$-5 \le x \le 5$ in steps of .1; note that $p(x)$ does not reproduce $f(x)$ well between the knots. Repeat this problem for the same function $f(x)$, but use knots $x_k = k, 0 \le k \le 10$; note that interpolation over $[0, 10]$ is better behaved than over $[-5, 5]$. (Interpolation errors are discussed in Section 5.2.3.)

11. Suppose that P and Q are both sets of $k + 1$ points where $P \cap Q$ has k points. Suppose that $p(x)$ in $\mathcal{P}_k$ interpolates $f(x)$ at the points of P, and $q(x)$ in $\mathcal{P}_k$ interpolates $f(x)$ at the points of Q. Let α denote the only point in $P - (P \cap Q)$; and β, the only point in $Q - (P \cap Q)$. Show that $r(x)$ in $\mathcal{P}_{k+1}$ interpolates $f(x)$ on $P \cup Q$ where

$$r(x) = \frac{(x - \beta)p(x) - (x - \alpha)q(x)}{\alpha - \beta}.$$

5.2.2. Interpolation at Equally Spaced Points

In applications in which functions are given in tabular form, the tabular points are frequently equally spaced (for example, in tables of experimental data or in trigonometric and logarithmic tables). For problems of interpolating functions or data at equally spaced points there are several simplifications in the Newton form of the interpolating polynomial. As clarification, suppose that the interpolation points $x_0, x_1, \ldots, x_n$ are equally spaced in $[a, b]$ with $x_0 = a$ and $x_n = b$. In this case, we have

$$x_i = x_0 + ih, \qquad h = (b - a)/n, \qquad 0 \le i \le n \tag{5.13}$$

(equivalently, $x_i = x_{i-1} + h$, $1 \le i \le n$).

For equally spaced data, divided differences simplify somewhat. For example since $x_{i+1} - x_i = h$, $x_{i+2} - x_i = 2h$, and $x_{i+3} - x_i = 3h$, after some calculation we have

$$\begin{aligned} f[x_i, x_{i+1}] &= \frac{f(x_{i+1}) - f(x_i)}{h}, \\ f[x_i, x_{i+1}, x_{i+2}] &= \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{2h^2}, \\ f[x_i, x_{i+1}, x_{i+2}, x_{i+3}] &= \frac{f(x_{i+3}) - 3f(x_{i+2}) + 3f(x_{i+1}) - f(x_i)}{6h^3}. \end{aligned} \tag{5.14}$$

The idea of forward differences can be used to exploit the simplications afforded by equally spaced data. For any function $f(x)$ and for a fixed step-size h, we define the forward difference operator Δ by

$$\Delta f(x) = f(x + h) - f(x). \tag{5.15}$$

In (5.15) the operator Δ associates a new function $f(x + h) - f(x)$ with a given function $f(x)$. Thus for example if $f(x) = \cos(x)$ and $h = 0.1$, then $\Delta f(x)$ is the function $\Delta f(x) = \cos(x + 0.1) - \cos(x)$. Higher-order forward differences are defined recursively by

$$\Delta^k f(x) \equiv \Delta(\Delta^{k-1} f(x)), \qquad k = 1, 2, \ldots \tag{5.16}$$

where we set $\Delta^0 f(x) \equiv f(x)$. For example,

$$\begin{aligned}\Delta^2 f(x) &= \Delta(\Delta f(x)) = \Delta(f(x + h) - f(x)) \\ &= (f(x + 2h) - f(x + h)) - (f(x + h) - f(x)) \\ &= f(x + 2h) - 2f(x + h) + f(x).\end{aligned}$$

Similarly we can easily verify that

$$\Delta^3 f(x) = f(x + 3h) - 3f(x + 2h) + 3f(x + h) - f(x).$$

A pattern soon emerges for higher-order forward differences, and we can easily prove that

$$\Delta^k f(x) = \sum_{i=0}^{k} \binom{k}{i} (-1)^i f(x + (k - i)h) \tag{5.17}$$

where $\binom{k}{i}$ is the usual symbol for binomial coefficients,

$$\binom{k}{i} = \frac{k!}{i!(k - i)!}$$

and where $\binom{k}{0} \equiv \binom{k}{k} \equiv 1$.

The connection between forward differences and divided differences is fairly direct. We observe that since $\Delta f(x) = f(x + h) - f(x)$, then $\Delta f(x_i) = f(x_{i+1}) - f(x_i)$ or

$$f[x_i, x_{i+1}] = \frac{\Delta f(x_i)}{h}. \tag{5.18}$$

It is easy to establish the general formula

$$f[x_i, x_{i+1}, \ldots, x_{i+k}] = \frac{\Delta^k f(x_i)}{k!h^k} \tag{5.19}$$

by induction. Specifically, (5.19) is true for $k = 1$ by (5.18). Assuming (5.19) is true for $k = j - 1$, consider

$$f[x_i, x_{i+1}, \ldots, x_{i+j}] = \frac{f[x_{i+1}, x_{i+2}, \ldots, x_{i+j}] - f[x_i, x_{i+1}, \ldots, x_{i+j-1}]}{x_{i+j} - x_i}. \tag{5.20}$$

But $x_{i+j} - x_i = jh$ and, given the validity of (5.19) for $k = j - 1$, (5.20) reduces to

$$f[x_i, x_{i+1}, \ldots, x_{i+j}] = \frac{\dfrac{\Delta^{j-1} f(x_{i+1})}{(j - 1)!h^{j-1}} - \dfrac{\Delta^{j-1} f(x_i)}{(j - 1)!h^{j-1}}}{jh}. \tag{5.21}$$

From (5.21), it follows that

$$f[x_i, x_{i+1}, \ldots, x_{i+j}] = \frac{\Delta^{j-1}f(x_{i+1}) - \Delta^{j-1}f(x_i)}{j!h^j} = \frac{\Delta^{j-1}f(x_i + h) - \Delta^{j-1}f(x_i)}{j!h^j};$$

and the definition (5.16) shows that (5.19) is valid for $k = j$.

