Interpolation

Let the explicit form of $y = f(x)$ be unknown in an analytical form but it is known only at the $(n+1)$ distinct values of x and $(n+1)$ data points as is the Table. Since the analytical formula for $f(x)$ is not known, the precise value can not be obtained for specific values of x. Here we are to evaluate the value of y, for any intermediate value of given $x \in (x_0, x_n)$, called *interpolation*. If x is outside of (x_0, x_n), the process is called *extrapolation*. The object is to estimate the values of the function at non-tabular points having an error bound between the estimated and the true values. The study of interpolation is based on the calculus of finite differences. The methods for interpolation may be put into two categories according to whether the tabular points are equispaced or not necessarily at equal interval. In any case, it will be assumed that the behavior of y with repeat is smooth, i.e., there are no sudden variation in the value of y.

1 Interpolating Polynomial

Let $f(x) \in \mathcal{C}^{\infty}(-\infty, \infty)$. The principle of interpolating polynomial is " the selection of a function $\phi(x)$ from a given class of functions such that the graph of $y = \phi(x)$ passes through a finite set of given points". The function $\phi(x)$ is called interpolating function or approximating function $\phi(x)$ which satisfy the *interpolating conditions* $\phi(x_i) = f(x_i); i = 0, 1, \cdots, n$. Although there may be several functions interpolating the same data, we shall be confined to polynomial approximation in one form or another. When $\phi(x)$ is a polynomial, the process of representing $f(x)$ by $\phi(x)$ is called *polynomial or parabolic interpolation* , and when $\phi(x)$ is a finite trigonometric series, the process is *trigonometric interpolation*. In like manner, $\phi(x)$ may be a series of exponential functions, Legendre polynomials etc. Now, polynomial interpolation is based on the following theorem known as *Weierstrass theorem* :

THEOREM **1.** *Let a function $f(x) \in C[a,b]$ and let $\epsilon > 0$ be any preassigned small number. Then, $\exists$ a polynomial $\phi(x)$ for which $|f(x) - \phi(x)| < \epsilon;\ \ x \in [a,b]$ i.e., any continuous function can be uniformly approximated by a polynomial of sufficiently high degree within any prescribed tolerance on the finite interval.*

(i) Generally $\phi(x)$ is assumed to be *smooth or continuous* in the interval of interpolation and that it is amenable to approximation by some type of function - polynomial, trigonometric,etc.

(ii) Polynomials are the simplest, easy to derive and most widely used class of functions.

(iii) We use interpolating polynomials to determine the formulas for numerical differentiation, integration, and numerical solution of ordinary and partial differential equations, etc.

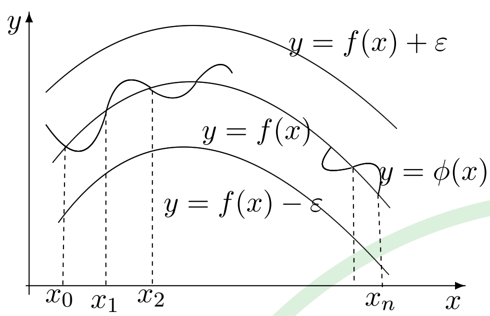

Figure 1: Interpolation

Thus the graph (Fig. 1) of the interpolating polynomial $\phi(x)$ is confined within the region bounded by $y = f(x) - \varepsilon$ and $y = f(x) + \varepsilon$ for all $x \in (a, b)$. This theorem does not guarantee the existence of an interpolating polynomial, and in fact, it is not obvious that such a polynomial will exist for any arbitrary table of values. Polynomial approximation has wide important uses because

THEOREM **2.** *Given any real valued function $f(x)$ and $n+1$ distinct points $x_0, x_1, \cdots, x_n$ there exists unique polynomial of maximum degree n which interpolates $f(x)$ at the points $x_0, x_1, \cdots, x_n$.*

Proof: Any polynomial $f(x)$ which has $(n+1)$ distinct zeros $x_0, x_1, \cdots, x_n$ can be factorized as

$$f(x) = (x - x_0)(x - x_1) \cdots (x - x_n) q(x)$$

where $q(x)$ is a polynomial such that either degree of $q(x)$ is 0 or degree of $q(x) =$ degree of $f(x) - (n+1)$. Let $\phi(x)$ and $\psi(x)$ be two polynomials of maximum degree n and both interpolates $f(x)$ at the $(n+1)$ distinct points $(x_0, y_0), \cdots, (x_n, y_n)$. Define $r(x) = \phi(x) - \psi(x)$ then $r(x)$ is a polynomial of maximum degree n and

$$r(x_i) = \phi(x_i) - \psi(x_i) = y_i - y_i = 0; i = 0, 1, \cdots, n.$$

This shows that $r(x)$ has $(n+1)$ distinct zeros. But $r(x)$ is of maximum degree n, so that $r(x)$ can have only n zeros. By fundamental theorem of algebra "a polynomial of degree n has at most n roots unless it is identically $zero$", i.e.,

$$r(x) = 0 \Rightarrow \phi(x) = \psi(x)$$

and the polynomial is *unique.* □

Since the polynomial of degree n has $n+1$ coefficients, we can calculate these coefficients in such a way that the polynomial fitted the given function at $n+1$ distinct points. Hence this theorem assures the existence and method of construction of unique polynomial of the prescribed degree. Practically we derive various interpolation formulae and uniqueness says they are actually different forms of the same polynomial. Thus the problem of polynomial interpolation is now completely solved.

THEOREM **3.** *[Error in approximating a function by a polynomial] Let a real valued function $f(x) \in \mathcal{C}^{n+1}[a, b]$ be prescribed at the $(n+1)$ known distinct interpolating points $x_0, x_1, \cdots, x_n$. Then the error in approximating $f(x)$ by the interpolating polynomial $\phi(x)$ of maximum degree n which passes through the point (x_k, y_k) satisfying $y_k = f(x_k) : k = 0(1)n$ is given by*

$$E(x) = f(x) - \phi(x) = \frac{\pi(x) f^{n+1}(\xi)}{(n+1)!} \tag{1}$$

where $\min_i \{x_i, x\} < \xi < \max_i \{x_i, x\}$ and $\pi(x) = (x - x_0) \cdots (x - x_n)$.

Proof: When $f(x)$ is approximated by $\phi(x)$, truncation error or error of approximation occurs due to using a finite degree polynomial. If x is the node point then it is trivial. When x is not a node point, define an auxiliary function by

$$F(z) = f(z) - \phi(z) - k\pi(z); \;\; k = \frac{f(x) - \phi(x)}{\pi(x)}$$

where $f(x)$ is the function from which the data points are sampled $y_i = f(x_i)$. Now $F(z) \in \mathcal{C}^{n+1}[a, b]$ as $f(z)$ and $\phi(z)$ are so. Also

$$F(x_i) = f(x_i) - \phi(x_i) - k\pi(x_i) = 0; \;\; i = 0, 1, \cdots, n$$

and for an arbitrary chosen point $z = x(\neq x_i)$ we have

$$F(x) = f(x) - \phi(x) - k\pi(x) = 0.$$

Hence the function $F(z)$ has $(n + 2)$ distinct roots in the interval $I = [x_0, x_n]$. By Roll's theorem $F'(z)$ must have $(n + 1)$ distinct roots in I, and in general $F^j(z)$ have $n + 2 - j$ zeros in $I; j = 0, 1, \cdots, n + 1$. Let ξ be one such root. Thus

$$F^{n+1}(\xi) = 0 \;\; \Rightarrow \;\; f^{n+1}(\xi) - k(n + 1)! = 0 \Rightarrow k = \frac{f^{n+1}(\xi)}{(n + 1)!} \;\; a < \xi < b.$$

which gives the truncation error formula (1). □

RESULT **1.** It must be realized that $E(x)$ gives an upper bound for the error while the actual error may be smaller. Eq. (1) has two components:

(i) First component, depends only on the choice of the tabular points $x_0, x_1, \cdots, x_n$, but independent of the interpolation function. If they are located too far apart, then the value of the factor $\pi(x)$ will be too large. Moreover, the oscillation of the approximating polynomial may attain too big amplitudes which may result in totally absurd result.

(ii) The second component is the value of the derivative $f^{(n+1)}(\xi)$ in the entire interval $[x_0, x_n]$ of interest. Its value should be remain small in the given interval otherwise, the magnitudes of the error $R(x)$ may be too large.

second part $f^{(n+1)}(\xi)$ which depends on the function being interpolated, but is essentially independent of the choice of interpolation points.

Since we seldom know $f^{(n+1)}(\xi)$, the truncation error formula is of limited use. However, it does give the order of the error if $|f^{(n+1)}(\xi)|$ is bounded. If f is known analytically, then an upper bound of $|f^{(n+1)}(x)|$ can be found over an entire interval $[x_0, x_n]$. Therefore, we can find an upper bound of the truncation error using Eq. (1).

RESULT **2.** Let the distinct arguments are equally spaced i.e., $x_i = x_0 + ih; i = 0, 1, \ldots, n; h > 0$. Let $x = x_0 + uh$ we have,

$$x - x_0 = uh; x - x_1 = x - x_0 + x_0 - x_1 = uh - h = (u - 1)h; \cdots;$$
$$x - x_n = x - x_0 + x_0 - x_n = uh - nh = (u - n)h$$

Therefore, using Eq.(1), we get the truncation error

$$E(x) = \frac{uh \cdot (u - 1)h \cdots (u - n)h}{(n + 1)!} f^{n+1}(\xi) = \frac{u(u - 1) \ldots (u - n)}{(n + 1)!} h^{n+1} f^{n+1}(\xi). \tag{2}$$

RESULT **3.** Since $f(x)$ is generally unknown, the above formula is almost useless in practical computation. Now, by Lagrange mean value theorem, we have

$$\triangle f(x) = f(x+h) - f(x) = hf'(\xi),$$

h is very small and so, if $f^{n+1}(x)$ does not vary too rapidly in $[x_0, x_n]$,

$$h^{n+1} f^{n+1}(\xi) \approx \triangle^{n+1} f(x_0) = \triangle^{n+1} y_0.$$

Thus, the suitable form for computation of error is:

$$E(x) \simeq \frac{u(u-1)\cdots(u-n)}{(n+1)!} \triangle^{n+1} y_0. \tag{3}$$

which is used in most practical purposes.

EXAMPLE **1.** *If linear interpolation is used to interpolate the function* $f(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2/2} dt$, *show that the error in such interpolation, using the data* (x_0, f_0) *and* (x_1, f_1) *can not exceed* $\frac{(x_1 - x_0)^2}{4\sqrt{\pi e}}$.

