BG Ch4 Bài 3. PP Nội suy Newton

Newton có câu nói nổi tiếng:

“Tôi nhìn xa hơn người khác, vì tôi đứng trên vai những người khổng lồ”

Thấm nhuần tư tưởng đó, Newton đã đưa ra ý tưởng xây dựng đa thức nội suy như sau: Xây dựng đa thức nội suy bậc n bằng cách sử dụng đa thức nội suy bậc (n – 1).

Với bộ dữ liệu đã cho

$$(x_0, f_0), \qquad (x_1, f_1), \qquad \cdots, \quad (x_n, f_n).$$

thì đa thức nội suy $p_n(x)$ thỏa mãn (1) là duy nhất, như đã được chỉ ra. Nhưng với các yêu cầu khác nhau, ta có thể sử dụng $p_n(x)$ ở các dạng thức khác nhau. Đa thức nội suy dạng Lagrange thuận lợi cho việc tính gần đúng giá trị đạo hàm và tích phân số.

Một trường hợp riêng, nhưng rất quan trọng là đa thức dạng Newton, dạng thức này thuận lợi trong việc giải gần đúng ptvp với giá trị ban đầu.

Hơn nữa, xây dựng đa thức dạng Newton yêu cầu phải tính toán ít hơn đa thức dạng Lagrange, đặc biệt khi n tăng để đạt được độ chính xác cần thiết. Khi đó, đa thức dạng Newton sử dụng tất cả các kết quả tính toán từ trước đó và cộng thêm một số hạng khác có thể không được tính từ đa thức dạng Lagrange. Số hạng này nhận được từ việc ứng dụng Nguyên lý sai số (Được sử dụng trong VD 3 của đa thức dạng Lagrange. Chi tiết của phương pháp này như sau:

Giả sử $p_{n-1}(x)$ là đa thức Newton bậc (n – 1) (Dạng của nó ta xác định sau); khi đó, $p_{n-1}(x_0) = f_0$, $p_{n-1}(x_1) = f_1$, …, $p_{n-1}(x_{n-1}) = f_{n-1}$. Từ đó, ta có thể viết đa thức Newton bậc n như sau:

(6) $p_n(x) = p_{n-1}(x) + g_n(x)$

Từ đây

(6’) $g_n(x) = p_n(x) – p_{n-1}(x)$

ở đây:

$g_n(x_k) = 0$, với k=0, 1, 2, …, n-1.

Và $p_n(x_n) = f_n$. Từ đây ta suy ra công thức của $g_n(x)$.

$g_n(x)=a_n(x-x_0)(x-x_1)\ldots(x – x_{n-1})$

Xác định a_n dựa vào công thức (6’)

$a_n = (f_n–p_{n-1}(x_n))/(x_n-x_0)(x_n-x_1)\ldots(x_n– x_{n-1})$

$$p_n(x) = p_{n-1}(x) + g_n(x); \tag{6}$$

hence

$$g_n(x) = p_n(x) - p_{n-1}(x). \tag{6'}$$

Here $g_n(x)$ is to be determined so that $p_n(x_0) = f_0, p_n(x_1) = f_1, \cdots, p_n(x_n) = f_n$.

Since p_n and p_{n-1} agree at $x_0, \cdots, x_{n-1}$, we see that g_n is zero there. Also, g_n will generally be a polynomial of nth degree because so is p_n, whereas p_{n-1} can be of degree $n-1$ at most. Hence g_n must be of the form

$$g_n(x) = a_n(x - x_0)(x - x_1) \cdots (x - x_{n-1}). \tag{6''}$$

We determine the constant a_n. For this we set $x = x_n$ and solve (6″) algebraically for a_n. Replacing $g_n(x_n)$ according to (6′) and using $p_n(x_n) = f_n$, we see that this gives

$$a_n = \frac{f_n - p_{n-1}(x_n)}{(x_n - x_0)(x_n - x_1) \cdots (x_n - x_{n-1})}. \tag{7}$$