Having shown how divided differences simplify when the interpolation points are equally spaced, we now turn our attention to the Newton form of the polynomial interpolating $f(x)$ at $x_0, x_1, \ldots, x_n$. From the last section, this polynomial $p(x)$ is given by

$$\begin{aligned} p(x) = f(x_0) &+ f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \cdots \\ &+ f[x_0, x_1, \ldots, x_n](x - x_0)(x - x_1) \ldots (x - x_{n-1}). \end{aligned} \tag{5.22}$$

In order to incorporate equally spaced knots into (5.22) we make the change of variable

$$x = x_0 + rh.$$

Here r is any number (not necessarily an integer), and in this regard r is a dimensionless variable and measures the directed distance from x_0 to x in units of the step size h. (For example with $r = 1.5$, $x = x_0 + rh$ is halfway between x_1 and x_2.) If we consider a typical term in the Newton form for $p(x)$ in (5.22),

$$f[x_0, x_1, \ldots, x_j](x - x_0)(x - x_1) \ldots (x - x_{j-1}),$$

we see that we can use (5.19) and the substitution $x = x_0 + rh$ to simplify this term. In particular, $x - x_0 = rh$, $x - x_1 = (r - 1)h$, $x - x_2 = (r - 2)h, \ldots$, and $x - x_{j-1} = (r - (j - 1))h$; and therefore we can write

$$f[x_0, x_1, \ldots, x_j](x - x_0)(x - x_1) \ldots (x - x_{j-1}) = \frac{\Delta^j f(x_0)}{j!} r(r - 1)(r - 2) \ldots (r - (j - 1)).$$

In this expression the factor $r(r - 1)(r - 2) \ldots (r - (j - 1))/j!$ is to be regarded as a polynomial of degree j in the variable r. It is conventional to denote this polynomial by the following binomial coefficient notation:

$$\binom{r}{j} = \frac{r(r - 1)(r - 2) \ldots (r - (j - 1))}{j!}, \qquad j \geq 1, \binom{r}{0} \equiv 1.$$

For example,

$$\binom{r}{1} = r, \qquad \binom{r}{2} = \frac{r(r - 1)}{2}, \qquad \binom{r}{3} = \frac{r(r - 1)(r - 2)}{6}.$$

Using this notation in the expression above, we can write the Newton form of $p(x)$ in (5.22) as

$$p(x) = p(x_0 + rh) = f(x_0) + \Delta f(x_0)\binom{r}{1} + \Delta^2 f(x_0)\binom{r}{2} + \cdots + \Delta^n f(x_0)\binom{r}{n},$$

or more briefly as

$$p(x_0 + rh) = \sum_{i=0}^{n} \Delta^i f(x_0)\binom{r}{i}. \tag{5.23}$$

The representation (5.23) is known as *Newton's forward formula* for the interpolating polynomial.

There are efficient ways to evaluate the Newton forward formula, given the numbers $\Delta^i f(x_0)$ (see Problem 10). In addition, the required differences, $\Delta^i f(x_0)$, are easily generated from a table (see Table 5.2).

Note that the table of forward differences is generated one column at a time, from left to right. For example, $\Delta f(x_1) = f(x_2) - f(x_1)$ and $\Delta^2 f(x_0) = \Delta f(x_1) - \Delta f(x_0)$. Note also that in order to compute $\Delta^{n+1} f(x_0)$, we need not start over. Instead, we merely compute one more entry in each column, that is, x_{n+1}, $f(x_{n+1})$, $\Delta f(x_n)$, $\Delta^2 f(x_{n-1})$, . . . , $\Delta^n f(x_1)$ and $\Delta^{n+1} f(x_0)$. The subroutine DIFINT shown in Fig. 5.2 implements the ideas outlined above. This subroutine first calculates the forward difference table, as in Table 5.2, for the interpolation points $x_1, x_2, \ldots, x_n$. When these differences are obtained, the interpolating polynomial is evaluated at $x = x_1 + rh$; and the nested multiplication algorithm given in Problem 10 is used. For ease of programming, the range on the subscript i is $1 \le i \le n$ instead of $0 \le i \le n$.

TABLE 5.2 The forward-difference table for $f(x_0), f(x_1), \ldots, f(x_n)$.

x	$f(x)$	$\Delta f(x)$	$\Delta^2 f(x)$	$\Delta^3 f(x)$	$\cdots$	$\Delta^n f(x)$
x_0	$f(x_0)$					
		$\Delta f(x_0)$				
x_1	$f(x_1)$		$\Delta^2 f(x_0)$			
		$\Delta f(x_1)$		$\Delta^3 f(x_0)$		
x_2	$f(x_2)$		$\Delta^2 f(x_1)$			
		$\Delta f(x_2)$			$\ddots$	
x_3	$f(x_3)$					$\Delta^n f(x_0)$
$\vdots$	$\vdots$					
x_{n-1}	$f(x_{n-1})$			$\cdots$		
		$\Delta f(x_{n-1})$				
x_n	$f(x_n)$					

```
      SUBROUTINE DIFINT(X,Y,R,PXPRH,N)
      DIMENSION X(20),Y(20),P(20),Q(20)
C
C   SUBROUTINE DIFINT CONSTRUCTS AND EVALUATES THE NEWTON FORWARD FORM OF
C   THE INTERPOLATING POLYNOMIAL, P(X), AT X=X(1)+R*H, WHERE P(X(1))=Y(I)
C   AND WHERE X(I)=X(1)+(I-1)*H, I=1,2,...,N.  THE CALLING PROGRAM MUST
C   SUPPLY THE ARRAYS X(I) AND Y(I), THE NUMBER R AND THE INTEGER N.
C   THE SUBROUTINE RETURNS P(X(1)+R*H) AS PXPRH.
C
      DO 1 I=1,N
    1 P(I)=Y(I)
C
C   SET UP THE FORWARD DIFFERENCE TABLE
C
      DO 3 J=2,N
      DO 2 K=J,N
    2 Q(K)=P(K)-P(K-1)
      DO 3 K=J,N
    3 P(K)=Q(K)
C
C   EVALUATE THE FORWARD FORM OF THE INTERPOLATING
C   POLYNOMIAL BY NESTED MULTIPLICATION
C
      C=P(N)
      DO 4 J=2,N
      I=N-J
    4 C=P(I+1)+(R-I)*C/(I+1)
      PXPRH=C
      RETURN
      END
```