Solution: Using Eq.(2), the truncation error in the linear interpolation is given by

$$|E(x)| \leq \frac{1}{2!} \max \Big|(x - x_0)(x - x_1)\, f''(\xi)\Big|; \qquad \xi \in (x_0, x_1)$$

Let $g(x) = (x - x_0)(x - x_1)$. Setting $g'(x) = 0$, we obtain the critical point of $g(x)$ as $x = \frac{1}{2}(x_0 + x_1)$. Hence, the maximum value of $|g(x)|$ occurs at $x = \frac{1}{2}(x_0 + x_1)$ and $|g(x)|_{\max} = \frac{1}{4}(x_0 - x_1)^2$. Also,

$$f'(x) = \frac{2}{\sqrt{\pi}} e^{-x^2/2}; f''(x) = -\frac{2x}{\sqrt{\pi}} e^{-x^2/2}; f'''(x) = \frac{2}{\sqrt{\pi}}(x^2 - 1)e^{-x^2/2}.$$

For optimal value of $f''(x)$, we have $f'''(x) = 0$ gives $x = \pm 1$ and the maximum value of $f''(x)$ occurs at $a = -1$. Therefore

$$\max |f''(x)| = \frac{2}{\sqrt{\pi}} e^{-1/2} = \frac{2}{\sqrt{e\pi}},$$

$$\therefore \quad |E| \leq \frac{1}{2!} \cdot \frac{1}{4}(x_0 - x_1)^2 \cdot \frac{2}{\sqrt{e\pi}} = \frac{(x_1 - x_0)^2}{4\sqrt{\pi e}}.$$

EXAMPLE **2.** *Determine the space h in a table of equally spaced values of the function* $y = \log_e x$ *between* 1 *and* 2 *so that the interpolation with a linear interpolation in this table will yield a desired accuracy of order* 10^{-6}.

Solution: The truncation error in the linear interpolation is given in Eq.(3) as

$$|E(x)| = \frac{|u(u-1)|h^2}{2!} |f''(\xi)| \leq \frac{M.h^2}{8}.$$

Since $f(x) = \log_e x$, so $f''(x) = -\frac{1}{x^2}$ and so, $|f''(x)| \leq 1$ for $x \in [1, 2]$. Forth the desired accuracy of order 10^{-6}, we must have,

$$\frac{h^2}{8} < 0.5 \times 10^{-7} \Rightarrow h < 6.32456 \times 10^{-4}.$$

EXAMPLE **3.** *Determine the step size h in a table of equally spaced values of the function $f(x) = \sqrt{x}$ in $[1,2]$, so that interpolation with a second degree polynomial in this table is less than 5×10^{-8}.*

Solution: By assumption,the table will contain y_i with $x_i = 1 + ih, i = 0, 1, \cdots, n$ where $h = \frac{2-1}{n}$. As in Eq.(2), the truncation error in quadratic polynomial which interpolates $f(x)$ at x_{i-1}, x_i, x_{i+1} is

$$\begin{aligned} E(x) &= (x - x_{i-1})(x - x_i)(x - x_{i+1})\frac{f'''(\xi)}{3!};\ \ \xi \in (x_{i-1}, x_{i+1}) \\ \therefore |E(x)| &\leq \frac{M_3}{6} \max_{x\in[x_{i-1},x_{i+1}]} \Big|(x - x_{i-1})(x - x_i)(x - x_{i+1})\Big|; M_3 = \max|f'''(\xi)| \\ &\leq \frac{M_3}{6} \max_{u\in[-h,h]} \Big|(u + h)u(u - h)\Big| = \frac{M_3}{6} \max_{u\in[-h,h]} \Big|u(u^2 - h^2)\Big| \end{aligned}$$

Let $g(u) = u(u^2-h^2)$. Setting $g'(u) = 0$, we obtain the critical point of $g(u)$ as $u = \pm\frac{h}{\sqrt{3}}$. Hence, the maximum value of $|g(u)|$ occurs at $u = \frac{h}{\sqrt{3}}$ and $|g(u)|_{\max} = \dfrac{h^3}{3\sqrt{3}}$. Also as $f(x) = \sqrt{x}$ so $f'''(x) = \frac{3}{8}x^{-\frac{5}{2}};\ \max_{1\leq x\leq 2} |f'''(x)| \leq \frac{3}{8}$. Thus

$$|E(x)| \leq \frac{M_3}{6} \max_{u\in[-h,h]} \Big|u(u^2 - h^2)\Big| = \frac{3/8}{6} \cdot \frac{h^3}{3\sqrt{3}} = \frac{h^3}{24\sqrt{3}}.$$

Given $\epsilon = 5 \times 10^{-8}$ as the accuracy, So the step size h is given by

$$\frac{h^3}{24\sqrt{3}} < \epsilon = 5 \times 10^{-8} \Rightarrow h < 0.012762.$$

2 Newton-Gregory's Form of the Interpolating Polynomial

Let $y = f(x)$ be not given explicitly but takes the values $y_0, \cdots, y_n$ corresponding to the $(n + 1)$ usually increasing sequence of equally spaced arguments as $x_0 + ih,\ \ i = 0, 1, \cdots, n$. We want to determine the value of y at some point $x \in (x_0, x_n) - \{x_0, x_1, \cdots, x_n\}$. Here we to discuss actually two table oriented interpolation formulae.

2.1 Newton's Forward Difference Formula

Newton's forward interpolation formula is a simple polynomial from which we are to calculate the value of y for a given non tabulated value of x, which lies near the beginning of the tabular values.

Let the polynomial $\phi(x)$ of degree n be written in the form

$$\phi(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_n(x - x_0)\cdots(x - x_{n-1})$$

where the coefficients a_i's are constants, to be determined so as to agree the interpolating conditions $y_i = f(x_i) = \phi(x_i); i = 0, 1, \cdots, n$. Substituting in the successive values we get

$$\begin{aligned} y_0 &= \phi_n(x_0) = a_0 \Rightarrow a_0 = y_0. \\ y_1 &= \phi_n(x_1) = a_0 + a_1(x_1 - x_0) \Rightarrow a_1 = \frac{y_1 - y_0}{h} = \frac{\Delta y_0}{h} \\ y_2 &= \phi_n(x_2) = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) \\ \Rightarrow a_2 2! h^2 &= y_2 - y_0 - 2\Delta y_0 = (E-1)^2 y_0 = \Delta^2 y_0 \Rightarrow a_2 = \frac{\Delta^2 y_0}{2! h^2} \end{aligned}$$

and similarly $a_n = \frac{\Delta^n y_0}{n! h^n}$. Using these values $\phi(x)$ can be written as

$$\begin{aligned} \phi(x) &= y_0 + \frac{x - x_0}{h}\Delta y_0 + \frac{(x-x_0)(x-x_1)}{2! h^2}\Delta^2 y_0 + \cdots + \frac{(x-x_0)\cdots(x-x_{n-1})}{n! h^n}\Delta^n y_0 \\ &= \sum_{i=0}^{n} \frac{1}{i! h^i} \triangle^i y_0 \prod_{j=0}^{i-1} (x - x_i) \end{aligned}$$

which is the *Newton's (Newton-Gregory) forward difference formula* for the interpolating polynomial and it is useful to interpolate near the beginning of a set of tabular values.

For practical purpose, let us linearly change the origin and scale as $u = \frac{x - x_0}{h} \Rightarrow x - x_0 = uh$, where the dimensionless variable $u(-1 \le u \le 1)$ is called a *phase*. Then

$$\begin{aligned} x - x_1 &= (x - x_0) - (x_1 - x_0) = uh - h = (u-1)h \\ &\vdots \\ x - x_n &= (x - x_0) - (x_n - x_0) = uh - nh = (u-n)h. \end{aligned}$$

Hence the interpolation formula can be written in more convenient form as

$$\begin{aligned} \phi(x) &= y_0 + u\Delta y_0 + \frac{u(u-1)}{2!}\triangle^2 y_0 + \ldots + \frac{u(u-1)\ldots(u-n+1)}{n! h^n}\Delta^n y_0 \\ &= \sum_{i=0}^{n} \triangle^i y_0 \prod_{j=0}^{i-1} \frac{u-j}{j+1} = \sum_{i=0}^{n} \binom{u}{i} \triangle^i y_0 \end{aligned} \quad (4)$$

where the coefficient of $\triangle$'s are binomial coefficients.

RESULT **4.** In terms of factorial notation Eq. (4) can be written as

$$\phi(x) = y_0 + \frac{u^{(1)}}{1!}\triangle y_0 + \frac{u^{(2)}}{2!}\triangle^2 y_0 + \cdots + \frac{u^{(n)}}{n!}\triangle^n y_0$$

(i) The point x_0 is known as *starting point* of the formula (4). The starting point may be any tabular values but then the formula will contain only those values which come after the value chosen as starting point. $1, u, \binom{u}{2}, \cdots$, are independent of the problems, called the *coefficients of the formula*.

(ii) It is called the 'forward' interpolation formula because it contains values of the tabulated function from y_0 onward to the right(forward from y_0) and none to the left of the value. This formula can not be used for interpolating when the values of arguments are not equally spaced and when the non tabulated given value of x lies in the bottom of the table.

Error estimate : To find the error committed in approximating $f(x)$ by the polynomial $\phi(x)$, we have from Equation (1), the remainder or truncation error or simply error is

$$R_{n+1}(x) = \frac{\pi(x) f^{n+1}(\xi)}{(n+1)!} = (x-x_0)(x-x_1)\cdots(x-x_n)\frac{f^{n+1}(\xi)}{(n+1)!}$$

where $\min\{x, x_0, x_1, \cdots, xn\} < \xi < \max\{x, x_0, x_1, \cdots, xn\}$. According to Equation (**??**), we have $f^{(n+1)}(\xi) \approx \frac{\Delta^{n+1}(x_0)}{h^{n+1}}$ and the above equation can be written as

$$R_{n+1} \simeq \frac{u(u-1)\cdots(u-n)}{(n+1)!}\,\Delta^{n+1}y_0 = A_n(u)\triangle^{n+1}y_0 \tag{5}$$

where, $A_n(u) = \frac{u(u-1)\cdots(u-n)}{(n+1)!}$. If $u \in (0,1)$, then $\max|u(u-1)| = \dfrac{1}{4}$ and $\max|(u-2)(u-3)\cdots(u-n)| = n!$. Also, in the last significant figure $\triangle^{n+1}y_0 \le 9$ so that $|R_{n+1}| \le \dfrac{9}{4(n+1)} < 1$ for $n > 2$ and $0 < u < 1$. Hence the maximum truncation error in forward formula is numerically less than 1 in the last significant figure. Again for $0 < u < 1$ we have for extrapolation $|A_n(-u)| = \left|\dfrac{u(u+1)\cdots(u+n)}{(n+1)!}\right|$. Also $|A_n(u)| > |A_n(-u)|$, shows that the use of interpolation in computing a function value is always preferable than the use of extrapolation.

