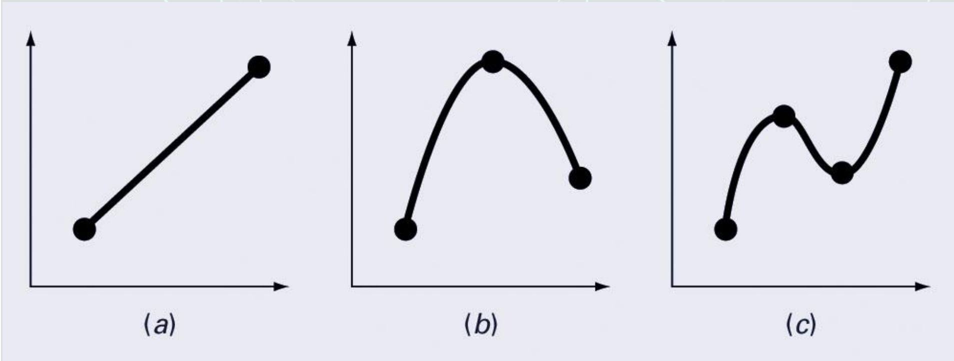
ریاضیات عالی پیشرفته

Numerical Methods for Engineers

مدرس دکتر پدرام پیوندی

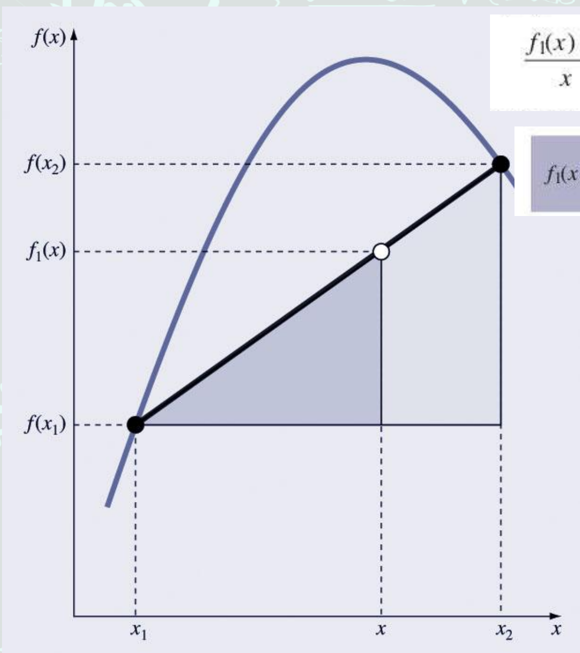
Interpolation

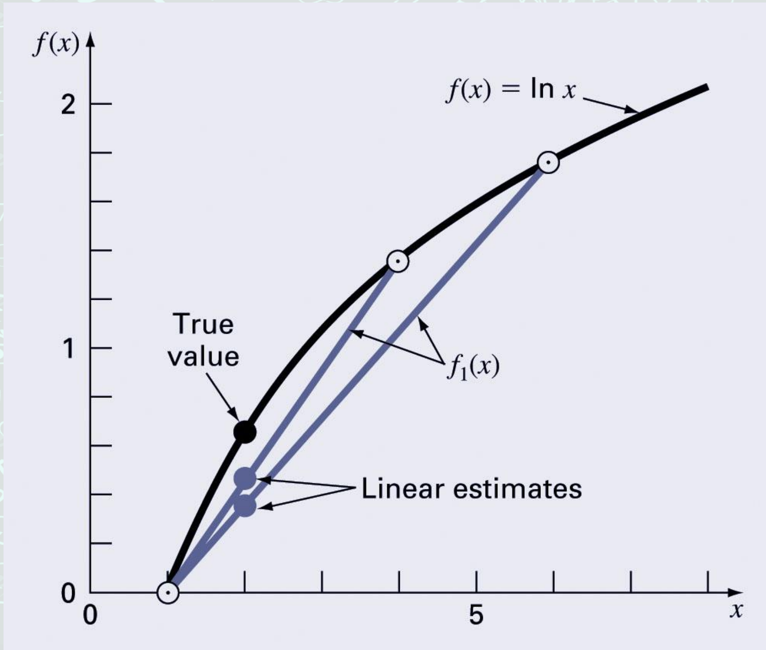
$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$