Figure 5.2

EXAMPLE 5.5. We consider the function $f(x) = \cos(x)$. Following Table 5.2, we construct this table.

x	$f(x)$	$\Delta f(x)$	$\Delta^2 f(x)$	$\Delta^3 f(x)$
0.3	0.955336			
		−0.034275		
0.4	0.921061		−0.009203	
		−0.043478		0.000434
0.5	0.877583		−0.008769	
		−0.052247		
0.6	0.825336			

Therefore $p(x) = 0.955336 - 0.034275\binom{r}{1} - 0.009203\binom{r}{2} + 0.000434\binom{r}{3}$. To find $p(0.44)$, since $0.44 = 0.3 + (1.4)h$, we set $r = 1.4$ in the expression above and obtain $p(0.44) = 0.904750$.

In many applications, such as those involving numerical solutions of differential equations, we are interested in interpolating at points labeled x_m, x_{m+1}, . . . , x_{m+n} instead of at points labeled x_0, x_1, . . . , x_n. Although we could rename the points $x_m, x_{m+1}, \ldots, x_{m+n}$ and then use (5.23) to find the interpolating polynomial, this procedure is both inconvenient and unnecessary. Instead, we can use the formula

$$p(x) = p(x_m + rh) = \sum_{j=0}^{n} \binom{r}{j} \Delta^j f(x_m) \tag{5.24}$$

where now $r = (x - x_m)/h$. Clearly, (5.24) is a valid formula for the interpolating polynomial since the formula can be gotten from (5.23) by relabeling points. [For example, set $z_0 = x_m$, $z_j = z_0 + jh$ and (5.24) reduces directly to (5.23).]

A somewhat different labeling problem occurs when, as often happens, we wish to interpolate at the points $x_m, x_{m-1}, \ldots, x_{m-n}$. This is a problem of "backwards interpolation" where we again want a formula in which we can add an additional point x_{m-n-1} without losing the effort we expended to interpolate at $x_m, x_{m-1}, \ldots, x_{m-n}$. The formula needed is known as *Newton's backward formula* for the interpolating polynomial:

$$p(x) = p(x_m - rh) = \sum_{j=0}^{n} \binom{r}{j} (-1)^j \Delta^j f(x_{m-j}). \tag{5.25}$$

Note that each difference $\Delta^j f(x_{m-j})$ used in (5.25) can be obtained from a difference table like that displayed in Table 5.2. In fact, when $m = n$, each difference $\Delta^j f(x_{n-j})$ is found as the *last entry* of the jth column of Table 5.2.

To show how formula (5.25) is derived, let us rename the points $\{x_m, x_{m-1}, \ldots, x_{m-n}\}$ as $\{t_m, t_{m+1}, \ldots, t_{m+n}\}$. Then $t_{m+i} = t_m + ih'$ where $h' = -h$ and where therefore $t_{m+i} = x_{m-i}$, $i = 0, 1, \ldots, n$. Using (5.24) with $x_m = t_m$, we obtain

$$p(x) = p(t_m + rh') = \sum_{j=0}^{n} \binom{r}{j} \Delta^j f(t_m). \tag{5.26}$$

To translate $\Delta^j f(t_m)$ in terms of our original labeling, note by (5.17)

$$\Delta^j f(t_m) = \sum_{i=0}^{j} \binom{j}{i} (-1)^i f(t_{m+j-i})$$

$$= \sum_{i=0}^{j} \binom{j}{i} (-1)^i f(x_{m+i-j}).$$

Next we change the index of the summation by setting $k = j - i$, and observe that $\binom{j}{i} = \binom{j}{j-i}$ and that $(-1)^i = (-1)^j(-1)^k$. With these changes, we find

$$\Delta^j f(t_m) = (-1)^j \sum_{k=0}^{j} \binom{j}{k} (-1)^k f(x_{m-k}) = (-1)^j \Delta^j f(x_{m-j})$$

where the last equality comes from using (5.17) again. When inserted into (5.26), this identity gives the Newton backward formula of (5.25).

