Interpolation

Polynomial Interpolation

The polynomial interpolation problem, also called Lagrange interpolation, can be described as follows: Given $n+1$ data points $(x_i, y_i),\ i = 0, 1, \ldots, n$, find a polynomial P of lowest possible degree such that

$$y_i = P(x_i), \qquad i = 0, 1, \ldots, n.$$

Such a polynomial is said to interpolate the data. Here y_i may be the value of some unknown function f at x_i, i.e., $y_i = f(x_i)$.

One reason for considering the class of polynomials in approximation of functions is that they uniformly approximate continuous function.

Theorem **(Weierstrass Approximation Theorem)** *Suppose that f is defined and continuous on $[a, b]$. For any $\varepsilon > 0$, there exists a polynomial $P(x)$, defined on $[a, b]$, with the property that*

$$|f(x) - P(x)| < \varepsilon, \qquad \text{for all } x \text{ in } [a, b].$$

Another important reason for considering the class of polynomials in approximation of functions is that the derivatives and indefinite integral of a polynomial are easy to determine and are also polynomials. For these reasons, polynomials are often used for approximating continuous functions.

Existence And Uniqueness

The theorem that governs the polynomial interpolation problem stats

Theorem *If (x_i, y_i), $x_i, y_i \in \mathbb{R}$, $i = 0, 1, \ldots, n$, are $n+1$ distinct pairs of data point, then there is a unique polynomial P_n of degree at most n such that*

$$P_n(x_i) = y_i, \quad (0 \leq i \leq n).$$

Proof: **Existence:** Proof by mathematical induction.

The theorem clearly holds for

$n = 0$ (only one data point (x_0, y_0))

since one may choose the constant polynomial

$P_0(x) = y_0$ for all x.

Assume that the theorem holds for $n \leq k$,

that is, there is a polynomial P_k, $\deg(P_k) \leq k$,

such that $y_i = P_k(x_i)$, for $0 \leq i \leq k$.

Next we try to

construct a polynomial of degree at most $k+1$ to interpolate (x_i, y_i), $0 \leq i \leq k+1$.

$$P_{k+1}(x) = P_k(x) + c(x - x_0)(x - x_1) \cdots (x - x_k),$$

where

$$c = \frac{y_{k+1} - P_k(x_{k+1})}{(x_{k+1} - x_0)(x_{k+1} - x_1) \cdots (x_{k+1} - x_k)}.$$

Since x_i are distinct, the polynomial $P_{k+1}(x)$ is well-defined and $\deg(P_{k+1}) \leq k+1$.

easy to verify that

$$P_{k+1}(x_i) = y_i, \quad 0 \leq i \leq k+1.$$

Uniqueness: Suppose there are two such polynomials P_n and Q_n satisfying

$$P_n(x_i) = y_i, \quad (0 \leq i \leq n).$$

Define $\quad S_n(x) = P_n(x) - Q_n(x).$

Since both $\deg(P_n) \leq n$ and $\deg(Q_n) \leq n$, $\deg(S_n) \leq n$. Moreover

$$S_n(x_i) = P_n(x_i) - Q_n(x_i) = y_i - y_i = 0,$$

for $0 \leq i \leq n$. This means that S_n has at least $n+1$ zeros, it therefore must be $S_n = 0$. Hence $P_n = Q_n$. ■

Naive Approach for Polynomial Interpolation

Suppose we require the interpolating polynomial to be expressed in power of x as

$$P_n(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_n x^n.$$

The straightforward approach for solving the polynomial interpolation is try to determine the coefficients $c_0, c_1, \ldots c_n$ using the conditions

$$P_n(x_i) = y_i, \qquad i = 0, 1, 2, \ldots, n.$$

These lead to

$$\begin{aligned} c_0 + c_1 x_1 + c_2 x_1^2 + \cdots + c_n x_1^n &= y_0, \\ c_0 + c_1 x_2 + c_2 x_2^2 + \cdots + c_n x_2^n &= y_1, \\ &\vdots \\ c_0 + c_1 x_n + c_2 x_n^2 + \cdots + c_n x_n^n &= y_n, \end{aligned}$$

which is equivalent to the system of linear equations:

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & & & & \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

The coefficient matrix is called Vandermonde matrix.