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

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





$$f_2(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1)$$

$$f_2(x) = b_0 + b_1 x - b_1 x_0 + b_2 x^2 + b_2 x_0 x_1 - b_2 x x_0 - b_2 x x_1$$

$$f_2(x) = a_0 + a_1 x + a_2 x^2$$

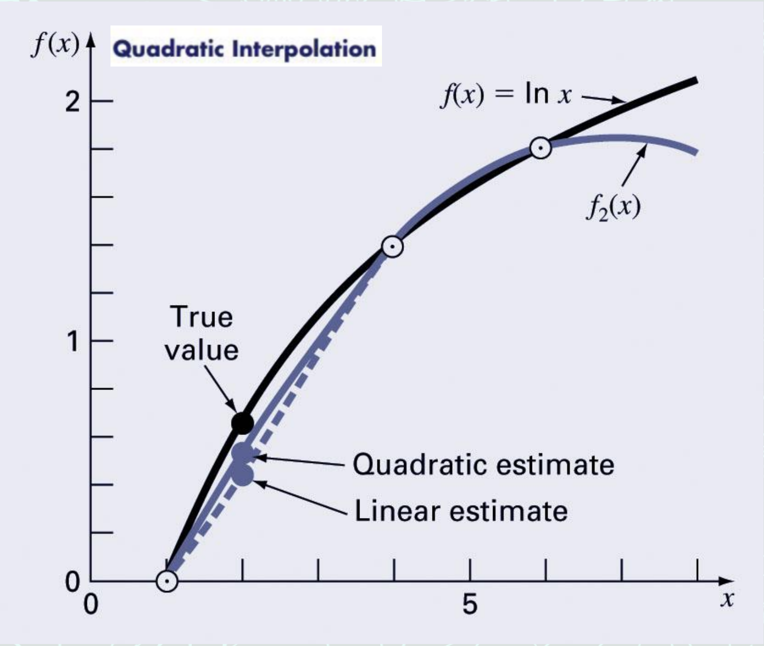
$$a_0 = b_0 - b_1 x_0 + b_2 x_0 x_1$$
$$a_1 = b_1 - b_2 x_0 - b_2 x_1$$
$$a_2 = b_2$$

$$b_0 = f(x_0)$$

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

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

6



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

$$b_0 = f(x_0)$$
$$b_1 = f[x_1, x_0]$$
$$b_2 = f[x_2, x_1, x_0]$$
$$\vdots$$
$$b_n = f[x_n, x_{n-1}, \ldots, x_1, x_0]$$

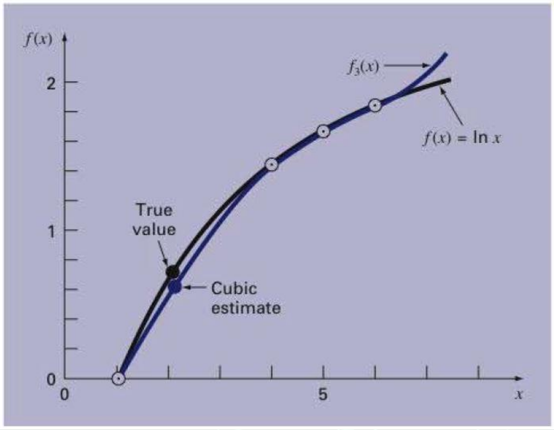
$$f[x_i, x_j] = \frac{f(x_i) - f(x_j)}{x_i - x_j}$$