EXAMPLE 5.6. As an illustration of how the backward and forward formulas can be set up, we consider interpolating $f(x) = e^{3x}$ at five points with $x_0 = -1$ and $h = 0.5$. As in Table 5.2, we obtain the following difference table:

x	$f(x)$	$\Delta f(x)$	$\Delta^2 f(x)$	$\Delta^3 f(x)$	$\Delta^4 f(x)$
−1	0.049787				
		0.173343			
−0.5	0.223130		0.603527		
		0.776870		2.10129	
0.	1.		2.70482		7.31599
		3.48169		9.41728	
0.5	4.48169		12.1221		
		15.6038			
1.	20.0855				

Using the backward formula (5.25) with $m = 4$ and $n = 2$, the second-degree polynomial interpolating $f(x)$ at x_4, x_3, amd x_2 is

$$p(x) = p(x_4 - rh) = f(x_4) - \binom{r}{1}\Delta f(x_3) + \binom{r}{2}\Delta^2 f(x_2).$$

For example for $x = 0.8$, we have $r = 0.4$ and $p(0.8) = 20.0855 - (0.4)(15.6038) + (-0.12)(12.1221) = 12.3893$. The forward formula (5.24), using the same three points, has $m = 2$ and $n = 2$, and gives

$$p(x) = p(x_2 + sh) = f(x_2) + \binom{s}{1}\Delta f(x_2) + \binom{s}{2}\Delta^2 f(x_2).$$

In this case, for $x = 0.8$ we have $s = 1.6$ and $p(0.8) = 1.0 + (1.6)(3.48169) + (0.48)(12.1221) = 12.3893$.

To include the tabulated information at x_1 and x_0 in our approximation to $f(0.8)$, we merely have to add the appropriate terms to the backward form of the interpolating polynomial. Thus the third-degree polynomial $p(x_4 - rh) - \binom{r}{3}\Delta^3 f(x_1)$ and the fourth-degree polynomial $p(x_4 - rh) - \binom{r}{3}\Delta^3 f(x_1) + \binom{r}{4}\Delta^4 f(x_0)$ interpolate $f(x)$ at $\{x_4, x_3, x_2, x_1\}$ and $\{x_4, x_3, x_2, x_1, x_0\}$, respectively. The respective estimates for $f(0.8)$ are $12.3893 - (0.064)(9.41728) = 11.7866$ and $11.7866 + (-0.0416)(7.31599) = 11.4823$ whereas the correct value of $f(0.8)$ is 11.0232.

PROBLEMS, SECTION 5.2.2

1. For $f(x) = x^2$ and h arbitrary, calculate $\Delta f(x)$, $\Delta^2 f(x)$, $\Delta^3 f(x)$, and $\Delta^4 f(x)$. Repeat this calculation for the function $f(x) = x^3$.

2. For $f(x) = x^3 + 1$ and $x_k = 2k$, $0 \le k \le 3$, construct the forward difference table and the interpolating polynomial $p(x)$ as in (5.23). Evaluate $p(x)$ at $x = -1, 1, 3$. [Note: Since the four data points come from a cubic polynomial, $p(x)$ is just a different representation for $f(x)$.]
3. Use the functions $f(x) = \sin(x)$ and $f(x) = \ell n(x)$, and rework Example 5.5.
4. For $x = x_0 + rh$ and $p(x)$ given by (5.23), we have

$$\frac{dp}{dx} = \frac{dp}{dr}\frac{dr}{dx} = \frac{1}{h}\frac{dp}{dr}.$$

Thus (5.23) can be used to approximate the derivative of $f(x)$; $f'(x) \simeq p'(x)$. For $n = 3$, derive such a "numerical differentiation" formula for use at the point $\alpha = x_0 + 1.5h$ by differentiating the right-hand side of (5.23). Check calculations by testing the formula on $f(x) = 1, f(x) = x, f(x) = x^2$, and $f(x) = x^3$ with $x_0 = 0$ and $h = 1$. [The formula should give the correct value for $f'(1.5)$]. Use the data in Example 5.5 to estimate $f'(.45)$ when $f(x) = \cos(x)$, and compare your estimate with the actual value.

5. If $p(x)$ interpolates $f(x)$, then we can use $\int_a^b p(x)dx$ as an approximation for $\int_a^b f(x)dx$. Use (5.23) to construct a "numerical integration" formula by choosing $n = 2$, $x_0 = a$, and $h = (b - a)/2$. That is, verify that

$$\int_a^b p(x)dx = h\int_0^2 p(x_0 + rh)dr;$$

and integrate the right-hand side of (5.23). Check your calculations by testing the formula on $f(x) = 1$, $f(x) = x$, and $f(x) = x^2$ with $a = 0$ and $b = 2$. [The formula should give the correct value for these integrals.) Use the data in Example 5.5 to estimate $\int_{.3}^{.5}\cos(x)dx$.

6. Use $n = 3$ and repeat Problem 5. Use the formula derived to estimate $\int_{.3}^{.6}\cos(x)dx$.
7. For $f(x) = x^k$, $1 \le k \le 3$, calculate $\Delta^i f(x)$ for $1 \le i \le 4$; and note that $\Delta^{i+1}f(x) \equiv 0$ for $f(x) = x^i$.
8. Show that $\Delta^m p(x) \equiv 0$ when $p(x) \in \mathcal{P}_n$ and $m \ge n + 1$. [Hint: If $p(x) \in \mathcal{P}_n$, show that $\Delta p(x)$ is a polynomial in $\mathcal{P}_{n-1}$; thus conclude that $\Delta^{n-1}p(x)$ is a linear polynomial, $\Delta^n p(x)$ is a constant, and $\Delta^{n+1}p(x) \equiv 0$.]
9. Continue Example 5.6, by adding the interpolation point $x_5 = 1.5$ and forming $\Delta^1 f(x_4), \Delta^2 f(x_3), \ldots, \Delta^5 f(x_0)$. Knowing the value of the fourth-degree interpolating polynomial at $x = 0.8$ (see Example 5.6), use the forward form to evaluate the fifth-degree polynomial interpolating $f(x)$ at $x_0, x_1, \ldots, x_5$ for $x = 0.8$. Next add the point $x_{-1} = -1.5$ and use the backwards form to evaluate (at $x = 0.8$) the sixth-degree polynomial interpolating $f(x)$ at $x_{-1}, x_0, \ldots, x_5$. Compare all these results with the estimates provided by the truncated Taylor's series of degrees 4, 5, and 6.
10. For $n = 1, 2$, and 3, verify that the following version of nested multiplication can be used to evaluate Newton's forward formula for the interpolating polynomial (5.23): Given any number r, form the numbers $C_n, C_{n-1}, \ldots, C_1$ and C_0 by

$$C_n = \Delta^n f(x_0)$$
$$C_i = \Delta^i f(x_0) + (r - i)C_{i+1}/(i + 1), \qquad i = n - 1, n - 2, \ldots, 1, 0.$$

