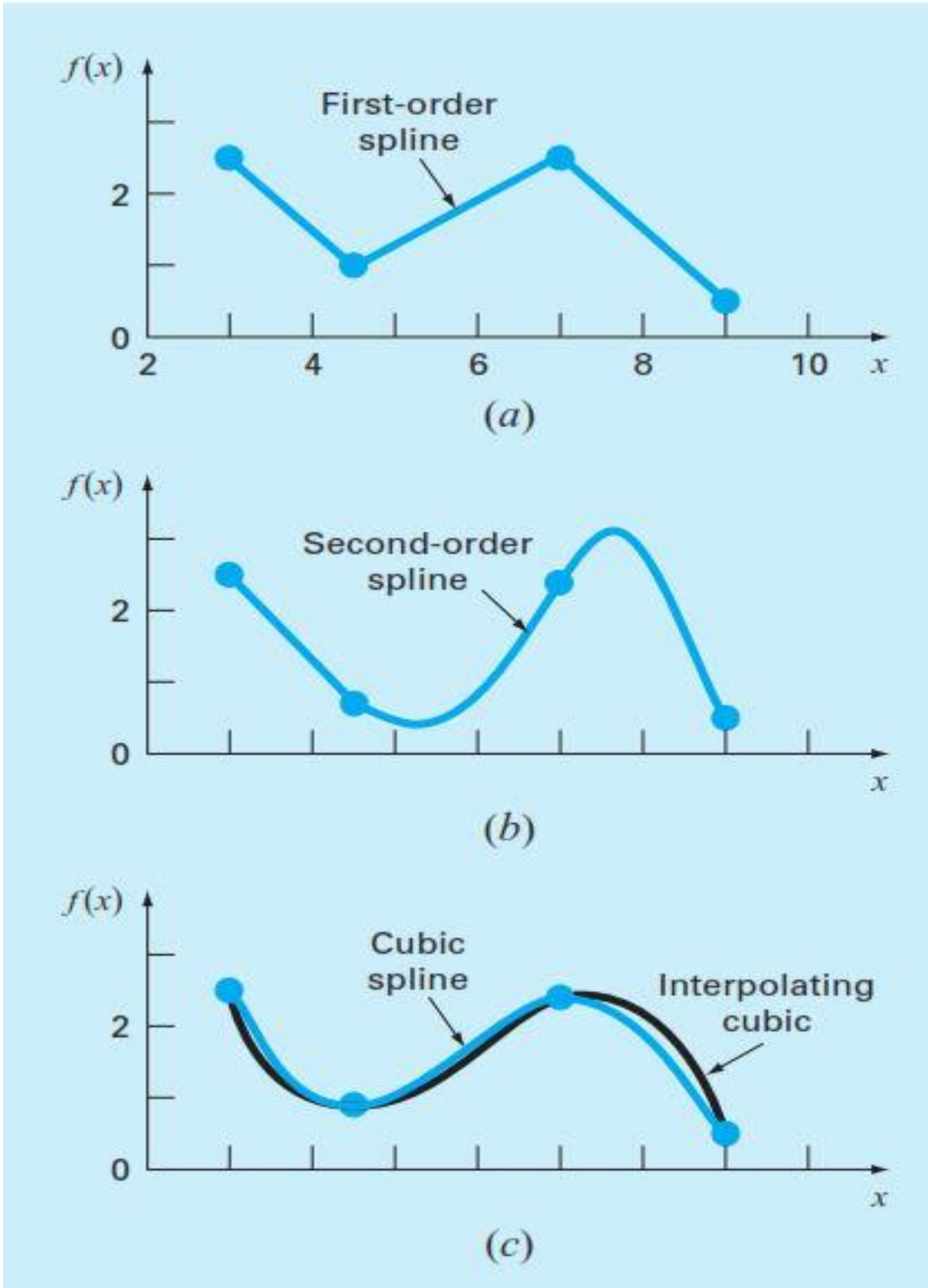

Figure (3.3) Spline fits of a set of four points. (*a*) Linear spline, (*b*) quadratic spline, and (*c*) cubic spline, with a cubic interpolating polynomial also plotted

2- Quadratic spline

To ensure that the nth derivatives are continuous at the points, a spline of at least $n + 1$ order must be used. Third-order polynomials or cubic splines that ensure continuous first and second derivatives are most frequently used in practice.