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

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

8

x_i	$f(x_i)$	First	Second	Third
x_1	$f(x_1)$	$f[x_2, x_1]$	$f[x_3, x_2, x_1]$	$f[x_4, x_3, x_2, x_1]$
x_2	$f(x_2)$	$f[x_3, x_2]$	$f[x_4, x_3, x_2]$	
x_3	$f(x_3)$	$f[x_4, x_3]$		
x_4	$f(x_4)$			




```
SUBROUTINE NewtInt (x, y, n, xi, yint, ea)
  LOCAL fdd_{n,n}
  DOFOR i = 0, n
    fdd_{i,0} = y_i
  END DO
  DOFOR j = 1, n
    DOFOR i = 0, n − j
      fdd_{i,j} = (fdd_{i+1,j−1} − fdd_{i,j−1})/(x_{i+j} − x_i)
    END DO
  END DO
  xterm = 1
  yint_0 = fdd_{0,0}
  DOFOR order = 1, n
    xterm = xterm * (xi − x_{order−1})
    yint2 = yint_{order−1} + fdd_{0,order} * xterm
    ea_{order−1} = yint2 − yint_{order−1}
    yint_{order} = yint2
  END order
END NewtInt
```

LAGRANGE INTERPOLATING POLYNOMIALS

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

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

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

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

11

$x_0 = 1$	$f(x_0) = 0$
$x_1 = 4$	$f(x_1) = 1.386294$
$x_2 = 6$	$f(x_2) = 1.791760$

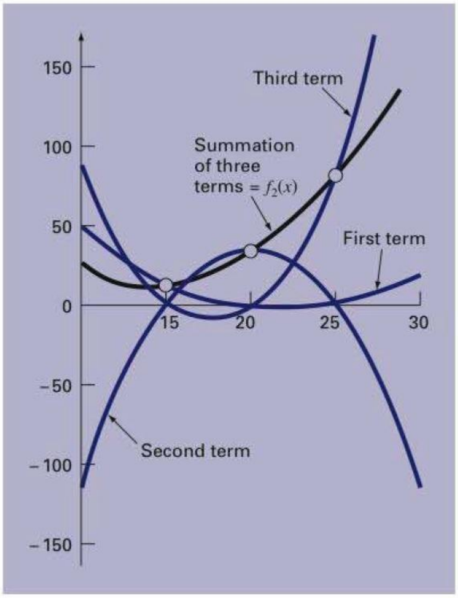
Solution. The first-order polynomial [Eq. (18.22)] can be used to obtain the estimate at $x = 2$,

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

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

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

12




```
FUNCTION Lagrng(x, y, n, xx)
  sum = 0
  DOFOR i = 0, n
    product = y_i
    DOFOR j = 0, n
      IF i ≠ j THEN
        product = product*(xx − x_j)/(x_i − x_j)
      ENDIF
    END DO
    sum = sum + product
  END DO
  Lagrng = sum
END Lagrng
```

13

SPLINE INTERPOLATION

Linear Splines

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

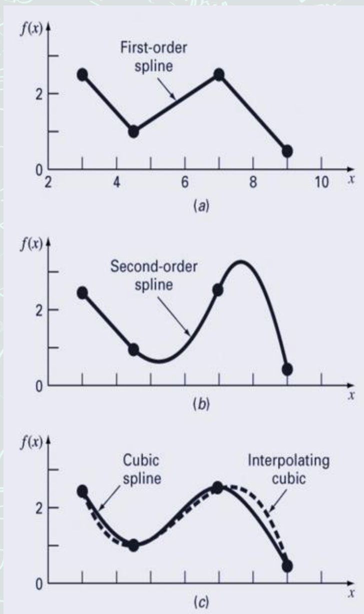
$$f(x) = f(x_1) + m_1(x - x_1) \qquad x_1 \le x \le x_2$$

.