Then $C_0 = p(x_0 + rh)$. Use this iteration to evaluate $p(0.44)$ in Example 5.5. (Note: This form of nested multiplication is valid for all n; but, in general, verification is a somewhat difficult exercise.)

5.2.3. Error of Polynomial Interpolation

We know that given a function, $f(x)$, and $(n + 1)$ distinct points $\{x_j\}_{j=0}^n$ in $[a, b]$, we can construct in $\mathcal{P}_n$ a polynomial, $p(x)$, which passes through the points $(x_j, f(x_j))$, $0 \leq j \leq n$. The question still remains as to how well $p(x)$ approximates $f(x)$ for other values of x in $[a, b]$. (See Fig. 5.3.) For example, as in Fig. 5.3, $f(x)$ may be very ill-behaved at one or more points, α; and $p(x)$ may not be a good approximation to $f(x)$ in a neighborhood of α. Thus if $e(x) \equiv f(x) - p(x)$ is defined to be the error function, a natural question is this: How large can $e(x)$ be for any x in $[a, b]$? If $f^{(n+1)}(x)$ is continuous on $[a, b]$, then the following theorem provides an answer to this question. Before this theorem is presented, it is convenient to give some notation. We let $C^{n+1}[a, b]$ denote the set of functions defined on $[a, b]$ that have a continuous $(n + 1)$st derivative on $[a, b]$, and use $\text{Spr}\{y_1, y_2, \ldots, y_n\}$ to denote the smallest interval containing $y_1, y_2, \ldots, y_n$ (thus $\text{Spr}\{1, -3, 2\} = [-3, 2]$).

Theorem 5.5

If $p(x) \in \mathcal{P}_n$ interpolates $f(x) \in C^{n+1}[a, b]$ at $(n + 1)$ distinct points $\{x_j\}_{j=0}^n$ in $[a, b]$, then

$$e(x) = f(x) - p(x) = \frac{f^{(n+1)}(\xi)}{(n + 1)!} W(x) \tag{5.27}$$

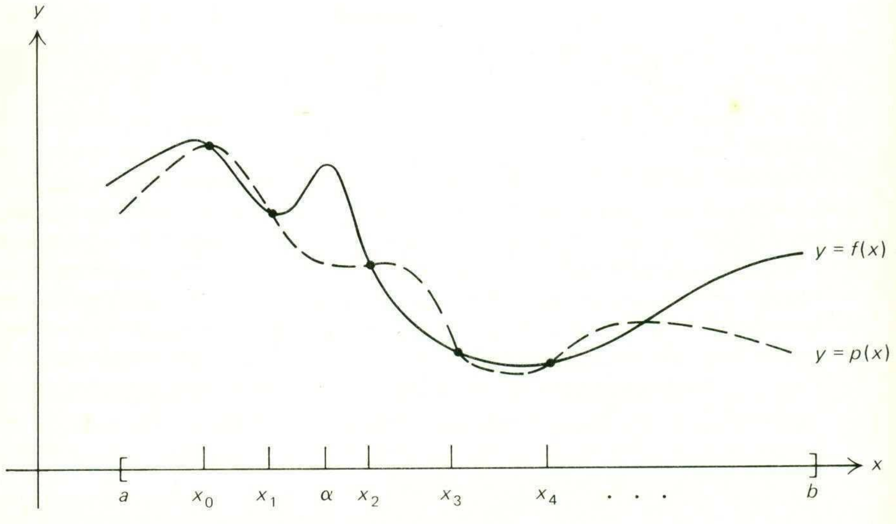

Figure 5.3 Graphs of $f(x)$ and its interpolating polynomial, $p(x)$.

where $W(x) \equiv (x - x_0)(x - x_1) \ldots (x - x_n)$ and where ξ is some point lying in $\text{Spr}\{x, x_0, x_1, \ldots, x_n\}$.

Proof. Let x be arbitrary but *fixed* in $[a, b]$. If $x = x_j$ for $0 \le j \le n$, then the theorem is trivially true. Thus we assume that $x \ne x_j$ for any j, and define the function $F(t) = f(t) - p(t) - CW(t)$ where the constant C is given by $C = (f(x) - p(x))/W(x)$. Now $F^{(n+1)}(t)$ is continuous on $[a, b]$, and furthermore $F(x) = F(x_0) = F(x_1) = \cdots = F(x_n) = 0$. By Rolle's theorem, $F'(t)$ has at least one zero between each distinct zero of $F(t)$, and thus $F'(t)$ has at least $(n + 1)$ distinct zeros in $\text{Spr}\{x, x_0, x_1, \ldots, x_n\}$. Using Rolle's theorem for $F'(t)$, we can reason that $F''(t)$ has at least n distinct zeros. Rolle's theorem applied to $F''(t)$ shows that $F'''(t)$ has at least $(n - 1)$ distinct zeros, etc. Finally we conclude that $F^{(n+1)}(t)$ has at least one zero, ξ, and $\xi \in \text{Spr}\{x, x_0, x_1, \ldots, x_n\}$. From the definition of $F(t)$ we see that $F^{(n+1)}(t) = f^{(n+1)}(t) - 0 - C(n + 1)!$. Thus, $0 = F^{(n+1)}(\xi) = f^{(n+1)}(\xi) - C(n + 1)!$. Substituting $(f(x) - p(x))/W(x)$ for C in this expression, we obtain (5.27). ∎

The reader should note that the value ξ in (5.27) changes as x changes; so ξ can be regarded as a function of x. Here is one of the drawbacks in using (5.27); since ξ is an unknown function of x, it is impossible to evalute $f^{(n+1)}(\xi)$ exactly.