We write a_k instead of a_n and show that a_k equals the ***k*th divided difference**, recursively denoted and defined as follows:

$$a_1 = f[x_0, x_1] = \frac{f_1 - f_0}{x_1 - x_0}$$

$$a_2 = f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$$

and in general

$$\textbf{(8)} \qquad a_k = f[x_0, \cdots, x_k] = \frac{f[x_1, \cdots, x_k] - f[x_0, \cdots, x_{k-1}]}{x_k - x_0}.$$

If $n = 1$, then $p_{n-1}(x_n) = p_0(x_1) = f_0$ because $p_0(x)$ is constant and equal to f_0, the value of $f(x)$ at x_0. Hence (7) gives

$$a_1 = \frac{f_1 - p_0(x_1)}{x_1 - x_0} = \frac{f_1 - f_0}{x_1 - x_0} = f[x_0, x_1],$$

and (6) and (6″) give the Newton interpolation polynomial of the first degree

$$p_1(x) = f_0 + (x - x_0) f[x_0, x_1].$$

If $n = 2$, then this p_1 and (7) give

$$a_2 = \frac{f_2 - p_1(x_2)}{(x_2 - x_0)(x_2 - x_1)} = \frac{f_2 - f_0 - (x_2 - x_0) f[x_0, x_1]}{(x_2 - x_0)(x_2 - x_1)} = f[x_0, x_1, x_2]$$

where the last equality follows by straightforward calculation and comparison with the definition of the right side. (Verify it; be patient.) From (6) and (6″) we thus obtain the second Newton polynomial

$$p_2(x) = f_0 + (x - x_0) f[x_0, x_1] + (x - x_0)(x - x_1) f[x_0, x_1, x_2].$$

For $n = k$, formula (6) gives

$$p_k(x) = p_{k-1}(x) + (x - x_0)(x - x_1) \cdots (x - x_{k-1}) f[x_0, \cdots, x_k]. \tag{9}$$

With $p_0(x) = f_0$ by repeated application with $k = 1, \cdots, n$ this finally gives **Newton's divided difference interpolation formula**

$$\begin{aligned} f(x) \approx f_0 &+ (x - x_0) f[x_0, x_1] + (x - x_0)(x - x_1) f[x_0, x_1, x_2] \\ &+ \cdots + (x - x_0)(x - x_1) \cdots (x - x_{n-1}) f[x_0, \cdots, x_n]. \end{aligned} \tag{10}$$

An algorithm is shown in Table 19.2. The first do-loop computes the divided differences and the second the desired value $p_n(\hat{x})$.

Example 4 shows how to arrange differences near the values from which they are obtained; the latter always stand a half-line above and a half-line below in the preceding column. Such an arrangement is called a (divided) **difference table**.

EXAMPLE 4 Newton's Divided Difference Interpolation Formula

Compute $f(9.2)$ from the values shown in the first two columns of the following table.

x_j	$f_j = f(x_j)$	$f[x_j, x_{j+1}]$	$f[x_j, x_{j+1}, x_{j+2}]$	$f[x_j, \cdots, x_{j+3}]$
8.0	2.079442			
		0.117783		
9.0	2.197225		−0.006433	
		0.108134		0.000411
9.5	2.251292		−0.005200	
		0.097735		
11.0	2.397895			

x_j	f_j	$f[x_j, x_{j+1}]$	$f[x_j, x_{j+1}, x_{j+2}]$	$f[x_j, x_{j+1}, x_{j+2}, x_{j+3}]$

-2	16	-15	7	-1
-1	1	-1	3	
0	0	8		
2	16			

$P_3(x)=16-15(x+2)+7(x+2)(x+1)-(x+2)(x+1)x$

$p_3(1)=16-45+42-6=7$

x_j	f_j	$f[x_j, x_{j+1}]$	$f[x_j, x_{j+1}, x_{j+2}]$	$f[x_j, x_{j+1}, x_{j+2}, x_{j+3}]$
0	0			
0.5	1			
1.5	5.0625			
2	16			

$p_3(x)=$

Solution. We compute the divided differences as shown. Sample computation:

$$(0.097735 - 0.108134)/(11 - 9) = -0.005200.$$

The values we need in (10) are circled. We have

$$f(x) \approx p_3(x) = 2.079442 + 0.117783(x - 8.0) - 0.006433(x - 8.0)(x - 9.0) + 0.000411(x - 8.0)(x - 9.0)(x - 9.5).$$