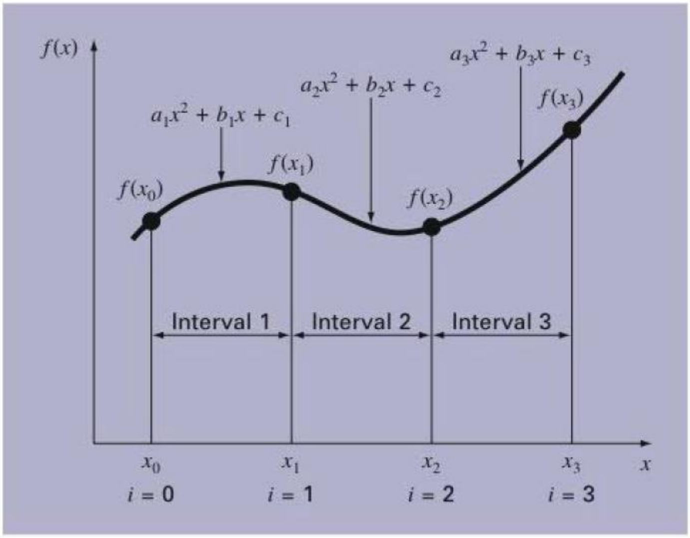
.

.

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



14



15

Quadratic 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

$$f_i(x) = a_i x^2 + b_i x + c_i \tag{18.28}$$

Figure 18.17 has been included to help clarify the notation. For $n + 1$ data points ($i = 0, 1, 2, \ldots, n$), there are n intervals and, consequently, $3n$ unknown constants (the a's, b's, and c's) to evaluate. Therefore, $3n$ 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

$$a_{i-1}x_{i-1}^2 + b_{i-1}x_{i-1} + c_{i-1} = f(x_{i-1}) \tag{18.29}$$

$$a_i x_{i-1}^2 + b_i x_{i-1} + c_i = f(x_{i-1}) \tag{18.30}$$

for $i = 2$ to n. Because only interior knots are used, Eqs. (18.29) and (18.30) each provide $n - 1$ conditions for a total of $2n - 2$ conditions.

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

$$a_1 x_0^2 + b_1 x_0 + c_1 = f(x_0) \tag{18.31}$$

$$a_n x_n^2 + b_n x_n + c_n = f(x_n) \tag{18.32}$$

for a total of $2n - 2 + 2 = 2n$ conditions.

16

3. *The first derivatives at the interior knots must be equal.* The first derivative of Eq. (18.28) is

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

Therefore, the condition can be represented generally as

$$2a_{i-1}x_{i-1} + b_{i-1} = 2a_i x_{i-1} + b_i \quad (18.33)$$

for $i = 2$ to n. This provides another $n - 1$ conditions for a total of $2n + n - 1 = 3n - 1$. Because we have $3n$ unknowns, we are one condition short. Unless we have some additional information regarding the functions or their derivatives, we must make an arbitrary choice to successfully compute the constants. Although there are a number of different choices that can be made, we select the following:

4. *Assume that the second derivative is zero at the first point.* Because the second derivative of Eq. (18.28) is $2a_i$, this condition can be expressed mathematically as

$$a_1 = 0 \quad (18.34)$$

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

17

Cubic Splines

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

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:

1. The function values must be equal at the interior knots ($2n - 2$ conditions).
2. The first and last functions must pass through the end points (2 conditions).
3. The first derivatives at the interior knots must be equal ($n - 1$ conditions).
4. The second derivatives at the interior knots must be equal ($n - 1$ conditions).
5. 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 (Fig. 18.15). 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.

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. Although the derivation of this method (Box 18.3) is somewhat less straightforward than that for quadratic splines, the gain in efficiency is well worth the effort.

18

MULTIDIMENSIONAL INTERPOLATION

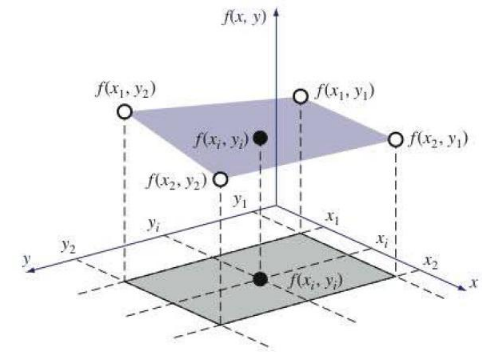


FIGURE 18.19
Graphical depiction of two-dimensional bilinear interpolation where an intermediate value (filled circle) is estimated based on four given values (open circles).