Because the derivation of cubic splines is somewhat involved, we have chosen to include them in a subsequent section. We have decided to first illustrate the concept of spline interpolation using second-order polynomials. These "quadratic splines" have continuous first derivatives at the points. Although quadratic splines do not ensure equal second derivatives at the points, they serve nicely to demonstrate the general procedure for developing higher-order splines.

The objective in quadratic splines is to derive a second-order polynomial for each interval between data points. The polynomial for each interval can be represented generally as

$$f_i(x) = a_i x^2 + b_i x + c_i$$

Figure (3.4) has been included to help clarify the notation. Notice that there are (n) intervals and $(n + 1)$ data points. The example shown in the figure is for $n = 3$.

For ***n* + 1 data points** ($i = 0, 1, 2, \ldots, n$), there are ***n* intervals** and, consequently, **3*n* unknown constants** (the a's, b's, and c's) to evaluate. Therefore, **3*n* equations** or conditions are required to evaluate the unknowns. These are:

1- The function values of adjacent polynomials must be equal at the interior knots. This condition can be represented as

$$a_{i-1}x_{i-1}^2 + b_{i-1}x_{i-1} + c_{i-1} = f(x_{i-1})$$

$$a_i x_{i-1}^2 + b_i x_{i-1} + c_i = f(x_{i-1})$$

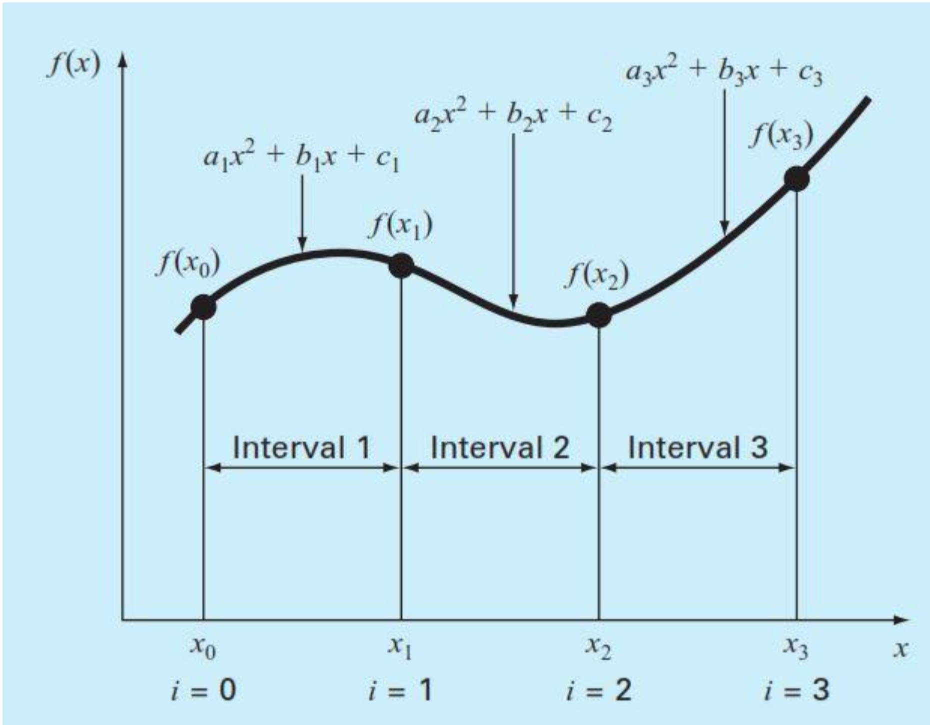

Figure (3.4) Notation used to derive quadratic splines

2- The first and last functions must pass through the end points. This adds two additional equations:

$$a_1 x_0^2 + b_1 x_0 + c_1 = f(x_0)$$

$$a_n x_n^2 + b_n x_n + c_n = f(x_n)$$

3- The first derivatives at the interior knots must be equal

$$f'(x) = 2ax + b$$

Therefore, the condition can be represented generally as

$$2a_{n-1}x_{n-1} + b_{n-1} = 2a_i x_{i-1} + b_i$$

4- Assume that the second derivative is zero at the first point. Because the second derivative is $2a_i$ this condition can be expressed mathematically as

$$a_1 = 0$$

The visual interpretation of this condition is that the first two points will be connected by a straight line.

Example (3.7)

Fit quadratic splines to the same data used in Example (3.6). Use the results to estimate the value at $x = 5$.

Solution

For the present problem, we have four data points and $n = 3$ intervals. Therefore, $3(3) = 9$ unknowns must be determined. And $2(3) - 2 = 4$ conditions

$$20.25a_1 + 4.5b_1 + c_1 = 1.0$$

$$20.25a_2 + 4.5b_2 + c_2 = 1.0$$

$$49a_2 + 7b_2 + c_2 = 2.5$$

$$49a_3 + 7b_3 + c_3 = 2.5$$

Passing the first and last functions through the initial and final values adds 2 more

$$9a_1 + 3b_1 + c_1 = 2.5$$

$$81a_3 + 9b_3 + c_3 = 0.5$$

Continuity of derivatives creates an additional $3 - 1 = 2$

$$9a_1 + b_1 = 9a_2 + b_2$$

$$14a_2 + b_2 = 14a_3 + b_3$$

Finally, $a_1 = 0$. Because we know the value of a_1 exactly, the problem reduces to solving eight simultaneous equations. These conditions can be expressed in matrix form as

$$\begin{bmatrix} 4.5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 20.25 & 4.5 & 1 & 0 & 0 & 0 \\ 0 & 0 & 49 & 7 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 49 & 7 & 1 \\ 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 81 & 9 & 1 \\ 1 & 0 & -9 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 14 & 1 & 0 & -14 & -1 & 0 \end{bmatrix} \begin{Bmatrix} b_1 \\ c_1 \\ a_2 \\ b_2 \\ c_2 \\ a_3 \\ b_3 \\ c_3 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 1 \\ 2.5 \\ 2.5 \\ 2.5 \\ 0.5 \\ 0 \\ 0 \end{Bmatrix}$$

These equations can be solved by using any method for solving a system of simultaneous linear equations in chapter one

$$a_1 = 0 \qquad b_1 = -1 \qquad c_1 = 5.5$$

$$a_2 = 0.64 \qquad b_2 = -6.76 \qquad c_2 = 18.46$$

$$a_3 = -1.6 \qquad b_3 = 24.6 \qquad c_3 = -91.3$$

Which can be substituted into the original quadratic equations to develop the following relationships for each interval:

$$f_1(x) = -x + 5.5 \qquad 3.0 \le x \le 4.5$$

$$f_2(x) = 0.64x^2 - 6.7x + 18.46 \qquad 4.5 \le x \le 7.0$$

$$f_3(x) = -1.6x^2 + 24.6x - 91.3 \qquad 7.0 \le x \le 9.0$$

When we use f_2, the prediction for $x = 5$ is, therefore,

$$f_2(5) = 0.64(5)^2 - 6.7(5) + 18.46 = 0.66$$

The total spline fit is depicted in figure (3.3b). Notice that there are two shortcomings that detract from the fit: (1) the straight line connecting the first two points and (2) the spline for the last interval seems to swing too high. The cubic splines in the next section not exhibit these shortcomings and, as a consequence, are better methods for spline interpolation.