At $x = 9.2$,

$$f(9.2) \approx 2.079442 + 0.141340 - 0.001544 - 0.000030 = 2.219208.$$

The value exact to 6D is $f(9.2) = \ln 9.2 = 2.219203$. Note that we can nicely see how the accuracy increases from term to term:

$$p_1(9.2) = 2.220782, \qquad p_2(9.2) = 2.219238, \qquad p_3(9.2) = 2.219208.$$

Công thức sai phân tiến của Newton

Equal Spacing: Newton's Forward Difference Formula

Newton's formula (10) is valid for *arbitrarily spaced* nodes as they may occur in practice in experiments or observations. However, in many applications the x_j's are *regularly spaced*—for instance, in measurements taken at regular intervals of time. Then, denoting the distance by h, we can write

Yêu cầu: Các mốc nội suy phải cách đều

$$x_0, \quad x_1 = x_0 + h, \quad x_2 = x_0 + 2h, \quad \cdots, \quad x_n = x_0 + nh. \tag{11}$$

We show how (8) and (10) now simplify considerably!

To get started, let us define the *first forward difference* of f at x_j by

$$\Delta f_j = f_{j+1} - f_j,$$

Bảng sai phân hữu hạn tiến

j	x_j	f_j	Δf_j			$\Delta^n f_j$
0	x_0	f_0	$\Delta f_0=f_1-f_0$		…	$\Delta^n f_0=\Delta^{n-1}f_1-\Delta^{n-1}f_0$
1	x_1	f_1	$\Delta f_1=f_2-f_1$			
2	x_2	f_2				
3	x_3	f_3				
…	…	…				
n-1	x_{n-1}	f_{n-1}	$\Delta f_{n-1}=f_n-f_{n-1}$			
n	x_n	f_n				

the *second forward difference* of f at x_j by

$$\Delta^2 f_j = \Delta f_{j+1} - \Delta f_j,$$

and, continuing in this way, the **kth forward difference** of f at x_j by

(12)

$$\Delta^k f_j = \Delta^{k-1} f_{j+1} - \Delta^{k-1} f_j$$

$(k = 1, 2, \cdots)$.

Examples and an explanation of the name "forward" follow on the next page. What is the point of this? We show that if we have regular spacing (11), then

(13)

$$f[x_0, \cdots, x_k] = \frac{1}{k!h^k} \Delta^k f_0.$$

PROOF We prove (13) by induction. It is true for $k = 1$ because $x_1 = x_0 + h$, so that

$$f[x_0, x_1] = \frac{f_1 - f_0}{x_1 - x_0} = \frac{1}{h}(f_1 - f_0) = \frac{1}{1!h} \Delta f_0.$$

Assuming (13) to be true for all forward differences of order k, we show that (13) holds for $k + 1$. We use (8) with $k + 1$ instead of k; then we use $(k + 1)h = x_{k+1} - x_0$, resulting from (11), and finally (12) with $j = 0$, that is, $\Delta^{k+1} f_0 = \Delta^k f_1 - \Delta^k f_0$. This gives

$$f[x_0, \cdots, x_{k+1}] = \frac{f[x_1, \cdots, x_{k+1}] - f[x_0, \cdots, x_k]}{(k+1)h}$$

$$= \frac{1}{(k+1)h}\left[\frac{1}{k!h^k}\Delta^k f_1 - \frac{1}{k!h^k}\Delta^k f_0\right]$$

$$= \frac{1}{(k+1)!h^{k+1}}\Delta^{k+1} f_0$$

which is (13) with $k+1$ instead of k. Formula (13) is proved.

