CHAPTER 18

Interpolation

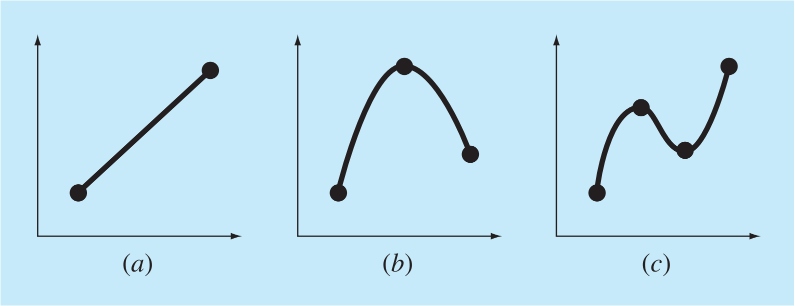
You will frequently have occasion to estimate intermediate values between precise data points. The most common method used for this purpose is polynomial interpolation. Recall that the general formula for an nth-order polynomial is

$$f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n \tag{18.1}$$

For $n + 1$ data points, there is one and only one polynomial of order n that passes through all the points. For example, there is only one straight line (that is, a first-order polynomial) that connects two points (Fig. 18.1*a*). Similarly, only one parabola connects a set of three points (Fig. 18.1*b*). *Polynomial interpolation* consists of determining the unique nth-order polynomial that fits $n + 1$ data points. This polynomial then provides a formula to compute intermediate values.

Although there is one and only one nth-order polynomial that fits $n + 1$ points, there are a variety of mathematical formats in which this polynomial can be expressed. In this chapter, we will describe two alternatives that are well-suited for computer implementation: the Newton and the Lagrange polynomials.

FIGURE 18.1
Examples of interpolating polynomials: (*a*) first-order (linear) connecting two points, (*b*) second-order (quadratic or parabolic) connecting three points, and (*c*) third-order (cubic) connecting four points.



18.2 LAGRANGE INTERPOLATING POLYNOMIALS

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

$$f_n(x) = \sum_{i=0}^{n} L_i(x)f(x_i) \tag{18.20}$$

where

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \tag{18.21}$$

where Π designates the "product of." For example, the linear version ($n = 1$) is

$$f_1(x) = \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1) \tag{18.22}$$

and the second-order version is

$$f_2(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f(x_0) + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f(x_1) + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f(x_2) \tag{18.23}$$

Equation (18.20) can be derived directly from Newton's polynomial (Box 18.1). However, the rationale underlying the Lagrange formulation can be grasped directly by realizing that each term $L_i(x)$ will be 1 at $x = x_i$ and 0 at all other sample points (Fig. 18.10). Thus, each product $L_i(x) f(x_i)$ takes on the value of $f(x_i)$ at the sample point x_i. Consequently, the summation of all the products designated by Eq. (18.20) is the unique nth-order polynomial that passes exactly through all $n + 1$ data points.

EXAMPLE 18.6 Lagrange Interpolating Polynomials

Problem Statement. Use a Lagrange interpolating polynomial of the first and second order to evaluate ln 2 on the basis of the data given in Example 18.2:

$x_0 = 1 \quad f(x_0) = 0$

$x_1 = 4 \quad f(x_1) = 1.386294$

$x_2 = 6 \quad f(x_2) = 1.791760$

Solution. The first-order polynomial [Eq. (18.22)] can be used to obtain the estimate at $x = 2$,

$$f_1(2) = \frac{2 - 4}{1 - 4} 0 + \frac{2 - 1}{4 - 1} 1.386294 = 0.4620981$$

In a similar fashion, the second-order polynomial is developed as [Eq. (18.23)]

$$f_2(2) = \frac{(2 - 4)(2 - 6)}{(1 - 4)(1 - 6)} 0 + \frac{(2 - 1)(2 - 6)}{(4 - 1)(4 - 6)} 1.386294 + \frac{(2 - 1)(2 - 4)}{(6 - 1)(6 - 4)} 1.791760 = 0.5658444$$

As expected, both these results agree with those previously obtained using Newton's interpolating polynomial.