EXAMPLE **4.** *Given a table of values of $\frac{1}{x}$, find $f(2.72)$ using quadratic interpolation.*

x	2.7	2.8	2.9
$f(x)$	0.3704	0.3571	0.3448

Find also the estimate of the error.

Solution: The forward difference table is given below. Here $u = \dfrac{x-2.70}{0.1} = 10(x-2.7)$.

x	y	$\triangle y$	$\triangle^2 y$
2.7	0.3704		
		−0.0133	
2.8	0.3571		0.0010
		−0.0123	
2.9	0.3448		

Using formula (4), we get

$$\begin{aligned} f(x) &= y_0 + u\triangle y_0 + \frac{u(u-1)}{2!}\triangle^2 y_0 + \cdots \\ &= 0.3704 + 10(x-2.7)\times(-0.0133) \\ &\quad + \frac{1}{2!}\times 100(x-2.7)(x-3.7)\times 0.0010 \\ &= 0.05x^2 - 0.453x + 0.8586. \end{aligned}$$

When $x = 2.72$ then $u = \frac{2.72-2.70}{0.1} = 0.2$ and so,

$$f(2.72) \;=\; 0.3704 + 0.2\times(-0.0133) - \frac{0.2\times 0.8}{2}\times 0.0010 \approx 0.3677.$$

Since $f(x) = \frac{1}{x}$, so $f'''(x) = -\frac{6}{x^4}$ and so the estimation of error is given by,

$$\begin{aligned} |E(x)| &\le \frac{h^3}{3!}\max\Big|u(u-1)(u-2)\Big|\max\Big|f'''(\xi)\Big| \\ &\le \frac{(0.1)^3}{6}\times(0.2\times 0.8\times 1.8)\times\frac{6}{(2.72)^4} \approx 0.5\times 10^{-5}. \end{aligned}$$

Listing 1: Program for Newton Forward Interpolation Formula

```
% Program for Newton's Forward Interpolation Formula
n=input('Enter the number of subintervals n:  ');
x=input('Enter the values of x:   ');
y=input('Enter the values of y:   ');
xp=input('Enter the interpolating point xp:   ');
for i=1:n+1
    fprintf('\n %d   %d',x(i),y(i));
end
h=x(2)-x(1);
u=(xp-x(1))/h;
for j=1:n+1
    dy(j)=y(j);
end
prod=1;
sum=y(1);
for i=1:n+1
    for j=1:(n+1)-i
        dy(j)=dy(j+1)-dy(j);
    end
        prod=prod*(u-i+1)/i;
        sum=sum+prod*dy(1);
end
   fprintf('\n the value of y at x=% 14.5f is % 13.11f\n',xp,sum
     );
```

2.2 Newton's Backward Difference Formula

Newton's backward interpolation formula is an simple expression from which we are to calculate the value of the entry y for a given non tabulated value of x, which lies near the end of the tabular values. Consider a polynomial $\phi(x)$ of maximum degree n as

$$\phi(x) = a_0 + a_1(x - x_n) + a_2(x - x_n)(x - x_{n-1}) + \ldots + a_n(x - x_n)\ldots(x - x_1)$$

which coincides with the $(n + 1)$ points $(x_0, y_0), \ldots, (x_n, y_n)$ and a_i's are constants to be determined by the interpolating conditions $y_i = f(x_i) = \phi_n(x_i); i = 0(1)n$. Substituting the successive values we have,

$$\begin{aligned}
y_n &= \phi(x_n) = a_0 \Rightarrow a_0 = y_n \\
y_{n-1} &= \phi(x_{n-1}) = a_0 + a_1(x_{n-1} - x_n) \Rightarrow a_1 = \frac{y_n - y_{n-1}}{h} = \frac{\nabla y_n}{h} \\
y_{n-2} &= \phi(x_{n-2}) = a_0 + a_1(x_{n-2} - x_n) + a_2(x_{n-2} - x_n)(x_{n-2} - x_{n-1}) \\
&= y_n + 2\nabla y_n + a_2 2!h^2 \Rightarrow a_2 2!h^2 = y_n - 2y_{n-1} + y_{n-2} \Rightarrow a_2 = \frac{\nabla^2 y_n}{2!h^2}
\end{aligned}$$

and similarly $a_n = \dfrac{\nabla^n y_n}{n!h^n}$. Therefore

$$\phi(x) = y_n + \frac{x-x_n}{h}\nabla y_n + \frac{(x-x_n)(x-x_{n-1})}{2!}\nabla^2 y_n + \ldots + \frac{(x-x_n)\ldots(x-x_1)}{n!}\nabla^n y_0. \quad (6)$$

which is *Gregory-Newton's backward difference interpolation formula,* and it is useful to interpolate near the end of a set of tabular values. Now, setting $v = \frac{x-x_n}{h}$, we can obtain

$$\begin{aligned} x - x_{n-1} &= x - (x_n - h) = (x - x_n) + h = (v+1)h \\ &\vdots \\ x - x_1 &= x - \{x_n - (n-1)h\} = (x - x_n) + (n-1)h = (v+n-1)h \end{aligned}$$

Hence the interpolation formula can be written as

$$\begin{aligned} \phi(x) &= y_n + v\nabla y_n + \frac{v(v+1)}{2!}\nabla^2 y_n + \ldots + \frac{v(v+1)\ldots(v+n-1)}{n!}\nabla^n y_n \\ &= \sum_{i=0}^{n}\binom{v+i-1}{i}\nabla^i y_n = \sum_{i=0}^{n}(-1)^i\binom{-v}{i}\triangle^i y_{n-i}. \end{aligned} \quad (7)$$

In the case of practical numerical computation, instead of Equation (6), we should use Equation (7) in order to ease the calculation involved with it.

(i) It is called *'backward' interpolation formula* because the formula contains values of the tabulated function from y_n backward to the left and none to the right of the value y_n. This formula is look symmetrical with respect to the starting point y_n.

(ii) The Newton's backward interpolation formula is especially useful in extending a tabulation, and for generating other formulas useful for advancing solution of differential equations. In fact the forward and backward polynomials ending at the same difference entry are identical.

Error estimate : To find the error committed in approximating $f(x)$ by the polynomial $\phi(x)$, we have from Equation (1), the remainder or truncation error or simply error is

$$R_{n+1}(x) = \frac{\pi(x)f^{n+1}(\xi)}{(n+1)!} = (x-x_0)(x-x_1)\cdots(x-x_n)\frac{f^{n+1}(\xi)}{(n+1)!}$$

where $\min\{x, x_0, x_1, \cdots, xn\} < \xi < \max\{x, x_0, x_1, \cdots, xn\}$. According to Equation (**??**), we have $f^{(n+1)}(\xi) \approx \frac{\nabla^{n+1} y_n}{h^{n+1}}$ and the above equation can be written as

$$R_{n+1} \simeq \frac{v(v+1)\cdots(v+n)}{(n+1)!}\nabla^{n+1}y_n = B_n(v)\nabla^{n+1}y_n \quad (8)$$

where, $B_n(v) = \frac{v(v+1)\cdots(v+n)}{(n+1)!}$. Since $|A_n(-u)| = |B_n(v)|$, the proper interpolation is always preferable than extrapolation.

EXAMPLE **5.** *The population(in lakhs) of a certain city according to census is given below:*

Year	1941	1971	1981	1991	2001
Population	46	66	81	93	101

Extrapolate the population for the year 2007.

Solution: We take 1961 as year zero as shown in table below.
Difference table for Example 5

	x	y	$\triangle y$	$\triangle^2 y$	$\triangle^3 y$	$\triangle^4 y$
1961	0	46				
			20			
1971	1	66		-5		
			15		2	
1981	2	81		-3		-3
			12		-1	
1991	3	93		-4		
			8			
2001	4	101				

Here $h = 10, x_n = 2001, v = \frac{2007-2001}{10} = 0.6$. Since the value in last column is greater than the lower order difference, the last difference may be negative.

$$\phi(2007) = 101 + (0.6)(8) + \frac{(0.6)(1.6)}{2!}(-4)$$
$$+\frac{0.6(0.6)(2.6)}{3!(-1)} = 105.8 - 1.92 - 0.416$$
$$= 103.46 \text{ lakhs}$$

EXAMPLE **6.** *In a class of* 100*, the students are placed into the following categories according to the marks they have obtained in a test out of* 60*.*

Marks obtained (x)	$0-9$	$10-19$	$20-29$	$30-39$	$40-49$	$50-59$
Number of Students (y)	3	12	15	35	25	10

Find by using Newton's backward formula, the number of students who have secured 75% *and above marks.*

Solution: We have to find the number of students who have secured 75% of 60 i.e. $\frac{75\times 60}{100} = 45$ marks and above. We construct the difference table as follows. Here, $h = 10, x_n = 50$ so, $v = \frac{45-50}{10} = -0.5$. Here, we should leave out 3[rd] and 4[rd] differences as they start increasing

x	y	$\triangle y$	$\triangle^2 y$	$\triangle^3 y$	$\triangle^4 y$
Less 10	3				
		12			
Less 20	15		3		
		15		17	
Less 30	30		20		-47
		35		-30	
Less 40	65		-10		25
		25		-5	
Less 50	90		-15		
		10			
Less 60	100				

$$\phi(x) = y_n + v\nabla y_n + \frac{v(v+1)}{\nabla^2 y_n}$$
$$= 77.5 + 1.25 = 78.75$$

Thus 79 students got less than 45 marks. Therefore, $(100 - 79) = 21$ students have got 75% marks and above.