In (10) we finally set $x = x_0 + rh$. Then $x - x_0 = rh$, $x - x_1 = (r-1)h$ since $x_1 - x_0 = h$, and so on. With this and (13), formula (10) becomes **Newton's** (or *Gregory*[2]*–Newton's*) **forward difference interpolation formula**

$$f(x) \approx p_n(x) = \sum_{s=0}^{n} \binom{r}{s} \Delta^s f_0 \qquad (x = x_0 + rh, \quad r = (x - x_0)/h)$$

(14)

$$= f_0 + r\Delta f_0 + \frac{r(r-1)}{2!}\Delta^2 f_0 + \cdots + \frac{r(r-1)\cdots(r-n+1)}{n!}\Delta^n f_0$$

Công thức Newton					
j	xj	fj	Δfj	Δ2fj	Δ3fj
0	8	2.07944154	0.11778304	-0.01242252	0.00237218
1	9	2.19722458	0.10536052	-0.01005034	
2	10	2.30258509	0.09531018		
3	11	2.39789527			

$p_3(x)$=2.07944154+(x-8)*
0.11778304+(x-8)(x-9)
*(-0.01242252/2)+(x-8)(x-9)(x-
10)*(0.00237218/6)

$p_3(9.2)=2.219226576$

$\ln 9.2 \approx 2.21919618$

$\ln 9.2 = 2.21920348$

where the **binomial coefficients** in the first line are defined by

$$\binom{r}{0} = 1, \quad \binom{r}{s} = \frac{r(r-1)(r-2)\cdots(r-s+1)}{s!} \tag{15}$$

($s > 0$, integer)

and $s! = 1 \cdot 2 \cdots s$.

Error. From (5) we get, with $x - x_0 = rh, \quad x - x_1 = (r-1)h$, etc.,

$$\epsilon_n(x) = f(x) - p_n(x) = \frac{h^{n+1}}{(n+1)!} r(r-1) \cdots (r-n) f^{(n+1)}(t) \tag{16}$$

with t as characterized in (5).

[2]JAMES GREGORY (1638–1675), Scots mathematician, professor at St. Andrews and Edinburgh. Δ in (14) and ∇^2 (on p. 818) have nothing to do with the Laplacian.

Formula (16) is an exact formula for the error, but it involves the unknown t. In Example 5 (below) we show how to use (16) for obtaining an error estimate and an interval in which the true value of $f(x)$ must lie.

Comments on Accuracy. **(A)** The order of magnitude of the error $\epsilon_n(x)$ is about equal to that of the next difference not used in $p_n(x)$.

(B) One should choose $x_0, \cdots, x_n$ such that the x at which one interpolates is as well centered between $x_0, \cdots, x_n$ as possible.

The reason for (A) is that in (16),

$$f^{n+1}(t) \approx \frac{\Delta^{n+1} f(t)}{h^{n+1}}, \qquad \frac{|r(r-1)\ \cdots\ (r-n)|}{1 \cdot 2 \cdots (n+1)} \leqq 1 \quad \text{if} \quad |r| \leqq 1$$

(and actually for any r as long as we do not *extrapolate*). The reason for (B) is that $|r(r-1)\ \cdots\ (r-n)|$ becomes smallest for that choice.

EXAMPLE 5 **Newton's Forward Difference Formula. Error Estimation**

Compute cosh 0.56 from (14) and the four values in the following table and estimate the error.

j	x_j	$f_j = \cosh x_j$	Δf_j	$\Delta^2 f_j$	$\Delta^3 f_j$
0	0.5	1.127626			
			0.057839		
1	0.6	1.185465		0.011865	
			0.069704		0.000697
2	0.7	1.255169		0.012562	
			0.082266		
3	0.8	1.337435			

Solution. We compute the forward differences as shown in the table. The values we need are circled. In (14) we have $r = (0.56 - 0.50)/0.1 = 0.6$, so that (14) gives

$$\begin{aligned} \cosh 0.56 &\approx 1.127626 + 0.6 \cdot 0.057839 + \frac{0.6(-0.4)}{2} \cdot 0.011865 + \frac{0.6(-0.4)(-1.4)}{6} \cdot 0.000697 \\ &= 1.127626 + 0.034703 - 0.001424 + 0.000039 \\ &= 1.160944. \end{aligned}$$