However, we have assumed in Theorem 5.5 that $f^{(n+1)}(x)$ is continuous; so there is a number K_n such that $|f^{(n+1)}(\xi)| \le K_n$ for any ξ in $[a, b]$. Consequently, a more useful form of (5.27), which we can use to bound the error for any x in $[a, b]$, is

$$|e(x)| = |f(x) - p(x)| \le \frac{K_n}{(n + 1)!} |W(x)|. \tag{5.28}$$

For any particular value of x, the number $|W(x)|$ can be calculated and thus (5.28) yields an upper bound on the error. If we wish to bound $|e(x)|$ for all possible x in $[a, b]$, however, we must undertake the more formidable task of calculating

$$\max_{a \le x \le b} |W(x)| \equiv \|W\|_\infty.$$

If n and the interpolating points, $\{x_j\}_{j=0}^n$, are prescribed beforehand, then this task can be performed numerically by the methods of Chapter 4, simply by finding the zeros of $W'(x)$ and checking them (and the endpoints a and b) to find the maxima of $|W(x)|$. Most often, however, one would like the choice of n and the x_j's to be flexible in using (5.28) so that they can be selected to satisfy

$$\frac{K_n}{(n + 1)!} \|W\|_\infty < \varepsilon$$

for some predetermined tolerance, $\varepsilon > 0$. This is a fairly difficult problem and we shall examine one approach to it after the following simple example of the use of (5.28).

EXAMPLE 5.7. We shall use two previous examples involving interpolation that will show that sometimes (5.28) is a very good bound, but on other occasions it may be overly pessimistic. In Example 5.5, $f(x) = \cos(x)$ was interpolated at four points, with $W(x) = (x - 0.3)(x - 0.4)(x - 0.5)(x - 0.6)$. In this case $n = 3$ and $f^{(4)}(x) = \cos(x)$. In the notation of (5.28), we have

$$K_3 = \max_{0.3 \le x \le 0.6} |\cos(x)| = |\cos(0.3)| \le 0.955336$$

and

$$|W(0.44)| = 0.5376 \times 10^{-4}.$$

Thus

$$|e(0.44)| \le K_3|W(0.44)|/4! \le 0.214 \times 10^{-5}$$

whereas, in fact, $|e(0.44)| = 0.2 \times 10^{-5}$.

In Example 5.6, $f(x) = e^{3x}$ was interpolated at five points, with

$$W(x) = (x + 1)(x + 0.5)x(x - 0.5)(x - 1).$$

Thus $n = 4$ and $f^{(5)}(x) = 243e^{3x}$. Since the interpolation points are in $[-1, 1]$, $K_4 = 243e^3 \le 4880.79$ and $|W(0.8)| = 0.11232$. Thus,

$$|e(0.8)| \le K_4|W(0.8)|/5! \le 4.57$$

whereas, in fact, $|e(0.8)| = 0.4591$. In this second case, the error bound has overestimated the true error by a factor of about 10.

When the error bound given by (5.28) is used, it is apparent that the function $|W(x)|$ plays an important role in determining the size of the bound. Clearly, $|W(x)|$ will be small when x is near an interpolation point x_j, but more can be said about the location of values x that make $|W(x)|$ small (or large). For instance, in the second case of Example 5.7 above, $W(x) = (x + 1)(x + 0.5)x(x - 0.5)(x - 1)$ and $|W(0.8)| = 0.11232$. Note that $x = 0.2$ is no nearer an interpolation point than is $x = 0.8$, but a simple calculation shows that $|W(0.2)| = 0.04032$. Intuitively, this statement makes sense since $W(0.2)$ has more "small" factors than $W(0.8)$ has. For equally spaced interpolation points in an interval $[a, b]$, it can be shown that the relative maxima of $|W(x)|$ decrease as x approaches the midpoint $(a + b)/2$ of $[a, b]$ (see Problems 5, 6, and 8). As a general rule, (5.28) shows that interpolation at equally spaced points is more accurate near the center of the interval than near the endpoints. [However, the error is also influenced by $f^{(n+1)}(\xi)$; and while $|W(x)|$ is small, $|f^{(n+1)}(\xi)|\,|W(x)|$ can be large.] Figure 5.4 shows the graph of $W(x)$ as given in the second case of Example 5.7.