Listing 2: Program for Newton Backward Interpolation Formula

```
% Program for Newton's Backward Interpolation Formula
n=input('Enter the number of subintervals n  ');
x=input('Enter the values of x  ');
y=input('Enter the values of y ');
xp=input('Enter the interpolating point xp  ');
 for i=1:n+1
      fprintf('\n %d   %d',x(i),y(i));
 end
h=x(2)-x(1);
v=(xp-x(n+1))/h;
 for j=1:n+1
```

```
    dy(j)=y(j);
end
prod=1;
sum=y(n+1);
for i=1:n
    for j=1:(n+1)-i
        dy(j)=dy(j+1)-dy(j);
    end
      prod=prod*(v+i-1)/i;
      sum=sum+(prod*dy(n+1-i));
end
   fprintf('\n the value of y at x=% 14.5f is % 13.11f\n',xp,sum
      );
```

3 Lagrange's Interpolation Formula

The problem of determining a polynomial $\phi(x)$ of maximum degree n that agrees with the $(n+1)$

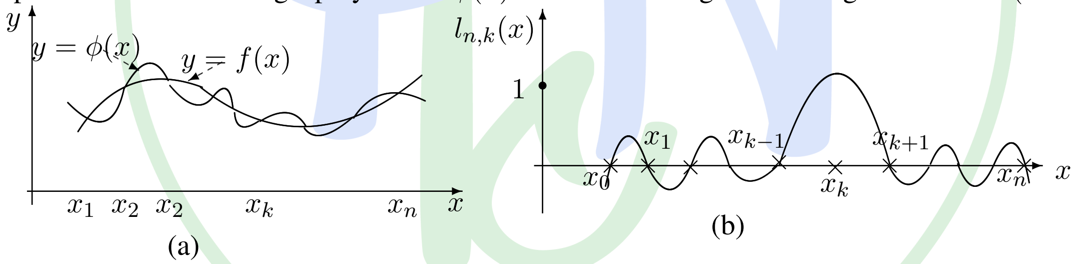

Figure 2: Lagrange interpolating polynomial

distinct points $(x_0, y_0), (x_1, y_1), \cdots, (x_n, y_n)$ as approximating a function f for which $f(x_k) = y_k$ by means of a n^{th} degree polynomial interpolation depicted in the Fig.(2 (a)). Let $\phi(x)$ be of the form

$$\begin{aligned}\phi(x) &= a_0(x-x_1)(x-x_2)\cdots(x-x_n) + a_1(x-x_0)(x-x_2)\cdots(x-x_n)\\ &+ a_2(x-x_0)(x-x_1)\cdots(x-x_n) + \cdots + a_n(x-x_0)(x-x_1)\cdots(x-x_{n-1})\end{aligned}$$

for appropriate constants $a_0, a_1, \cdots, a_n$. To determine the constants, we use the *interpolating condition* $y_i = f(x_i) = \phi(x_i); i = 0(1)n$. Thus

$$\begin{aligned}y_0 &= \phi(x_0) = a_0(x_0-x_1)(x_0-x_2)\cdots(x_0-x_n) \Rightarrow a_0 = \frac{y_0}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_n)}.\\ y_1 &= \phi(x_1) = a_1(x_1-x_0)(x_1-x_2)\ldots(x_1-x_n) \Rightarrow a_1 = \frac{y_1}{(x_1-x_0)(x_1-x_2)\cdots(x_1-x_n)},\end{aligned}$$

and so on. Lastly $a_n = \dfrac{y_n}{(x_n - x_0)(x_n - x_1)\ldots(x_n - x_{n-1})}$. Thus the Lagrange interpolation formula can be written as

$$\begin{aligned}\phi(x) &= \frac{(x-x_1)(x-x_2)\cdots(x-x_n)}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_n)}y_0 + \frac{(x-x_0)(x-x_2)\cdots(x-x_n)}{(x_1-x_0)(x_1-x_2)\cdots(x_1-x_n)}y_1 + \cdots \\ &\quad + \frac{(x-x_0)(x-x_1)\cdots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)\cdots(x_n-x_{n-1})}y_n \\ &= \sum_{i=0}^{n}\left[\prod_{j=0,j\neq i}^{n}\frac{x-x_j}{x_i-x_j}\right]y_i = \sum_{i=0}^{n} l_i(x)y_i \end{aligned} \tag{9}$$

where, we define the functions $l_i(x)$ as

$$l_i(x) = \begin{cases} \displaystyle\prod_{j=0,i\neq j}^{n}\frac{x-x_j}{x_i-x_j}; & i = 0(1)n \\ \displaystyle\frac{\pi(x)}{(x-x_i)\pi'(x_i)}; & \pi(x) = \prod_{i=0}^{n}(x-x_i) \end{cases} \tag{10}$$

The interpolating polynomial given by Eq. (9) with $l_i(x)$ defined by Eq. (10) is called the *Lagrange fundamental polynomial.* Now,

$$\begin{aligned} l_0(x_0) &= \frac{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_n)}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_n)} = 1 \\ l_0(x_1) &= \frac{(x_1-x_1)(x_1-x_2)\cdots(x_1-x_n)}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_n)} = 0 \\ &\vdots \\ l_0(x_n) &= 0 \end{aligned}$$

$l_i(x)$ is zero at every tabular argument except x_k, and 1 at x_k; in other words

$$l_i(x_k) = \delta_{ik} = \begin{cases} 1; & \text{if } i = k \\ 0; & \text{if } i \neq k \end{cases}$$

as depicted in the Fig.(2 (b)). The Lagrange interpolating polynomial (9) can also be written as

$$\begin{vmatrix} \phi(x) & 1 & x & x^2 & \ldots & x^n \\ y_0 & 1 & x_0 & x_0^2 & \ldots & x_0^n \\ \vdots & \ddots & \vdots & & & \\ y_n & 1 & x_n & x_n^2 & \ldots & x_n^n \end{vmatrix} = 0. \tag{11}$$

Deduction 3.1. Let $x = x_0 + uh$ and $x_i = x_0 + ih$, the Lagrangian functions are given by

$$\begin{aligned} l_i(x) &= \frac{u(u-1)\cdots(u-i+1)(u-i-1)\cdots(u-n)}{i(i-1)\cdots 1\cdot(-1)(-2)\cdots(-n-i)} \\ &= (-1)^n\frac{u(u-1)\cdots(u-n)}{n!}\frac{(-1)^i}{u-i}\binom{n}{i}; i = 0(1)n. \end{aligned}$$

These functions depend only on n and not on the particular table and may be tabulated for different values of n. Thus the formula (9) for equally spaced points is

$$\phi(x) = \sum_{i=0}^{n} \frac{(-1)^{n+i}}{i!(n-i)!} \frac{u(u-1)\cdots(u-n)}{u-i} y_i.$$

Thus the method is applicable to both equispaced and unequispaced arguments. It is used to derive Newton-Cote's formula in numerical integration.

Error : There are two kinds of errors: round-off error and truncation error. The truncation error is due to using a finite degree polynomial. Let $f(x) \in \mathcal{C}^{n+1}[x_0, x_n]$. The truncation error in Lagrange formula is

$$E(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n}(x - x_i); \qquad x_0 < \xi < x_n.$$

Advantage : The major advantages of Lagrange interpolation formula are

(i) it does not require the construction of a difference table.

(ii) Also, it is applicable in any part of the table and for this, the values of the argument need not be equidistant and ordered.

(iii) It is also used to find the value of the independent variable x corresponding to a given vale of the function y, known as *inverse interpolation*.

Disadvantages : The major pitfalls of the Lagrange's interpolation are

(i) If one or more additional tabular point is added at the tabular data the all Lagrangian polynomials $l_i(x)$ are to be newly constructed and without of already known polynomial $\phi(x)$. Such an interpolating polynomial is said to posses *permanence property*.

(ii) For computational purpose, it can be easily seen that Lagrange interpolation polynomial is not very efficient. This can be measured by the efficiency of any algorithm by counting the number of floating-point operations involved.

   (a) To compute $(n+1)$ factors $(x-x_0), \ldots, (x-x_n)$ for $\pi(x)$ we have $(n+1)$ subtractions and n multiplications.

   (b) To compute $(x - x_i)\pi'(x_i)$, there are $(n + 1)$ factors, so, $(n + 1)$ subtractions and n multiplications are required.

   (c) Thus, to compute $l_i(x) = \dfrac{\pi(x)}{(x - x_i)\pi'(x_i)}$, we shall require $(n+1)+(n+1) = 2(n+1)$ subtractions, $n + n = 2n$ multiplications and 1 division. Thus the total number of operations required to calculate $l_i(x)$ are $2(n + 1) + 2n + 1 = 4n + 3$.

   (d) 1 multiplication required to multiply $l_i(x)$ by y_i. Thus to calculate $l_i(x)y_i$, we have required $(4n + 3) + 1 = 4n + 4$ operations.

   (e) There are $(n + 1)$ such terms $l_i(x)y_i$, so there are $(n + 1)$ such multiplications. Thus the no of arithmetic operations required is $= (n + 1)(4n + 4)$.

(f) n additions required for $\sum_{i=0}^{n} l_i(x)y_i$.

(iii) The drawback of the method is that even for a moderately large value of n, it maybe a tedious job to represent the polynomial as power series in x due to multiplication of n factors, $(n+1)$ number of times. Therefore the total number of arithmetic operations

$$= (n+1)(4n+4) + n = 4n^3 + 12n^2 + 13n + 1.$$

This is actually referred to as the complexity of the algorithm.

EXAMPLE **7.** *Find the unique polynomial of degree* 2 *or less, which fits the data* $(0,1)$, $(1,3)$ *and* $(3,55)$. *Find the bound on the error.*

Solution: Here, $x_0 = 0, x_1 = 1, x_3 = 3$; $y_0 = 1, y_1 = 3$ and $y_2 = 55$. Using Eq. (10), the Lagrange fundamental polynomials are given by

$$l_0(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} = \frac{(x-1)(x-3)}{(-1)\cdot(-3)} = \frac{1}{3}(x^2-4x+3)$$
$$l_1(x) = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} = \frac{(x-0)(x-3)}{(1)\cdot(-2)} = \frac{1}{2}(3x-x^2)$$
$$l_2(x) = \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} = \frac{(x-0)(x-1)}{(3)\cdot(2)} = \frac{1}{6}(x^2-x)$$

Therefore, using Eq. (9), the Lagrange quadratic interpolating polynomial is given by

$$\begin{aligned}\phi(x) &= l_0(x)\cdot y_0 + l_1(x)\cdot y_1 + l_2(x)\cdot y_2\\ &= \frac{1}{3}(x^2-4x+3)\cdot 1 + \frac{1}{2}(3x-x^2)\cdot 3 + \frac{1}{6}(x^2-x)\cdot 55\\ &= \frac{1}{6}\Big[2(x^2-4x+3) + 9(3x-x^2) + 55(x^2-x)\Big] = 8x^2-6x+1.\end{aligned}$$

The truncation error in this formula is given by

$$|E(x)| = \frac{|x(x-1)(x-3)|}{3!}|f'''(\xi)| \le \frac{1}{6}\max_{0\le x\le 3}|f'''(x)|\Big[\max_{0\le x\le 3}|x(x-1)(x-3)|\Big].$$

Here $f(x)$ is unknown, so take $M_3 = \max_{0\le x\le 3}|f'''(x)|$. The maximum value of $|x(x-1)(x-3)|$ occurs at $x = 2.21525$ and $\max_{0\le x\le 3}|x(x-1)(x-3)| = 2.11261179$. Therefore, the truncation error is given by

$$|E(x)| \le \frac{1}{6}M_3\cdot(2.11261179) = 0.3521M_3.$$

EXAMPLE **8.** *Calculate* $\log(47)$ *by Lagrange interpolation formula*

x :	40	42	45	48	49	50
$f(x)$:	1.60206	1.6232493	1.6532126	1.6812413	1.690196	1.69897

Find the bound on the truncation error.