Error estimate. From (16), since the fourth derivative is $\cosh^{(4)} t = \cosh t$,

$$\epsilon_3(0.56) = \frac{0.1^4}{4!} \cdot 0.6(-0.4)(-1.4)(-2.4) \cosh t$$

$$= A \cosh t,$$

where $A = -0.00000336$ and $0.5 \leqq t \leqq 0.8$. We do not know t, but we get an inequality by taking the largest and smallest $\cosh t$ in that interval:

$$A \cosh 0.8 \leqq \epsilon_3(0.62) \leqq A \cosh 0.5.$$

Since

$$f(x) = p_3(x) + \epsilon_3(x),$$

this gives

Table 19.2 Newton's Divided Difference Interpolation

ALGORITHM INTERPOL ($x_0, \cdots, x_n; f_0, \cdots, f_n; \hat{x}$)

This algorithm computes an approximation $p_n(\hat{x})$ of $f(\hat{x})$ at $\hat{x}$.

INPUT: Data $(x_0, f_0), (x_1, f_1), \cdots, (x_n, f_n)$; $\hat{x}$

OUTPUT: Approximation $p_n(\hat{x})$ of $f(\hat{x})$

Set $f[x_j] = f_j \quad (j = 0, \cdots, n)$.

For $m = 1, \cdots, n - 1)$ do:

For $j = 0, \cdots, n - m$ do:

$$f[x_j, \cdots, x_{j+m}] = \frac{f[x_{j+1}, \cdots, x_{j+m}] - f[x_j, \cdots, x_{j+m-1}]}{x_{j+m} - x_j}$$

End

End

Set $p_0(x) = f_0$.

For $k = 1, \cdots, n$ do:

$$p_k(\hat{x}) = p_{k-1}(\hat{x}) + (\hat{x} - x_0) \cdots (\hat{x} - x_{k-1}) f[x_0, \cdots, x_k]$$

End

OUTPUT $p_n(\hat{x})$

End INTERPOL

this gives

Công thức sai phân Newton lùi

Equal Spacing: Newton's Backward Difference Formula

Instead of forward-sloping differences we may also employ backward-sloping differences. The difference table remains the same as before (same numbers, in the same positions), except for a very harmless change of the running subscript j (which we explain in Example 6, below). Nevertheless, purely for reasons of convenience it is standard to introduce a second name and notation for differences as follows. We define the *first backward difference* of f at x_j by

$$\nabla f_j = f_j - f_{j-1},$$

the *second backward difference* of f at x_j by

$$\nabla^2 f_j = \nabla f_j - \nabla f_{j-1},$$

and, continuing in this way, the ***k*th backward difference** of f at x_j by

(17) $$\nabla^k f_j = \nabla^{k-1} f_j - \nabla^{k-1} f_{j-1}$$

$(k = 1, 2, \cdots)$.

A formula similar to (14) but involving backward differences is **Newton's** (or *Gregory–Newton's*) **backward difference interpolation formula**

(18) $$\begin{aligned} f(x) \approx p_n(x) &= \sum_{s=0}^{n} \binom{r+s-1}{s} \nabla^s f_0 \qquad (x = x_0 + rh,\ r = (x - x_0)/h) \\ &= f_0 + r\nabla f_0 + \frac{r(r+1)}{2!}\nabla^2 f_0 + \cdots + \frac{r(r+1)\cdots(r+n-1)}{n!}\nabla^n f_0. \end{aligned}$$

EXAMPLE 6 Newton's Forward and Backward Interpolations

Compute a 7D-value of the Bessel function $J_0(x)$ for $x = 1.72$ from the four values in the following table, using (a) Newton's forward formula (14), (b) Newton's backward formula (18).

j_{for}	j_{back}	x_j	$J_0(x_j)$	1st Diff.	2nd Diff.	3rd Diff.
0	−3	1.7	0.3979849			
				−0.0579985		
1	−2	1.8	0.3399864		−0.0001693	
				−0.0581678		0.0004093
2	−1	1.9	0.2818186		0.0002400	
				−0.0579278		
3	0	2.0	0.2238908			

Solution. The computation of the differences is the same in both cases. Only their notation differs.