Examination of the derivation of (5.28) reveals that we can do nothing to improve the value of K_n since we do not know the location of the mean-value point ξ. However, for $\|W\|_\infty = \max_{a \le x \le b} |W(x)|$, (5.28) tells us that the inequality $|e(x)| \le K_n\|W\|_\infty/(n + 1)!$ holds for all x in $[a, b]$. Thus if we wish to make the error bound as small as possible *for all* x in $[a, b]$, we should concentrate on the

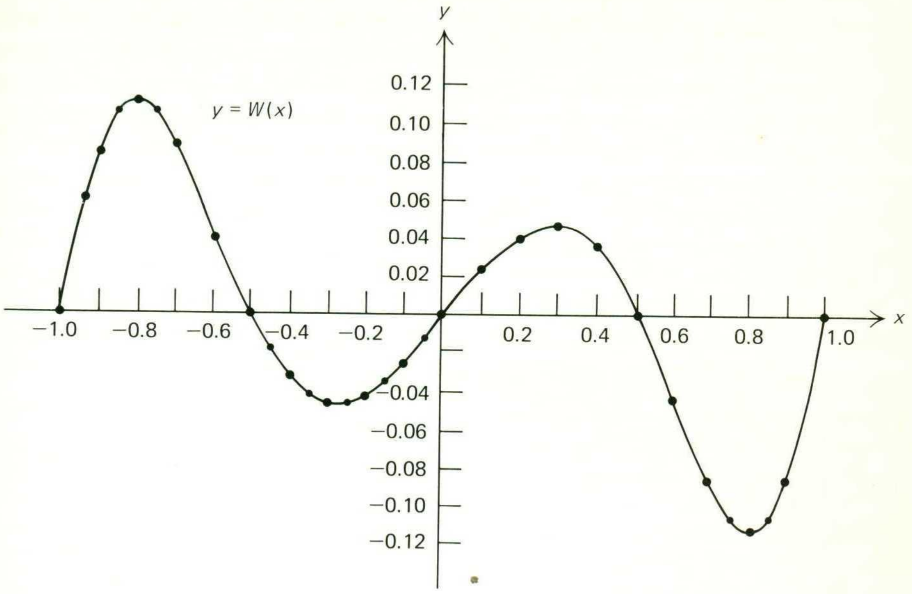

Figure 5.4 Graph of $W(x) = (x + 1)(x + 0.5)x(x - 0.5)(x - 1)$.

problem of choosing the interpolating points, $\{x_j\}_{j=0}^{n}$, so that $\|W\|_\infty$ is minimized. On the surface this choice seems to be a very difficult task. This problem was investigated by P. L. Chebyshev in the 19th century for the interval $[-1, 1]$, and the solution involves polynomials that were named after him. The solution is not too hard to obtain, and by a simple change of variable can even be extended to an arbitrary interval, $[a, b]$.

Before giving the result, we must introduce and investigate the *Chebyshev polynomials of the first kind*, $T_k(x) \equiv \cos(k \cos^{-1}(x))$, $k = 0, 1, 2, \ldots$. At first glance it probably seems difficult to believe that each $T_k(x)$ is actually an algebraic polynomial. However, we easily see that $T_0(x) \equiv 1$ and $T_1(x) \equiv x$. Now we make the variable substitution, $x = \cos(\theta)$, $0 \le \theta \le \pi$. Then $T_k(x) = T_k(\cos(\theta)) = \cos(k\theta)$. By elementary trigonometric identities, we can easily verify that

$$\begin{aligned} T_{k+1}(x) &= \cos((k+1)\theta) = 2\cos(\theta)\cos(k\theta) - \cos((k-1)\theta) \\ &= 2xT_k(x) - T_{k-1}(x), \qquad \text{for } k \ge 1. \end{aligned} \tag{5.29}$$

Therefore $T_2(x) = 2xT_1(x) - T_0(x) = 2x^2 - 1$, and $T_3(x) = 2xT_2(x) - T_1(x) = 4x^3 - 3x$, etc. We can easily verify inductively that $T_k(x)$ is a kth-degree polynomial for $k = 0, 1, 2, \ldots$. Furthermore, we see that the leading coefficient of $T_k(x)$ equals 2^{k-1}, and that $T_k(x_i) = 0$, $0 \le i \le k - 1$, when

$$x_i = \cos\frac{(2i+1)\pi}{2k}.$$

[Note: $\cos(kt) = 0$ whenever t is an odd multiple of $\pi/(2k)$.] Last we see that $|T_k(x)|$ is never larger than 1 for x in $[-1, 1]$; that is, $\|T_k\|_\infty \leq 1$. Moreover, for $y_i = \cos(i\pi/k)$, we have $T_k(y_i) = \cos(i\pi) = (-1)^i$, so $\|T_k\|_\infty = |T_k(y_i)| = 1$ at each of the $(k + 1)$ distinct points $y_0, y_1, \ldots, y_k$. Chebyshev polynomials are very important in many types of numerical approximations as we shall see in subsequent sections. The graphs of the first five Chebyshev polynomials are given in Fig. 5.5.

Now we are ready to prove the famous result for minimizing

$$\max_{-1 \leq x \leq 1} |W(x)| \equiv \|W\|_\infty.$$

Theorem 5.6

Let $W(x) = (x - x_0)(x - x_1) \ldots (x - x_n) \in \mathcal{P}_{n+1}$. Among all possible choices for distinct $\{x_j\}_{j=0}^n$ in $[-1, 1]$, $\|W\|_\infty \equiv \max_{-1 \leq x \leq 1} |W(x)|$ is minimized if $W(x) = (1/2^n)T_{n+1}(x)$ [i.e., the x_j's are the zeros of $T_{n+1}(x)$].