Solution: Here, the pair of points $(x_i, y_i); i = 0, 1, \cdots, 5$ as $(40, 1.60206)$, $(42, 1.6232493)$, $(45, 1.6532126)$, $(48, 1.6812413)$, $(49, 1.690196)$ and $(50, 1.69897)$ are given.

$$\begin{aligned} l_0(x) &= \frac{(x-x_1)(x-x_2)(x-x_3)(x-x_4)(x-x_5)}{(x_1-x_0)(x_2-x_0)(x_3-x_0)(x_4-x_0)(x_5-x_0)} \\ \text{or, } l_0(47) &= \frac{(47-42)(47-45)(47-48)(47-49)(47-50)}{(40-42)(40-45)(40-48)(40-49)(40-50)} = \frac{-60}{-7200} = 0.008333 \end{aligned}$$

Similarly, $l_1(47) = \frac{-84}{2016} = -0.041667$, $l_2(47) = \frac{-210}{-900} = 0.233333$, $l_3(47) = \frac{420}{288} = 1.458333$, $l_4(47) = \frac{210}{-252} = -0.833333$ and $l_5(47) = \frac{140}{800} = 0.175$. Therefore, using Eq. (9), we get

$$\begin{aligned} \log(47) &= (0.008333)\cdot(1.60206) + (-0.041667)\cdot(1.6232493) + (0.233333)\cdot(1.6532126) \\ &\quad + (1.458333)\cdot(1.6812413) + (-0.833333)\cdot(1.690196) + (0.175)\cdot(1.69897) \\ &= 1.672098. \end{aligned}$$

Hence $\log 47 = 1.6721$, correct upto 4 decimal places. The truncation error in this formula is given by

$$\begin{aligned} |E(x)| &= \frac{|(x-40)(x-42)(x-45)(x-48)(x-49)(x-50)|}{6!}|f^{(vi)}(\xi)| \\ &\leq \frac{1}{6!}\max_{40\leq x\leq 50}|f^{(vi)}(x)|\Big|(47-40)(47-42)(47-45)(47-48)(47-49)(47-50)\Big| \\ &\leq \frac{7\times 5\times 2\times 1\times 2\times 3}{6!}\max_{40\leq x\leq 50}|f^{(vi)}(x)| = \frac{7}{12}\max_{40\leq x\leq 50}|f^{(vi)}(x)|. \end{aligned}$$

$f(x) = \log_{10} x$, then $f^{(vi)}(x) = -\frac{5!}{x^6}\log_{10} e$. Therefore, $\max_{40\leq x\leq 50}|f^{(vi)}(\xi)| = 1.272347\times 10^{-8}$ and it occurs at $\xi = 40$. Therefore, the truncation error is given by

$$|E| \leq \frac{7}{12}\times 1.272347\times 10^{-8} = 7.742202\times 10^{-9}.$$

EXAMPLE **9.** *Find Lagrange's polynomial for the function* $\sin \pi x$, *when* $0, \frac{1}{6}, \frac{1}{2}$. *Also, compute the value of* $\sin\frac{\pi}{3}$ *with estimate of error.*

Solution: Let, $x_0 = 0, x_1 = \frac{1}{6}, x_2 = \frac{1}{2}$ so that $y_0 = 0, y_1 = \sin\frac{\pi}{6} = \frac{1}{2}$ and $y_3 = \sin\frac{\pi}{2} = 1$. Using Eq. (9), the Lagrange polynomial becomes,

$$\begin{aligned} \phi(x) &= \frac{\left(x-\frac{1}{6}\right)\left(x-\frac{1}{2}\right)}{\left(0-\frac{1}{6}\right)\left(0-\frac{1}{2}\right)}\times 0 + \frac{(x-0)\left(x-\frac{1}{2}\right)}{(\frac{1}{6}-0)\left(\frac{1}{6}-\frac{1}{2}\right)}\times\frac{1}{2} + \frac{(x-0)\left(x-\frac{1}{6}\right)}{(\frac{1}{2}-0)\left(\frac{1}{2}-\frac{1}{6}\right)}\times 1 \\ &= 0 - \frac{9}{2}x(2x-1) + x(6x-1) = -3x^2 + \frac{7}{2}x. \end{aligned}$$

To evaluate $\sin\dfrac{\pi}{3}$, we have $x = \frac{1}{3}$. Therefore, using the above formula, $\sin\dfrac{\pi}{3} = \phi(\dfrac{1}{3}) = -3(\dfrac{1}{3})^2 + \dfrac{7}{2}\dfrac{1}{3} \approx 0.83333$. The truncation error in this formula is given by

$$\begin{aligned} |E(x)| &= \frac{|(x-0)\left(x-\frac{1}{6}\right)\left(x-\frac{1}{2}\right)|}{4!}|f^{(iv)}(\xi)| \\ &\leq \frac{1}{4!}\max_{0\leq x\leq\frac{1}{2}}\left|(x-0)\left(x-\frac{1}{6}\right)\left(x-\frac{1}{2}\right)\right| \max_{0\leq x\leq\frac{1}{2}}|f^{(iv)}(x)|. \end{aligned}$$

Now $\max_{0\leq x\leq\frac{1}{2}}|(x-0)(x-\frac{1}{6})(x-\frac{1}{2})| = 9.7806\times10^{-3}$, occurs at $x = 0.3692$ and $\max_{0\leq x\leq\frac{1}{2}}|f^{(iv)}(x)| = 1$. Thus the estimation of truncation error is given by

$$|E| \leq \frac{1}{4!}\times 9.7806\times10^{-3}\times 1 = 4.07525\times10^{-4}.$$

The exact value of $\sin\frac{\pi}{3} = 0.86602$, so the error is 0.03269.

Listing 3: Program for Lagrange Interpolation Formula

```
% Program for Lagrange Interpolation formula
n=input('Enter the value of n    ');
x=input('Enter the values of x    ');
y=input('Enter the values of y    ');
xp=input('Enter the interpolating point xp    ');
sum=0;
for i=1:n+1
    fprintf('\n %f    %f',x(i),y(i));
end
for i=1:n+1
    prod=1;
    for j=1:n+1
        if (i~=j)
            prod=prod*(xp-x(j))/(x(i)-x(j));
        end
    end
    sum=sum+y(i)*prod;
end
fprintf('\n the value of y at x=% 14.5f is % 13.11f\n',xp,sum);
```

EXAMPLE **10.** *A certain function $f(x)$ defined on the interval $(0,1)$ is such that $f(0) = 0$, $f(\frac{1}{2}) = -1$ and $f(1) = 0$. Find the quadratic polynomial $\phi(x)$ which agrees with $f(x)$ for $x = 0, \frac{1}{2}, 1$. If $\left|\dfrac{d^3f}{dx^3}\right| \leq 1$ for $0 \leq x \leq 1$, show that $|f(x)-\phi(x)| \leq \frac{1}{12}$ for $0 \leq x \leq 1$.*

Solution: Here $f(0) = 0$, $f(\frac{1}{2}) = -1$ and $f(1) = 0$. Therefore, the Lagrange quadratic interpolating polynomial is given by

$$\begin{aligned}\phi(x) &= \frac{(x-\frac{1}{2})(x-1)}{(0-\frac{1}{2})(0-1)}\times 0 + \frac{(x-0)(x-1)}{(\frac{1}{2}-0)(\frac{1}{2}-1)}\times(-1) + \frac{(x-0)(x-\frac{1}{2})}{(1-0)(1-\frac{1}{2})}\times 0\\ &= 0-4x(x-1)+0 = -4x+4x^2.\end{aligned}$$

The truncation error in this formula is given by

$$\begin{aligned}|E(x)| &= \frac{|(x-0)\left(x-\frac{1}{2}\right)(x-1)|}{3!}|f'''(\xi)|\\ &\leq \frac{1}{3!}\max_{0\leq x\leq 1}\left|(x-0)\left(x-\frac{1}{2}\right)(x-1)\right| \max_{0\leq x\leq 1}|f'''(x)|.\end{aligned}$$

Now, for $0\leq x\leq 1$, we have,

$$\begin{aligned}|x(x-1)| &= |x(1-x)| = \left|\frac{1}{4}-(x-\frac{1}{2})^2\right| \leq \frac{1}{4} < 1\\ 0\leq 2x\leq 2 &\Rightarrow -1\leq 2x-1\leq 1 \Rightarrow |2x-1|\leq 1.\end{aligned}$$

Since, $\left|\frac{d^3f}{dx^3}\right| \leq 1$ for $0\leq x\leq 1$,, the truncation error is given by

$$|E(x)| = |f(x)-\phi(x)| \leq \frac{1}{12}\times 1\times 1 = \frac{1}{12}.$$

THEOREM **4.** *Lagrangian interpolating polynomial is unique.*

Proof: Let $\phi(x)$ and $\psi(x)$ of maximum degree n, interpolate $f(x)$ at $(n+1)$ distinct points $x_0,\cdots,x_n$. Therefore $\phi(x_i)=f(x_i)$; and $\psi(x_i)=f(x_i); i=0,1,2,\cdots,n$. Consider another polynomial $g(x)=\phi(x)-\psi(x)$. Then $g(x)$ is a polynomial of degree $\leq n$. But

$$g(x_i)=\phi(x_i)-\psi(x_i)=f(x_i)-f(x_i)=0; i=0(1)n.$$

This shows that $g(x)$ vanishes at $(n+1)$ distinct points. But $g(x)$ is a polynomial of degree n which can have only n zeros. Thus the incident is possible if $g(x)=0\Rightarrow\phi(x)=\psi(x)$.

Hence Lagrange polynomial is unique. □

THEOREM **5.** *The Lagrange interpolation formula can be written as* $y=\sum_{i=0}^{n}\frac{\pi(x)}{(x-x_i)\pi'(x_i)}y_i$, *where* $\pi(x)=(x-x_0)(x-x_1)\cdots(x-x_n)$.