(a) Forward. In (14) we have $r = (1.72 - 1.70)/0.1 = 0.2$, and j goes from 0 to 3 (see first column). In each column we need the first given number, and (14) thus gives

$$
\begin{aligned}
J_0(1.72) &\approx 0.3979849 + 0.2(-0.0579985) + \frac{0.2(-0.8)}{2}(-0.0001693) + \frac{0.2(-0.8)(-1.8)}{6} \cdot 0.0004093 \\
&= 0.3979849 - 0.0115997 + 0.0000135 + 0.0000196 = 0.3864183,
\end{aligned}
$$

which is exact to 6D, the exact 7D-value being 0.3864185.

(b) Backward. For (18) we use j shown in the second column, and in each column the last number. Since $r = (1.72 - 2.00)/0.1 = -2.8$, we thus get from (18)

$$
\begin{aligned}
J_0(1.72) &\approx 0.2238908 - 2.8(-0.0579278) + \frac{-2.8(-1.8)}{2} \cdot 0.0002400 + \frac{-2.8(-1.8)(-0.8)}{6} \cdot 0.0004093 \\
&= 0.2238908 + 0.1621978 + 0.0006048 - 0.0002750 \\
&= 0.3864184.
\end{aligned}
$$

There is a third notation for differences, called the **central difference notation**. It is used in numerics for ODEs and certain interpolation formulas. See Ref. [E5] listed in App. 1.

Bài tập Chương 4 Bài 3. Nội suy hàm số

PROBLEM SET 19.3

1. **Linear interpolation.** Calculate $p_1(x)$ in Example 1 and from it ln 9.3.

2. **Error estimate.** Estimate the error in Prob. 1 by (5).

Bài 3. Hãy tính đa thức Lagrange $p_2(x)$ với các giá trị

3. **Quadratic interpolation. Gamma function.** Calculate the Lagrange polynomial $p_2(x)$ for the values $\Gamma(1.00) = 1.0000$, $\Gamma(1.02) = 0.9888$, $\Gamma(1.04) = 0.9784$ of the gamma function [(24) in App. A3.1] and from it approximations of $\Gamma(1.01)$ and $\Gamma(1.03)$.

4. **Error estimate for quadratic interpolation.** Estimate the error for $p_2(9.2)$ in Example 2 from (5).

Bài 4. Hãy tìm sai số $p_2(9.2)$ trong ví dụ tính ln(9.2)

5. Linear and quadratic interpolation. Find $e^{-0.25}$ and $e^{-0.75}$ by linear interpolation of e^{-x} with $x_0 = 0$, $x_1 = 0.5$ and $x_0 = 0.5$, $x_1 = 1$, respectively. Then find $p_2(x)$ by quadratic interpolation of e^{-x} with $x_0 = 0$, $x_1 = 0.5$, $x_2 = 1$ and from it $e^{-0.25}$ and $e^{-0.75}$. Compare the errors. Use 4S-values of e^{-x}.

Bài 5. Hãy tìm $e^{-0.25}$ và $e^{-0.75}$ bởi nội suy tuyến tính hàm e^{-x} với $x_0=0$, $x_1=0.5$ và $x_0=0.5$, $x_1=1$. Sau đó, tìm $p_2(x)$ của nội suy bậc 2 của hàm e^{-x} với $x_0=0$, $x_1=0.5$ và $x_2=1$ và từ đó tính $e^{-0.25}$ và $e^{-0.75}$. Tính toán sai số. Dùng giá trị của e^{-x} đến 4S.

(Đến 4 chữ số có nghĩa)

x	e^(-x)
0	1.0000
0.5	0.6065
1	0.3679

6. Interpolation and extrapolation. Calculate $p_2(x)$ in Example 2. Compute from it approximations of ln 9.4, ln 10, ln 10.5, ln 11.5, and ln 12. Compute the errors by using exact 5S-values and comment.

7. **Interpolation and extrapolation.** Find the quadratic polynomial that agrees with $\sin x$ at $x = 0, \pi/4, \pi/2$ and use it for the interpolation and extrapolation of $\sin x$ at $x = -\pi/8, \pi/8, 3\pi/8, 5\pi/8$. Compute the errors.

8. **Extrapolation.** Does a sketch of the product of the $(x - x_j)$ in (5) for the data in Example 2 indicate that extrapolation is likely to involve larger errors than interpolation does?