Proof. By the remarks above on Chebyshev polynomials, the leading coefficient of the $(n + 1)$st-degree polynomial, $(1/2^n)T_{n+1}(x)$, is 1. Therefore, $(1/2^n)T_{n+1}(x) = (x - x_0)(x - x_1) \ldots (x - x_n)$, $x_i = \cos[(2i + 1)\pi/2(n + 1)]$; and hence $W(x) = (1/2^n)T_{n+1}(x)$ is a candidate for the minimum W. Also from the

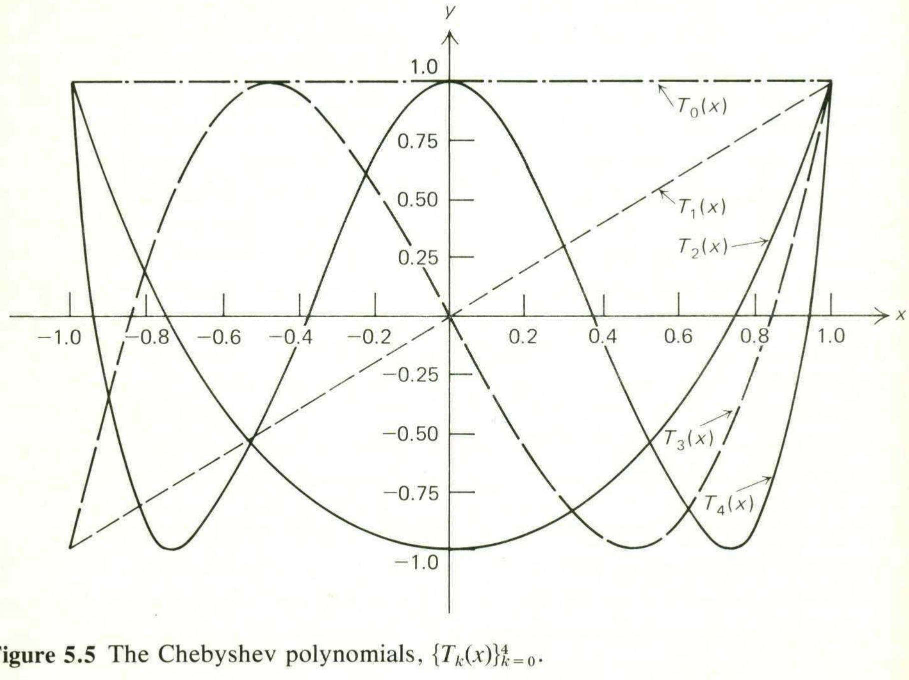

Figure 5.5 The Chebyshev polynomials, $\{T_k(x)\}_{k=0}^4$.

remarks above, $\|W\|_\infty = \|(1/2^n)T_{n+1}\|_\infty = 1/2^n$, $W(y_i) = (1/2^n)(-1)^i$, $0 \le i \le n+1$, and $W(y_i) = -W(y_{i+1})$, $0 \le i \le n$, where

$$y_i = \cos \frac{i\pi}{n+1}, \qquad 0 \le i \le n+1.$$

Now assume there exists a polynomial, $V(x) \in \mathcal{P}_{n+1}$, where $V(x)$ is monic (leading coefficient equals 1) and $\|V\|_\infty < \|W\|_\infty$. If i is even, then $V(y_i) < W(y_i) = 1/2^n$, or else $\|V\|_\infty \ge \|W\|_\infty$. If i is odd, then $V(y_i) > W(y_i) = -1/2^n$, or again $\|V\|_\infty \ge \|W\|_\infty$. Thus $V(y_0) < W(y_0)$, $V(y_1) > W(y_1)$, $V(y_2) < W(y_2)$, $V(y_3) > W(y_3)$, etc.; and so $H(x) \equiv V(x) - W(x)$ has a zero in each interval (y_i, y_{i+1}), $0 \le i \le n$. Moreover, $H(x)$ is in $\mathcal{P}_n$ since both $V(x)$ and $W(x)$ are monic. Therefore $H(x) \in \mathcal{P}_n$ and has at least $(n+1)$ zeros; so we may conclude that no such $V(x)$ exists. ■

Thus if $[a, b] = [-1, 1]$ with $f \in C^{n+1}[-1, 1]$, and if the interpolating points are the zeros of $T_{n+1}(x)$, the error bound on the interpolation given by (5.28) yields

$$|e(x)| = |f(x) - p(x)| \le \frac{K_n}{(n+1)!}\|W\|_\infty = \frac{K_n}{(n+1)!2^n}. \tag{5.30}$$

EXAMPLE 5.8. Again consider the function $f(x) = e^{3x}$ for x in $[-1, 1]$. In Example 5.7, we computed the error bound at $x = 0.8$ for interpolation at five equally spaced points ($n = 4$), and obtained a bound of 4.57. Finding this bound required us to compute $W(0.8)$ where $W(x) = (x+1)(x+0.5)x(x-0.5)(x-1)$. For interpolation at the zeros of $T_5(x)$, we can bound the error at any point x (including $x = 0.8$) by the constant $K_4/(2^4 5!)$ $= 4880.79/(2^4 5!) = 2.54$. In fact, the polynomial interpolating at the zeros of $T_5(x)$ gives a value of 11.2776 as an estimate to $f(0.8) = 11.0232$, which is an error of 0.2544 (roughly half the error of equally spaced interpolation).

To determine how many interpolation points are required to obtain a desired accuracy when interpolating $f(x)$ at the zeros of $T_{n+1}(x)$, we note that $f^{(n+1)}(x) = 3^{n+1}e^{3x}$; so $K_n = 3^{n+1}e^3$. Using (5.30), we have

$$|e(x)| \le \left(\frac{3}{2}\right)^n \frac{3e^3}{(n+1)!},$$

which can be shown to go to zero as $n \to \infty$. For example with $n = 10$, $|e(x)| \le 0.000087$; and with $n = 20$, $|e(x)| \le 0.39 \times 10^{-16}$. Using 11 equally spaced interpolation points ($n = 10$) and evaluating the error bound of formula (5.28) at $x = 0.9$ (a noninterpolation point), we find $|W(0.9)| = 0.0065$; and hence the error bound for the equally spaced case is 0.00058, which is more than six times as large as the Chebyshev interpolation error bound with $n = 10$.

Obviously we do not expect the error of Chebyshev interpolation, $|e_T(x)|$, to be smaller than the error for equally spaced interpolation, $|e_h(x)|$, for every x