Proof: Since $\pi(x)=(x-x_0)(x-x_1)\cdots(x-x_n)$, so

$$\begin{aligned}\pi'(x_0) &= \left[(x-x_1)\cdots(x-x_n)\frac{d}{dx}(x-x_0)+(x-x_0)\frac{d}{dx}\{(x-x_1)\cdots(x-x_n)\}\right]_{x=x_0}\\ &= (x_0-x_1)(x_0-x_2)\cdots(x_0-x_n).\end{aligned}$$

Using, the result we get

$$\begin{aligned}l_0(x) &= \frac{(x-x_1)(x-x_2)\cdots(x-x_n)}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_n)}\\ &= \frac{(x-x_0)(x-x_1)(x-x_2)\cdots(x-x_n)}{(x-x_0)(x_0-x_1)(x_0-x_2)\cdots(x_0-x_n)} = \frac{\pi(x)}{(x-x_0)\pi'(x_0)}\end{aligned}$$

Similarly, $\pi'(x_1) = (x_1 - x_0)(x_1 - x_2)\cdots(x_1 - x_n)$. Thus

$$l_1(x) = \frac{(x-x_0)(x-x_1)(x-x_2)\cdots(x-x_n)}{(x-x_1)(x_1-x_0)(x_1-x_2)\cdots(x_1-x_n)} = \frac{\pi(x)}{(x-x_1)\pi'(x_1)}$$

Similarly, for others. Therefore, using Eq. (9), the Lagrange interpolating formula can be written

$$\phi(x) = \sum_{i=0}^{n} l_i(x) y_i = \sum_{i=0}^{n} \frac{\pi(x)}{(x-x_i)\pi'(x_i)} y_i.$$

as a weighted sum of the given ordinates. This compact form of Lagrange formula is used in practical computations. □

THEOREM **6.** *The sum of the Lagrange coefficients is unity.*

Proof: Here $\pi(x) = (x-x_0)(x-x_1)\cdots(x-x_n)$. Thus,

$$\frac{1}{\pi(x)} = \frac{1}{(x-x_0)(x-x_1)\cdots(x-x_n)} = \frac{a_0}{x-x_0} + \frac{a_1}{x-x_1} + \cdots + \frac{a_n}{x-x_n}$$

$$\text{or,} \quad a_0(x-x_1)(x-x_2)\cdots(x-x_n) + a_1(x-x_0)(x-x_2)\cdots(x-x_n) + \cdots + a_n(x-x_0)(x-x_1)\cdots(x-x_{n-1}) = 1 \tag{12}$$

where $a_0, a_1, \cdots, a_n$ are independent of x, because to each linear and non repeated factor $x - x_r$ of $\pi(x)$ there corresponds a partial fraction $\frac{a_r}{x-x_r}$, where a_r is a constant and $\frac{1}{\pi(x)}$ can be expressed as a sum of such functions. Since this is an identity in x and is true for all values of x, we can obtain the values of a_i from (12) by putting successively $x = x_i$. Therefore,

$$a_0(x_0-x_1)(x_0-x_2)\cdots(x_0-x_n) = 1 \Rightarrow a_0\pi'(x_0) = 1 \Rightarrow a_0 = \frac{1}{\pi'(x_0)},$$
$$a_1(x_1-x_0)(x_1-x_2)\cdots(x_1-x_n) = 1 \Rightarrow a_1\pi'(x_1) = 1 \Rightarrow a_1 = \frac{1}{\pi'(x_1)},$$

and similarly $a_n = \frac{1}{\pi'(x_n)}$. So from (12) we get,

$$\frac{1}{(x-x_0)\pi'(x_0)} + \frac{1}{(x-x_1)\pi'(x_1)} + \cdots + \frac{1}{(x-x_n)\pi'(x_n)} = \frac{1}{\pi(x)}$$
$$\text{or,} \quad \frac{\pi(x)}{(x-x_0)\pi'(x_0)} + \frac{\pi(x)}{(x-x_1)\pi'(x_1)} + \cdots + \frac{\pi(x)}{(x-x_n)\pi'(x_n)} = 1$$
$$\text{or,} \quad \sum_{i=0}^{n} \frac{\pi(x)}{(x-x_i)\pi'(x_i)} = 1.$$

Practically it is very helpful for checking calculations. □

THEOREM **7.** *Lagrangian fundamental functions are invariant under a linear transformation of the independent variable.*

Proof: If we make a linear transformation $x' = \alpha + \beta x$, where α is the origin and β is called scale. Thus, the new nodes are $x_i^{'} = \alpha + \beta x_i;\ \ i = 0(1)n$. So

$$\begin{aligned} x' - x_i^{'} &= \beta(x - x_i) \Rightarrow (x - x_i) = \beta^{-1}(x' - x_i^{'}) \\ \pi(x) &= \prod_{i=0}^{n}(x - x_i) = \beta^{-n}\prod_{i=0}^{n}(x' - x_i') = \beta^{-n}\pi_1(x') \\ \pi(x_i) &= \prod_{r=0, r\neq i}^{n}(x_i - x_r) = \beta^{-n+1}\prod_{r=0, r\neq i}^{n}(x' - x_i') = \beta^{-n+1}\pi_1'(x_i'). \end{aligned}$$

Now, the Lagrangian functions are

$$\begin{aligned} l_i(x) &= \frac{\pi(x)}{(x - x_i)\pi'(x_i)} = \frac{\beta^{-n}\pi_1(x')}{\beta^{-1}(x' - x_i')\beta^{-n+1}\pi_1'(x_i)} \\ &= \frac{\pi_1(x')}{(x' - x_i')\pi_1'(x_i)} = l_i(x');\ \ i = 0(1)n \end{aligned}$$

which is independent of α and β. When the pair of data (x_i, y_i) are numerically large using this transformation we can calculate y easily. □

EXAMPLE **11.** *Determine the interpolation of $f(x)$ on the set of (distinct) points $x_0, x_1, \cdots, x_n$ by, $\sum_{k=0}^{n} l_k(x)f(x_k)$, find an expression for $\sum_{k=0}^{n} l_k(0)x_k^{n+1}$.*

Solution: The Lagrange interpolating polynomial $\phi(x)$ on the set of distinct points $x_0, x_1, \cdots, x_n$ is given by Eq. (9),

$$\phi(x) = \sum_{k=0}^{n} l_k(x)f(x_k) + \frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{i=0}^{n}(x - x_i); \qquad x_0 < \xi < x_n, \qquad (i)$$

where, the Lagrange fundamental polynomial $l_k(x)$ are given by Eq. (10). The interpolating conditions are $\phi(x) = f(x)$, at the points $x_0, x_1, \cdots, x_n$. Taking $f(x) = x^{n+1}$, so that $f^{(n+1)}(\xi) = (n+1)!$. Therefore, form (i), we get

$$x^{n+1} = \sum_{k=0}^{n} l_k(x)x_k^{n+1} + (x - x_0)(x - x_1)\cdots(x - x_n)$$

Thus, the interpolating polynomial for $f(x) = x^{n+1}$ at the interpolating points $x_0, x_1, \cdots, x_n$ is given by,

$$\sum_{k=0}^{n} l_k(x)x_k^{n+1} = x^{n+1} - (x - x_0)(x - x_1)\cdots(x - x_n).$$

Taking $x = 0$, we obtain, $\sum_{k=0}^{n} l_k(0)x_k^{n+1} = (-1)^n x_0 x_1 \cdots x_n$.

EXAMPLE **12.** Use Lagrange interpolating polynomial, find the missing term in the following table:

x:	0	1	2	3	4
y:	1	2	4	?	16

Solution:

$$\begin{aligned} l_0(x) &= \frac{(x-1)(x-2)(x-4)}{(0-1)(0-2)(0-4)} = -\frac{1}{8}(x^3-7x^2+14x-8) \\ l_1(x) &= \frac{(x-0)(x-2)(x-4)}{(1-0)(1-2)(1-4)} = \frac{1}{3}(x^3-6x^2+8x) \\ l_2(x) &= \frac{(x-0)(x-1)(x-4)}{(2-0)(0-1)(0-4)} = -\frac{1}{4}(x^3-5x^2+4x) \\ l_3(x) &= \frac{(x-0)(x-1)(x-2)}{(4-0)(4-1)(4-2)} = \frac{1}{24}(x^3-3x^2+2x) \end{aligned}$$

Thus, $\phi(x) \simeq \frac{5}{24}x^3 - \frac{1}{8}x^2 + \frac{11}{12}x + 1$. Therefore, $\phi(3) = 8.25$ and here the missing term is 8.25.

EXAMPLE **13.** *Express* $\dfrac{3x^2+x+1}{x^3-6x^2+11x-6}$ *as the sum of partial fractions.*

Solution: Here $x^3-6x^2+11x-6 = (x-1)(x-2)(x-3)$. We consider $f(x) = 3x^2+x+1$ and tabulate its values for $x = 1, 2, 3$ as:

x	1	2	3
f(x)	5	15	31

The Lagrange's interpolating polynomial of $f(x)$ for three points $(1,5)$, $(2,15)$, and $(3,31)$ is given by

$$\begin{aligned} f(x) &= \frac{(x-2)(x-3)}{2} \times 5 + \frac{(x-1)(x-3)}{-1} \times 15 + \frac{(x-1)(x-2)}{2} \times 31 \\ &= \frac{5}{2}(x-2)(x-3) - 15(x-1)(x-3) + \frac{31}{2}(x-1)(x-2). \end{aligned}$$

Hence the partial fraction representation of the given function can be written as

$$\frac{3x^2+x+1}{x^3-6x^2+11x-6} = \frac{5/2}{x-1} - \frac{15}{x-2} + \frac{31/2}{x-3}.$$

EXAMPLE **14.** *Using the three point Lagrange's formula for the value* $x_0, x_0+\epsilon, x_1$ *and taking* $\epsilon \to 0$*, show that the formula takes the from:* $f(x) = \dfrac{(x_1-x)(x+x_1-2x_0)}{(x-x_0)^2}f(x_0) + \dfrac{(x-x_0)(x_1-x)}{x_1-x_0}f'(x_0) + \dfrac{(x-x_0)^2}{(x_1-x_0)^2}f(x_1) + \dfrac{1}{6}(x-x_0)^2(x-x_1)f'''(\xi); x_0 < \xi < x_1.$