9. **Error function** (35) in App. A3.1. Calculate the Lagrange polynomial $p_2(x)$ for the 5S-values $f(0.25) = 0.27633, f(0.5) = 0.52050, f(1.0) = 0.84270$ and from $p_2(x)$ an approximation of $f(0.75)$ $(= 0.71116)$.

10. **Error bound.** Derive an error bound in Prob. 9 from (5).

11. **Cubic Lagrange interpolation. Bessel function J_0.** Calculate and graph L_0, L_1, L_2, L_3 with $x_0 = 0$, $x_1 = 1, x_2 = 2, x_3 = 3$ on common axes. Find $p_3(x)$ for the data $(0, 1)$, $(1, 0.765198)$, $(2, 0.223891)$, $(3, -0.260052)$ [values of the Bessel function $J_0(x)$]. Find p_3 for $x = 0.5, 1.5, 2.5$ and compare with the 6S-exact values $0.938470, 0.511828, -0.048384$.

12. Newton's forward formula (14). Sine integral. Using (14), find $f(1.25)$ by linear, quadratic, and cubic interpolation of the data (values of (40) in App. A31); 6S-value $\mathrm{Si}(1.25) = 1.14645$) $f(1.0) = 0.94608$, $f(1.5) = 1.32468$, $f(2.0) = 1.60541$, $f(2.5) = 1.77852$, and compute the errors. For the linear interpolation use $f(1.0)$ and $f(1.5)$, for the quadratic $f(1.0)$, $f(1.5)$, $f(2.0)$, etc.

13 Lower degree. Find the degree of the interpolation polynomial for the data $(-4, 50)$, $(-2, 18)$, $(0, 2)$, $(2, 2)$, $(4, 18)$, using a difference table. Find the polynomial.

14. Newton's forward formula (14). Gamma function. Set up (14) for the data in Prob. 3 and compute $\Gamma(1.01)$, $\Gamma(1.03)$, $\Gamma(1.05)$.

15. Divided differences. Obtain p_2 in Example 2 from (10).

16. Divided differences. Error function. Compute $p_2(0.75)$ from the data in Prob. 9 and Newton's divided difference formula (10).

17. Backward difference formula (18). Use $p_2(x)$ in (18) and the values of $\mathrm{erf}\, x$, $x = 0.2, 0.4, 0.6$ in Table A4 of App. 5, compute erf 0.3 and the error. (4S-exact erf $0.3 = 0.3286$).

18. In Example 5 of the text, write down the difference table as needed for (18), then write (18) with general x and then with $x = 0.56$ to verify the answer in Example 5.

19. CAS EXPERIMENT. Adding Terms in Newton Formulas. Write a program for the forward formula (14). Experiment on the increase of accuracy by successively adding terms. As data use values of some function of your choice for which your CAS gives the values needed in determining errors.

20. TEAM PROJECT. Interpolation and Extrapolation. **(a) Lagrange practical error estimate** (after Theorem 1). Apply this to $p_1(9.2)$ and $p_2(9.2)$ for the data $x_0 = 9.0$, $x_1 = 9.5$, $x_2 = 11.0$, $f_0 = \ln x_0$, $f_1 = \ln x_1$, $f_2 = \ln x_2$ (6S-values).

(b) Extrapolation. Given $(x_j, f(x_j)) = (0.2, 0.9980)$, $(0.4, 0.9686)$, $(0.6, 0.8443)$, $(0.8, 0.5358)$, $(1.0, 0)$. Find $f(0.7)$ from the quadratic interpolation polynomials based on (α) 0.6, 0.8, 1.0, (β) 0.4, 0.6, 0.8, (γ) 0.2, 0.4, 0.6. Compare the errors and comment. [Exact $f(x) = \cos\left(\frac{1}{2}\pi x^2\right)$, $f(0.7) = 0.7181$ (4S).]

(c) Graph the product of factors $(x - x_j)$ in the error formula (5) for $n = 2, \cdots, 10$ separately. What do these graphs show regarding accuracy of interpolation and extrapolation?

and extrapolation?

21. WRITING PROJECT. Comparison of interpolation methods. List 4–5 ideas that you feel are most important in this section. Arrange them in best logical order. Discuss them in a 2–3 page report.