19

Bilinear Interpolation

Two-dimensional interpolation deals with determining intermediate values for functions of two variables, $z = f(x_i, y_i)$. As depicted in Fig. 18.19, we have values at four points: $f(x_1, y_1)$, $f(x_2, y_1)$, $f(x_1, y_2)$, and $f(x_2, y_2)$. We want to interpolate between these points to estimate the value at an intermediate point $f(x_i, y_i)$. If we use a linear function, the result is a plane connecting the points as in Fig. 18.19. Such functions are called *bilinear*.

A simple approach for developing the bilinear function is depicted in Fig. 18.20. First, we can hold the y value fixed and apply one-dimensional linear interpolation in the x direction. Using the Lagrange form, the result at (x_i, y_1) is

$$f(x_i, y_1) = \frac{x_i - x_2}{x_1 - x_2} f(x_1, y_1) + \frac{x_i - x_1}{x_2 - x_1} f(x_2, y_1) \qquad (18.38)$$

and at (x_i, y_2) is

$$f(x_i, y_2) = \frac{x_i - x_2}{x_1 - x_2} f(x_1, y_2) + \frac{x_i - x_1}{x_2 - x_1} f(x_2, y_2) \qquad (18.39)$$

These points can then be used to linearly interpolate along the y dimension to yield the final result,

$$f(x_i, y_i) = \frac{y_i - y_2}{y_1 - y_2} f(x_i, y_1) + \frac{y_i - y_1}{y_2 - y_1} f(x_i, y_2) \qquad (18.40)$$

A single equation can be developed by substituting Eqs. (18.38) and (18.39) into Eq. (18.40) to give

20

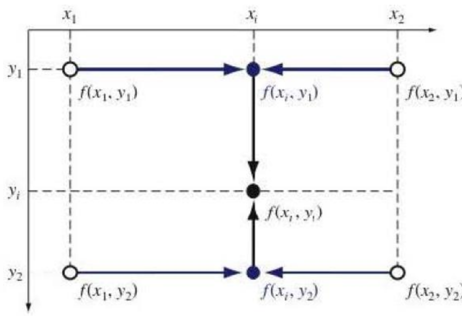


FIGURE 18.20
Two-dimensional bilinear interpolation can be implemented by first applying one-dimensional linear interpolation along the x dimension to determine values at x_i. These values can then be used to linearly interpolate along the y dimension to yield the final result at x_i, y_i.

$$f(x_i, y_i) = \frac{x_i - x_2}{x_1 - x_2}\frac{y_i - y_2}{y_1 - y_2} f(x_1, y_1) + \frac{x_i - x_1}{x_2 - x_1}\frac{y_i - y_2}{y_1 - y_2} f(x_2, y_1) + \frac{x_i - x_2}{x_1 - x_2}\frac{y_i - y_1}{y_2 - y_1} f(x_1, y_2) + \frac{x_i - x_1}{x_2 - x_1}\frac{y_i - y_1}{y_2 - y_1} f(x_2, y_2)$$

Bilinear Interpolation

Problem Statement. Suppose you have measured temperatures at a number of coordinates on the surface of a rectangular heated plate:

$T(2, 1) = 60 \qquad T(9, 1) = 57.5$

$T(2, 6) = 55 \qquad T(9, 6) = 70$

Use bilinear interpolation to estimate the temperature at $x_i = 5.25$ and $y_i = 4.8$.

Solution. Substituting these values into Eq. (18.41) gives

$$f(5.5, 4) = \frac{5.25 - 9}{2 - 9}\frac{4.8 - 6}{1 - 6}60 + \frac{5.25 - 2}{9 - 2}\frac{4.8 - 6}{1 - 6}57.5 + \frac{5.25 - 9}{2 - 9}\frac{4.8 - 1}{6 - 1}55 + \frac{5.25 - 2}{9 - 2}\frac{4.8 - 1}{6 - 1}70 = 61.2143$$

Note that beyond the simple bilinear interpolation described in the foregoing example, higher-order polynomials and splines can also be used to interpolate in two dimensions. Further, these methods can be readily extended to three dimensions. We will return to this topic when we review software applications for interpolation at the end of Chap. 19.