Solution: The Lagrange formula for three arguments $x_0, x_0+\epsilon, x_1$ with $\epsilon \to 0$ is

$$\begin{aligned}
f(x) &= \lim_{\epsilon\to 0}\left[\frac{(x-x_0-\epsilon)(x-x_1)}{(x_0-x_0-\epsilon)(x_0-x_1)}f(x_0)+\frac{(x-x_0)(x-x_1)}{(x_0+\epsilon-x_0)(x_0+\epsilon-x_1)}f(x_0+\epsilon)\right.\\
&\quad\left.+\frac{(x-x_0-\epsilon)(x-x_0)}{(x_1-x_0)(x_1-x_0-\epsilon)}f(x_1)+\frac{(x-x_0)(x-x_0-\epsilon)(x-x_1)}{3!}f'''(\xi)\right]\\
&= \lim_{\epsilon\to 0}\left[\frac{(x-x_0-\epsilon)(x-x_1)}{-\epsilon(x_0-x_1)}f(x_0)+\frac{(x-x_0)(x-x_1)}{(x_0+\epsilon-x_1)}\left\{\frac{f(x_0+\epsilon)-f(x_0)}{\epsilon}\right\}+\frac{(x-x_0)(x-x_1)}{\epsilon(x_0+\epsilon-x_1)}f(x_0)\right.\\
&\quad\left.+\frac{(x-x_0-\epsilon)(x-x_0)}{(x_1-x_0)(x_1-x_0-\epsilon)}f(x_1)+\frac{(x-x_0)(x-x_0-\epsilon)(x-x_1)}{3!}f'''(\xi)\right]\\
&= \lim_{\epsilon\to 0}\frac{x-x_1}{\epsilon}\left[\frac{(x-x_0)(x_0-x_1)-(x-x_0-\epsilon)(x_0-x_1+\epsilon)}{(x_0-x_1)(x_0-x_1+\epsilon)}\right]f(x_0)\\
&\quad+\frac{(x-x_0)(x-x_1)}{(x_0-x_1)}f'(x_0)+\frac{(x-x_0)^2}{(x_1-x_0)^2}f(x_1)+\frac{(x-x_0)^2(x-x_1)}{6}f'''(\xi)\\
&= \lim_{\epsilon\to 0}(x-x_1)\left[\frac{(-x-2x_0-x_1)}{(x_0-x_1)(x_0-x_1+\epsilon)}\right]f(x_0)+\frac{(x-x_0)(x-x_1)}{(x_0-x_1)}f'(x_0)\\
&\quad+\frac{(x-x_0)^2}{(x_1-x_0)^2}f(x_1)+\frac{(x-x_0)^2(x-x_1)}{6}f'''(\xi)\\
&= \left[\frac{(x_1-x)(x-2x_0+x_1)}{(x_0-x_1)^2}\right]f(x_0)+\frac{(x-x_0)(x_1-x)}{x_1-x_0}f'(x_0)\\
&\quad+\frac{(x-x_0)^2}{(x_1-x_0)^2}f(x_1)+\frac{(x-x_0)^2(x-x_1)}{6}f'''(\xi); x_0<3<x_1
\end{aligned}$$

3.1 Newton's Divided Difference Interpolation Formula

Here we derive Newton's general interpolation formula, which satisfy permanence property and from which most of the other interpolation formulas can be deducted. Suppose the function $y = f(x)$ is known at the points $x_0, x_1, \ldots, x_n$ and $y_i = f(x_i)$, $i = 0(1)n$, where, the points x_i, $i = 0(1)n$ need not be equispaced. Divided difference method introduced in this section are used to successively generate the polynomials themselves.

Now, from the definition of divided difference of first order

$$\begin{aligned}
f[x,x_0] &= \frac{y-y_0}{x-x_0} \Rightarrow y = y_0+(x-x_0)f[x,x_0]\\
\therefore \quad y &= f[x_0]+(x-x_0)f[x,x_0]. \qquad (13)
\end{aligned}$$

Eq. (13) is called the *linear Newton interpolating polynomial* with divided differences. From second order divided difference,

$$f[x,x_0,x_1] = \frac{f[x,x_0]-f[x_0,x_1]}{x-x_1} \Rightarrow f[x,x_0] = f[x_0,x_1]+(x-x_1)f[x,x_0,x_1].$$

Therefore, from Eq. (13), we get

$$\begin{aligned}
y &= y_0+(x-x_0)\Big\{f[x_0,x_1]+(x-x_1)f[x,x_0,x_1]\Big\}\\
\text{or,}\quad y &= f[x_0]+(x-x_0)f[x_0,x_1]+(x-x_0)(x-x_1)f[x,x_0,x_1]. \qquad (14)
\end{aligned}$$

From third order divided difference,

$$\begin{aligned} f[x_3,x_2,x_1,x_0] &= \frac{f[x_3,x_2,x_1]-f[x_2,x_1,x_0]}{x_3-x_0} \\ \Rightarrow \qquad f[x,x_0,x_1] &= f[x_0,x_1,x_2]+(x-x_2)f[x,x_0,x_1,x_2]. \end{aligned}$$

Therefore, from Eq. (14), we get

$$\begin{aligned} y &= f[x_0]+(x-x_0)f[x_0,x_1]+(x-x_0)(x-x_1)\Big\{f[x_0,x_1,x_2]+(x-x_2)f[x,x_0,x_1,x_2]\Big\} \\ &= f[x_0]+(x-x_0)f[x_0,x_1]+(x-x_0)(x-x_1)f[x_0,x_1,x_2] \\ &\quad +(x-x_0)(x-x_1)(x-x_2)f[x,x_0,x_1,x_2]. \qquad (15) \end{aligned}$$

Continuing in this manner (by mathematical induction) the Newton divided difference formula for the interpolating polynomial is

$$\begin{aligned} y &= f[x_0]+(x-x_0)f[x_0,x_1]+(x-x_0)(x-x_1)f[x_0,x_1,x_2]+\cdots \\ &\quad +(x-x_0)(x-x_1)\cdots(x-x_{n-2})f[x_0,x_1,\cdots,x_{n-1}] \\ &\quad +(x-x_0)(x-x_1)\cdots(x-x_{n-1})f[x,x_0,x_1,\cdots,x_n] \\ &= f[x_0]+\sum_{i=1}^{n} f[x_0,x_1,\cdots,x_i]\prod_{j=0}^{i-1}(x-x_j)+f[x,x_0,x_1,\cdots,x_n]\prod_{i=0}^{n}(x-x_i). \qquad (16) \end{aligned}$$

Therefore the *Newton's divided difference formula* for the interpolating polynomial is given by

$$\phi(x) = f[x_0]+\sum_{i=1}^{n} f[x_0,x_1,\cdots,x_i]\prod_{j=0}^{i-1}(x-x_j). \qquad (17)$$

This formula is specially suitable for computation of an optimal degree interpolating polynomial. By choosing the proper order and assuming that the spacing between points is uniform, most of the classical interpolation formulas can be derived from Newton's general formula.

Using the notations as in (**??**), the formula (17) can be written in the following form

$$\begin{aligned} f(x) &= d_{0,0}+d_{1,1}(x-x_0)+d_{2,2}(x-x_0)(x-x_1) \\ &\quad +\cdots+d_{n,n}(x-x_0)(x-x_1)\cdots(x-x_{n-1}). \end{aligned}$$

If $f(x)$ is approximated by $\phi(x)$, then the truncation error $E(x)$ is given by

$$\begin{aligned} E(x) &= f(x)-\phi(x) = f[x,x_0,x_1,\cdots,x_n]\prod_{i=0}^{n}(x-x_i) \\ &= f[x,x_0,x_1,\cdots,x_n]\pi(x). \qquad (18) \end{aligned}$$

This form of the truncation error will be useful in considering the accuracy of numerical differentiation and integration formulas.

THEOREM **8.** *Let $f(x) \in \mathcal{C}^n[a,b]$ and $x_0,x_1,\cdots,x_n$ are $(n+1)$ distinct points in $[a,b]$. Then there exists $\xi \in (a,b)$ such that $f[x_0,x_1,\cdots,x_n] = \frac{1}{n!}f^{(n)}(\xi)$.*

Proof: Let $E(x) = f(x) - \phi(x)$. Now $E(x) \in \mathcal{C}^n[a,b]$ as $f(x)$ and $\phi(x)$ is are so and

$$E(x_i) = f(x_i) - \phi(x_i) = 0;\;\; i = 0(1)n.$$

So the function $E(x)$ has $(n+1)$ distinct zeros in $[a,b]$. Generalised Rolle's theorem implies that, a number $\xi \in (a,b)$ exists with $E^{(n)}(\xi) = f^{(n)}(\xi) - \phi^{(n)}(\xi) = 0$.

Since $\phi(x)$ is a polynomial of degree n whose leading coefficient is $f[x_0, x_1, \cdots, x_n]$, so

$$\phi^{(n)}(x) = n! f[x_0, x_1, \cdots, x_n]$$

for all values of x. As a consequence

$$\begin{aligned} E^{(n)}(\xi) &= 0 = f^{(n)}(\xi) - n! f[x_0, x_1, \cdots, x_n] \\ \Rightarrow \quad f[x_0, \cdots, x_n] &= \frac{f^{(n)}(\xi)}{n!}; x_0 < \xi < x_n. \end{aligned} \tag{19}$$

This proves the theorem. □

Using this theorem, the error formula (18) can be written as

$$E(x) = \frac{\pi(x)}{n!} f^{(n)}(\xi);\;\; \min\{x, x_0, \cdots, x_n\} < \xi < \max\{x, x_0, \cdots, x_n\}.$$

Deduction 3.2. Newton's divided difference formula can be expressed in a simplified form when the nodes $x_0, x_1, \cdots, x_n$ are arranged consecutively with equal spacing. In this case, we introduce the notation $h = x_{i+1} - x_i; i = 0(1)n-1$ and $x = x_0 + uh$. Then the difference $x - x_i$ is $x - x_i = (u-i)h$. So Eq. (17) becomes

$$\begin{aligned} \phi(x) &= \phi(x_0 + uh) = f[x_0] + uhf[x_0, x_1] + u(u-1)h^2 f[x_0, x_1, x_2] \\ &\quad + \cdots + u(u-1)\cdots(u-n+1)h^n f[x_0, x_1, \cdots, x_n] \\ &= f[x_0] + \sum_{k=1}^{n} u(u-1)\cdots(u-k+1)h^k f[x_0, x_1, \cdots, x_k] \end{aligned} \tag{20}$$

The Newton's forward-difference formula, is constructed by making use of the forward difference notation Δ. With this notation

$$\begin{aligned} f[x_0, x_1] &= \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{y_1 - y_0}{h} = \frac{1}{h}\Delta y_0 \\ f[x_0, x_1, x_2] &= \frac{f[x_0, x_1] - f[x_1, x_2]}{x_0 - x_2} = \frac{\frac{1}{h}\Delta y_0 - \frac{1}{h}\Delta y_1}{-2h} = \frac{1}{2!h^2}\Delta^2 y_0 \end{aligned}$$

and in general, $f[x_0, x_1, \cdots, x_n] = \dfrac{1}{n!h^n}\Delta^n y_0$. Substituting the values in Eq. (20), we get,

$$\phi(x) = f[x_0] + \sum_{k=1}^{n} u(u-1)\cdots(u-k+1)h^k \frac{1}{k!h^k}\Delta^k y_0 = y_0 + \sum_{k=1}^{n} \binom{u}{k} \Delta^k y_0$$

which is Newton's forward interpolation formula.