It can be proved that:

$$\det \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & & & & \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix} = \prod_{0 \le j < k \le n} (x_k - x_j).$$

Since x_i are distinct, the Vandermonde system is nonsingular, and solving this s
gives the coefficients $c_0, c_1, \ldots, c_n$ which solves the polynomial interpolation problem

Remark:

However, a Vandermonde matrix is often very ill-conditioned, thus the computed solutions c_i will be inaccurate. Moreover, the amount of computational cost, including forming the matrix, factorization, and triangular substitutions, is excessive. Therefore, this approach is not recommended.

Lagrange Form and Neville's Method

Polynomial interpolation in Lagrange form expresses the polynomial as

$$P_n(x) = y_0 L_0(x) + y_1 L_1(x) + \cdots + y_n L_n(x) = \sum_{k=0}^{n} y_k L_k(x),$$

where

$$L_k(x) = \frac{(x - x_0) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_n)}{(x_k - x_0) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_n)} = \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{x - x_i}{x_k - x_i},$$

are known as cardinal polynomials.

Example *Given the following 4 data points,*

x_i	0	1	3	5
y_i	1	2	6	7

find a polynomial in Lagrange form to interpolate these data.

Sol: The cardinal functions are

$$\begin{aligned}
L_0(x) &= \frac{(x-1)(x-3)(x-5)}{(0-1)(0-3)(0-5)} = -\frac{1}{15}(x-1)(x-3)(x-5),\\
L_1(x) &= \frac{(x-0)(x-3)(x-5)}{(1-0)(1-3)(1-5)} = \frac{1}{8}x(x-3)(x-5),\\
L_2(x) &= \frac{(x-0)(x-1)(x-5)}{(3-0)(3-1)(3-5)} = -\frac{1}{12}x(x-1)(x-5),\\
L_3(x) &= \frac{(x-0)(x-1)(x-3)}{(5-0)(5-1)(5-3)} = \frac{1}{40}x(x-1)(x-3).
\end{aligned}$$

The interpolating polynomial in the Lagrange form is

$$P_3(x) = L_0(x) + 2L_1(x) + 6L_2(x) + 7L_3(x).$$

Remarks:

One difficulty for the interpolating polynomial in the Lagrange form is that if more data points are added to the interpolation problem, all the cardinal functions have to be recalculated. We will now derive the interpolating polynomials in a manner that uses the previous calculations to greater advantage.

Definition . *Let (x_i, y_i), $0 \le i \le n$, where $y_i = f(x_i)$ for some unknown function f, be $(n+1)$ given distinct data points. Suppose that $m_1, m_2, \ldots, m_k$ are k distinct integers with $0 \le m_i \le n$ for each i. The Lagrange polynomial that interpolates f at the k points $x_{m_1}, x_{m_2}, \ldots, x_{m_k}$ is denoted $P_{m_1,m_2,\ldots,m_k}(x)$.*

Neville's Method

Theorem 6.3 *Let f be defined at distinct points $x_0, x_1, \ldots, x_k$, and $0 \leq, i, j \leq k$, . . . Then*

$$P_{0,1,\ldots,k}(x) = \frac{(x - x_j)}{(x_i - x_j)} P_{0,1,\ldots,j-1,j+1,\ldots,k}(x) - \frac{(x - x_i)}{(x_i - x_j)} P_{0,1,\ldots,i-1,i+1,\ldots,k}(x) \tag{6.8}$$

describes the k-th Lagrange polynomial that interpolates f at the $k+1$ points $x_0, x_1, \ldots, x_k$.