3- Cubic spline

The objective in cubic splines is to derive a third-order polynomial for each interval between knots, as in

$$f_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i$$

Thus, for $n + 1$ data points ($i = 0, 1, 2, \ldots, n$), there are n intervals and, consequently, $4n$ unknown constants to evaluate. Just as for quadratic splines, $4n$ conditions are required to evaluate the unknowns. These are:

A. The function values must be equal at the interior knots ($2n - 2$ conditions).

B. The first and last functions must pass through the end points (2 conditions).

C. The first derivatives at the interior knots must be equal ($n - 1$ conditions).

D. The second derivatives at the interior knots must be equal ($n - 1$ conditions).

E. The second derivatives at the end knots are zero (2 conditions).

The visual interpretation of condition 5 is that the function becomes a straight line at the end knots. Specification of such an end condition leads to what is termed a “natural” spline. It is given this name because the drafting spline naturally behaves in this fashion figure (3.5). If the value of the second derivative at the end knots is nonzero (that is, there is some curvature), this information can be used alternatively to supply the two final conditions.

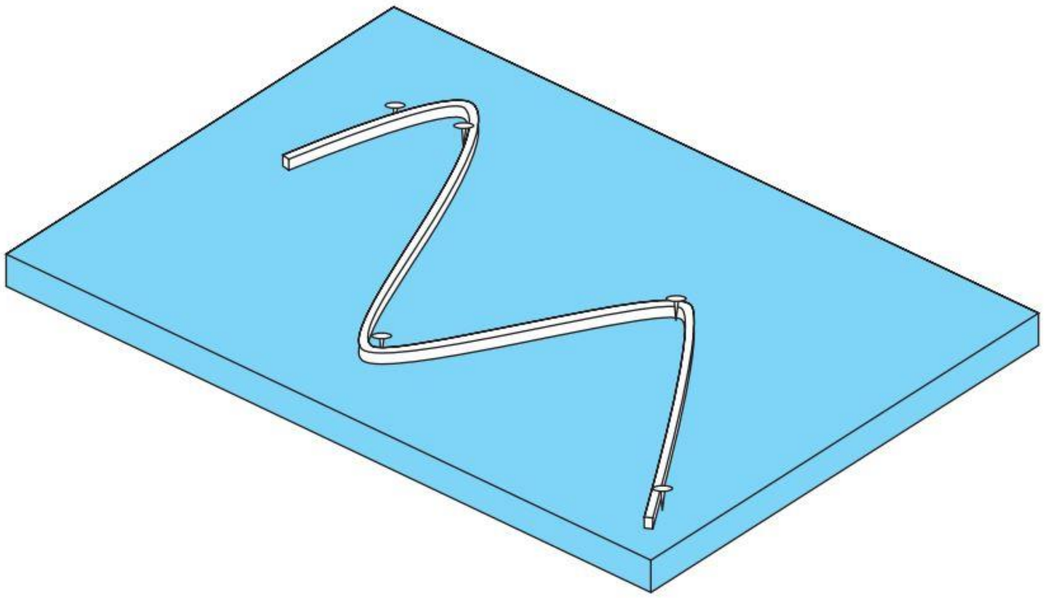

Figure (3.5) the drafting technique of using a spline to draw smooth curves through a series of points. Notice how, at the end points, the spline straightens out. This is called a "natural" spline

The above five types of conditions provide the total of 4n equations required to solve for the 4n coefficients. Whereas it is certainly possible to develop cubic splines in this fashion, we will present an alternative technique that requires the solution of only $n - 1$ equations.