Deduction 3.3. Here also $x_i = x_0 + ih, (h > 0); i = 0(1)n$. If the interpolating nodes are recorded from last to first as $x_n, x_{n-1}, \cdots, x_0$, we can write the interpolatory formula as

$$\begin{aligned}\phi(x) &= f[x_n] + (x - x_n)f[x_n, x_{n-1}] + (x - x_n)(x - x_{n-1})f[x_n, x_{n-1}, x_{n-2}] \\ &\quad + \cdots + (x - x_n)(x - x_{n-1}) \cdots (x - x_1)f[x_n, x_{n-1}, \cdots, x_0]\end{aligned}$$

If in addition, the nodes are equally spaced with $x = x_n + vh$ and $x = x_i + (v + n - i)h$, then

$$\begin{aligned}\phi(x) &= \phi(x_n + vh) = f[x_n] + vhf[x_n, x_{n-1}] + v(v + 1)h^2 f[x_n, x_{n-1}, x_{n-2}] \\ &\quad + \cdots + v(v + 1) \cdots (v + n - 1)h^n f[x_n, x_{n-1}, \cdots, x_0] \\ &= f[x_n] + \sum_{k=1}^{n} v(v + 1) \cdots (v + k - 1)h^k f[x_n, x_{n-1}, \cdots, x_0] \qquad (21)\end{aligned}$$

The Newton's backward-difference formula, is constructed by making use of the backward difference notation ∇. With this notation

$$f[x_n, x_{n-1}] = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}} = \frac{y_n - y_{n-1}}{h} = \frac{1}{h}\nabla y_n$$

$$f[x_n, x_{n-1}, x_{n-2}] = \frac{f[x_n, x_{n-1}] - f[x_{n-1}, x_{n-2}]}{x_n - x_{n-2}} = \frac{\frac{1}{h}\nabla y_n - \frac{1}{h}\nabla y_{n-1}}{2h} = \frac{1}{2!h^2}\nabla^2 y_n$$

and in general, $f[x_n, x_{n-1}, \cdots, x_0] = \dfrac{1}{n!h^n}\nabla^n y_n$. Substituting the values in Eq. (21), we get,

$$\begin{aligned}\phi(x) &= f[x_n] + v\nabla y_n + \frac{v(v + 1)}{2}\nabla^2 y_n + \cdots + \frac{v(v + 1) \cdots (v + n - 1)}{n!}\nabla^n y_n \\ &= y_n + \sum_{k=1}^{n}(-1)^k \binom{-v}{k}\Delta^k y_n; \qquad y_n = f[x_n]\end{aligned}$$

which is the *Newton's backward interpolation formula.*

Deduction 3.4. Let the nodes be $x_i = x_0 + ih, h > 0; i = 0, \pm 1, \pm 2, \cdots, \pm m$. The divided difference formula with the sequence of arguments is

$$\begin{aligned}f(x) &= f[x_0] + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_{-1}] \\ &\quad + (x - x_0)(x - x_1)(x - x_1)f[x_0, x_1, x_{-1}, x_2] + \cdots \\ &\quad + (x - x_0)(x - x_1) \cdots (x - x_m)f[x_0, x_1, x_{-1}, \cdots, x_{-m}] \\ &= f[x_0] + \sum_{k=0}^{m-1}(x - x_0)(x - x_1)...(x - x_{-k})f[x_0, x_1, x_{-1}, \cdots, x_{k+1}] \\ &\quad + \sum_{k=1}^{m}(x - x_0)(x - x_1) \cdots (x - x_k)f[x_0, x_1, x_{-1}, \cdots, x_{-k}] + R_{n+1} \\ &= y_0 + \sum_{k=0}^{m-1}(x - x_0)(x - x_1) \cdots (x - x_{-k})\frac{\Delta^{2k+1}y_{-k}}{(2k + 1)!h^{2k+1}} \\ &\quad + \sum_{k=1}^{m}(x - x_0)(x - x_1) \cdots (x - x_k)\frac{\Delta^{2k}y_{-k}}{(2k)!h^{2k}} + R_{n+1}\end{aligned}$$

where the remainder term is

$$\begin{aligned} R_{n+1}(x) &= (x-x_0)(x-x_1)\cdots(x-x_{-m})f[x_0,x_1,x_{-1},\cdots,x_{-m}] \\ &= (x-x_0)(x-x_1)\cdots(x-x_{-m})\frac{f^{2m+1}(\xi)}{(2m+1)!}; \end{aligned}$$

$a = min\{x, x_{-m}\} < \xi < b = max\{x, x_{-m}\}$, which is the Gauss's forwarded formula with an odd number of nodes. The other cases being similar.

EXAMPLE **15.** *Using Newton's divided difference interpolation formula, find* $y(3.4)$*, given*

$x:$	2.5	2.8	3.0	3.1	3.6
$y:$	12.182494	16.444647	20.085537	22.197951	36.598234

Solution: Let us first construct following divided difference table:

x	y	$f(,)$	$f(,,)$	$f(,,,)$	
2.5	12.182494				
		14.207177			
2.8	16.444647		7.994546		
		18.204450		2.896256	
3.0	20.085537		9.732300		0.846293
		21.124140		3.827178	
3.1	22.197951		12.794043		
		28.800566			
3.6	36.598234				

Using Newton's divided difference formula (17), we get,

$$\begin{aligned} y(3.4) &= 12.182494 + (3.4-2.5)\times 14.207177 + (3.4-2.5)(3.4-2.8)\times 7.994546 \\ &\quad +(3.4-2.5)(3.4-2.8)(3.4-3.0)\times 2.896256 \\ &\quad +(3.4-2.5)(3.4-2.8)(3.4-3.0)(3.4-3.1)\times 0.846293 = 29.966439. \end{aligned}$$

Deduction 3.5. Divided difference formula is similar to Taylor's's series. Here, we consider the relation

$$f^n[x_0] = \frac{d^n}{dx^n}f[x_0] = n!f[x_0,\ldots,x_0].$$

as in Eq. (**??**). In Eq. (19), if we take the limit $x_i \to x_0 (i = 1, 2, \cdots, n)$ then

$$f[x, x_0, x_0, \cdots, x_0] = \frac{f^{(n+1)}(\xi)}{(n+1)!}; a < \xi < b.$$

Hence the formula (17) reduces to

$$f(x) = f(x_0) + \frac{x-x_0}{1!}f'(x_0) + \cdots + \frac{(x-x_0)^{n+1}}{(n+1)!}f^{(n+1)}(\xi); a < \xi < b$$

which is the Taylor's's series with the truncation error term in Lagrange form.

Deduction 3.6. Lagrange interpolation formula can be derived from divided difference. Let $\phi(x)$ denotes a polynomial of the nth degree which takes the values $y_0, \cdots, y_n$ when x has the values $x_0, \cdots, x_n$ respectively. Then the $(n+1)th$ differences of the polynomial is 0 i.e., $f[x, x_0, \cdots, x_n] = 0$

$$\text{or,} \quad \frac{y}{(x-x_0)\cdots(x-x_n)} + \frac{y_n}{(x_n-x)(x_n-x_0)\cdots(x_n-x_{n-1})} + \cdots + \frac{y_0}{(x_0-x)(x_0-x_1)\cdots(x_0-x_n)} = 0$$

$$\text{or, } y = \frac{(x-x_1)\cdots(x-x_n)}{(x_0-x_1)\cdots(x_0-x_n)}y_0 + \cdots + \frac{(x-x_1)\cdots(x-x_{n-1})}{(x_n-x_0)\cdots(x_n-x_{n-1})}y_n$$

which is the *Lagrange interpolation formula.* Note that, since the interpolating polynomial is unique, Lagrange and Newton's divided difference polynomials are two different forms of the same polynomial.

EXAMPLE **16.** *For the following table, find the interpolation polynomial using*

x	:	0	2	4	8
$f(x)$	:	3	8	11	18

(i) Lagrange's formula and (ii) Newton's divided difference formula, and hence show that both represent same interpolating polynomial.

Solution: (i) The Lagrange's interpolation polynomial is

$$\begin{aligned}\phi(x) &= \frac{(x-2)(x-4)(x-8)}{(0-2)(0-4)(0-8)} \times 3 + \frac{(x-0)(x-4)(x-8)}{(2-0)(2-4)(2-8)} \times 8 \\ &\quad + \frac{(x-0)(x-2)(x-8)}{(4-0)(4-2)(4-8)} \times 11 + \frac{(x-0)(x-2)(x-4)}{(8-0)(8-2)(8-4)} \times 19 \\ &= \frac{1}{24}x^3 - \frac{1}{2}x^2 + \frac{10}{3}x + 3.\end{aligned}$$

(ii) The divided difference table is

x	$f(x)$	1st divided difference	2nd divided difference	3rd divided difference
0	3			
2	8	5/2		
4	11	3/2	$-1/4$	
8	19	2	1/12	1/24

Newton's divided difference polynomial is

$$\begin{aligned}\phi(x) &= 3 + (x-0) \times \frac{5}{2} + (x-0)(x-2) \times (-\frac{1}{4}) + (x-0)(x-2)(x-4) \times \frac{1}{24} \\ &= 3 + \frac{5}{2}x - \frac{1}{4}(x^2-2x) + \frac{1}{24}(x^3 - 6x^2 + 8x) \\ &= \frac{1}{24}x^3 - \frac{1}{2}x^2 + \frac{10}{3}x + 3.\end{aligned}$$

Thus, the interpolating polynomial by both Lagrange's and Newton's divided difference formulae are one and same.

Listing 4: Program for Newton Divided Difference Formula

```
% Program for Newton's Divided Difference Interpolation Formula
n=input('Enter the number of subintervals n  ');
x=input('Enter the values of x  ');
y=input('Enter the values of y ');
xp=input('Enter the interpolating point xp  ');
for i=1:n+1
    fprintf('\n %d    %d',x(i),y(i));
end
for k=1:n+1
    d(k,1)=y(k);
end
for i=2:n+1
    for k=i:n+1
        d(k,i)=(d(k,i-1)-d(k-1,i-1))/(x(k)-x(k+1-i));
    end
end
sum=d(n+1,n+1);
for k=n:-1:1
        sum=sum*(xp-x(k))+d(k,k);
end
    fprintf('\n the value of y at x=% 14.5f is % 13.11f\n',xp,sum);
```