The first step in the derivation is based on the observation that because each pair of knots is connected by a cubic, the second derivative within each interval is a straight line. The equation below

$$f_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i \qquad \text{(A-3)}$$

Can be differentiated twice to verify this observation. On this basis, the second derivatives can be represented by a first-order Lagrange interpolating polynomial

$$f_i''(x) = f_i''(x_{i-1}) \frac{x - x_i}{x_{i-1} - x_i} + f_i''(x_i) \frac{x - x_{i-1}}{x_i - x_{i-1}}$$

Where $f_i''(x)$ is the value of the second derivative at any point x within the ith interval. Thus, this equation is a straight line connecting the second derivative at the first knot $f_i''(x_{i-1})$ with the second derivative at the second knot $f_i''(x)$.

Next, the equation above can be integrated twice to yield an expression for *fi*(*x*). However, this expression will contain two unknown constants of integration. These constants can be evaluated by invoking the function-equality conditions $f(x)$ must equal $f(x_{i-1})$ at x_{i-1} and $f(x)$ must equal $f(x_i)$ at x_i. By performing these evaluations, the following cubic equation results:

$$f_i(x) = \frac{f_i''(x_{i-1})}{6(x_i - x_{i-1})}(x_i - x)^3 + \frac{f_i''(x_i)}{6(x_i - x_{i-1})}(x - x_{i-1})^3 + \left[\frac{f(x_{i-1})}{x_i - x_{i-1}} - \frac{f''(x_{i-1})(x_i - x_{i-1})}{6}\right](x_i - x) + \left[\frac{f(x_i)}{x_i - x_{i-1}} - \frac{f''(x_i)(x_i - x_{i-1})}{6}\right](x - x_{i-1}) \qquad \text{(B-3)}$$

Now, admittedly, this relationship is a much more complicated expression for the cubic spline for the ith interval than equation (A-3). However, notice that it contains only two unknown "coefficients," the second derivatives at the beginning and the end of the interval $f''(x_{i-1})$ and $f''(x_i)$. Thus, if we can determine the proper second derivative at each knot, equation (B-3) is a third-order polynomial that can be used to interpolate within the interval.

The second derivatives can be evaluated by invoking the condition that the first derivatives at the knots must be continuous:

$$f_i'(x_i) = f_{i+1}'(x_i) \qquad \text{(C-3)}$$

Equation (B-3) can be differentiated to give an expression for the first derivative. If this is done for both the $(i-1)$th and the ith intervals and the two results are set equal according to equation (C-3) the following relationship results:

$$(x_i - x_{i-1})f''(x_{i-1}) + 2(x_{i+1} - x_{i-1})f''(x_i) + (x_{i+1} - x_i)f''(x_{i+1}) = \frac{6}{x_{i+1} - x_i}[f(x_{i+1}) - f(x_i)] + \frac{6}{x_i - x_{i-1}}[f(x_{i-1}) - f(x_i)] \qquad \text{(D-3)}$$

If equation (D-3) is written for all interior knots, $n - 1$ simultaneous equations result with $n + 1$ unknown second derivatives. However, because this is a natural cubic spline, the second derivatives at the end knots are zero and the problem reduces to $n - 1$ equations with $n - 1$ unknowns. In addition, notice that the system of equations will be tridiagonal. Thus, not only have we reduced the number of equations but we have also cast them in a form that is extremely easy to solve.

The derivation from the above results in the following cubic equation for each interval:

$$f_i(x) = \frac{f_i''(x_{i-1})}{6(x_i - x_{i-1})}(x_i - x)^3 + \frac{f_i''(x_i)}{6(x_i - x_{i-1})}(x_{i-1} - x)^3 + \left[\frac{f(x_{i-1})}{x_i - x_{i-1}} - \frac{f''(x_{i-1})(x_i - x_{i-1})}{6}\right](x_i - x) + \left[\frac{f(x_i)}{x_i - x_{i-1}} - \frac{f''(x_i)(x_i - x_{i-1})}{6}\right](x - x_{i-1}) \qquad \text{(E-3)}$$

This equation contains only two unknowns, the second derivatives at the end of each interval. These unknowns can be evaluated using the following equation:

$$f''(x_{i-1})(x_i - x_{i-1}) + 2f''(x_i)(x_{i+1} - x_{i-1}) + f''(x_{i+1})(x_{i+1} - x_i) = \frac{6}{x_{i+1} - x_i}[f(x_{i+1}) - f(x_i)] + \frac{6}{x_i - x_{i-1}}[f(x_{i-1}) - f(x_i)] \qquad \text{(D-3)}$$

If this equation is written for all the interior knots, $n - 1$ simultaneous equations result with $n - 1$ unknowns. (Remember, the second derivatives at the end knots are zero.) The application of these equations is illustrated in the following example.

Example (3.8)

Fit cubic splines to the same data used in examples (3.6). Utilize the results to estimate the value at $x = 5$.

Solution

The first step is to employ equation (D-3) to generate the set of simultaneous equations that will be utilized to determine the second derivatives at the knots. For example, for the first interior knot, the following data is used:

$$x_0 = 3 \qquad f(x_0) = 2.5$$
$$x_1 = 4.5 \qquad f(x_1) = 1$$
$$x_2 = 7 \qquad f(x_2) = 2.5$$

These values can be substituted into equation (E-3) to yield

$$f''(3)(4.5-3) + 2f''(4.5)(7-3) + f''(7)(7-4.5) = \frac{6}{7-4.5}[2.5-1] + \frac{6}{4.5-3}[2.5-1]$$

Because of the natural spline condition, $f''(3) = 0$, and the equation reduces to

$$8f''(4.5) + 2.5f''(7) = 9.6$$

In a similar fashion, equation (D-3) can be applied to the second interior point to give

$$2.5f''(4.5) + 9f''(7) = -9.6$$

These two equations can be solved simultaneously for

$$f''(4.5) = 1.67909$$
$$f''(7) = -1.53308$$

These values can then be substituted into equation (D-3), along with values for the x's and the $f(x)$'s, to yield

$$f_1(x) = \frac{1.67909}{6(4.5-3)}(x-3)^3 + \frac{2.5}{4.5-3}(4.5-x) + \left[\frac{1}{5.4-3} - \frac{1.67909(4.5-3)}{6}\right](x-3)$$

Or

$$f_1(x) = 0.186566(x-3)^3 + 1.666667(4.5-x) + 0.246894(x-3)$$

This equation is the cubic spline for the first interval. Similar substitutions can be made to develop the equations for the second and third intervals:

$$f_2(x) = 0.111939(7-x)^3 - 0.102205(x-4.5)^3 - 0.299621(7-x) + 1.638783(x-4.5)$$

And

$$f_3(x) = -.127757(9-x)^3 + 1.761027(9-x) + 0.25(x-7)$$

The three equations can then be employed to compute values within each interval. For example, the value at $x = 5$, which falls within the second interval, is calculated as

$$f_2(x) = 0.111939(7-5)^3 - 0.102205(5-4.5)^3 - 0.299621(7-5) + 1.638783(5-4.5) = 1.102886$$

The results of examples (3.6) through (3.8) are summarized in figure (3.3). Notice the progressive improvement of the fit as we move from linear to quadratic to cubic splines. We have also superimposed a cubic interpolating polynomial on figure (3.3c). Although the cubic spline consists of a series of third-order curves, the resulting fit differs from that obtained using the third-order polynomial. This is due to the fact that the natural spline requires zero second derivatives at the end knots, whereas the cubic polynomial has no such constraint.

H.W (4)

Develop quadratic splines for the data in the table below and predict $f\,(3.4)$ and $f\,(2.2)$.

x	$F(x)$
1.6	2
2	8
2.5	14
3.2	15
4	8

H.W (5)

Develop cubic splines for the data given below and **(a)** predict $f\,(4)$ and $f\,(2.5)$ and **(b)** verify that $f_2\,(3)$ and $f_3(3)\; = 19$.

x	$F(x)$
1	3
2	6
3	19
5	99
7	291
8	444