Proof: Since $\deg(P_{0,1,\ldots,j-1,j+1,\ldots,k}) \leq k-1$ and $\deg(P_{0,1,\ldots,i-1,i+1,\ldots,k}) \leq k-1$, by definition $\deg(P_{0,1,\ldots,k}) \leq k$. If $r \neq i, j$, then

$$\begin{aligned} P_{0,1,\ldots,k}(x_r) &= \frac{(x_r - x_j)}{(x_i - x_j)} P_{0,1,\ldots,j-1,j+1,\ldots,k}(x_r) - \frac{(x_r - x_i)}{(x_i - x_j)} P_{0,1,\ldots,i-1,i+1,\ldots,k}(x_r) \\ &= \frac{(x_r - x_j)}{(x_i - x_j)} f(x_r) - \frac{(x_r - x_i)}{(x_i - x_j)} f(x_r) = f(x_r). \end{aligned}$$

Moreover

$$P_{0,1,\ldots,k}(x_i) = \frac{(x_i - x_j)}{(x_i - x_j)} P_{0,1,\ldots,j-1,j+1,\ldots,k}(x_i) - \frac{(x_i - x_i)}{(x_i - x_j)} P_{0,1,\ldots,i-1,i+1,\ldots,k}(x_i) = f(x_i)$$

and

$$P_{0,1,\ldots,k}(x_j) = \frac{(x_j - x_j)}{(x_i - x_j)} P_{0,1,\ldots,j-1,j+1,\ldots,k}(x_j) - \frac{(x_j - x_i)}{(x_i - x_j)} P_{0,1,\ldots,i-1,i+1,\ldots,k}(x_j) = f(x_j).$$

Therefore $P_{0,1,\ldots,k}(x)$ agrees with f at all points $x_0, x_1, \ldots, x_k$. By the uniqueness theorem, $P_{0,1,\ldots,k}(x)$ is the k-th Lagrange polynomial that interpolates f at the $k+1$ points $x_0, x_1, \ldots, x_k$. ■

The theorem implies that the Lagrange interpolating polynomial can be generated recursively.

The procedure is called the **Neville's method**.

For example, the polynomials can be computed in a manner as shown in the following table.

x_0	$P_0 = Q_{0,0}$				
x_1	$P_1 = Q_{1,0}$	$P_{0,1} = Q_{1,1}$			
x_2	$P_2 = Q_{2,0}$	$P_{1,2} = Q_{2,1}$	$P_{0,1,2} = Q_{2,2}$		
x_3	$P_3 = Q_{3,0}$	$P_{2,3} = Q_{3,1}$	$P_{1,2,3} = Q_{3,2}$	$P_{0,1,2,3} = Q_{3,3}$	
x_4	$P_4 = Q_{4,0}$	$P_{3,4} = Q_{4,1}$	$P_{2,3,4} = Q_{4,2}$	$P_{1,2,3,4} = Q_{4,3}$	$P_{0,1,2,3,4} = Q_{4,4}$

For ease of notation, we denote, in the table

$$Q_{i,j} = P_{i-j,i-j+1,\ldots,i-1,i}.$$

Hence $Q_{i,j}$, $0 \leq j \leq i$, denotes the interpolating polynomial of degree j on the $j+1$ points $x_{i-j}, x_{i-j+1}, \ldots, x_{i-1}, x_i$.

Note that the Neville's method is usually used to generated successively higher degree polynomial approximations at a specified point, but not to generate the polynomials themselves.

Newton's Form of Polynomial Interpolation

Since there is one and only one polynomial of degree $\leq n$ that takes prescribed values at given $n+1$ distinct points, various algorithms will produce the same polynomial but in different forms.

For example, the polynomial can be constructed as a linear combination of the basic polynomials $1, x, x^2, \ldots, x^n$.

For numerical work, it is preferable to use following basis

$$\begin{aligned} q_0(x) &= 1, \\ q_1(x) &= (x - x_0), \\ q_2(x) &= (x - x_0)(x - x_1), \\ &\vdots \\ q_n(x) &= (x - x_0)(x - x_1) \cdots (x - x_{n-1}), \end{aligned}$$

and express $P(x)$ as

$$\begin{aligned} P_n(x) &= c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \cdots + c_n(x - x_0)(x - x_1) \cdots (x - x_{n-1}). \\ &= \sum_{k=0}^{n} c_k q_k(x) \end{aligned}$$

This polynomial is called the interpolating polynomial in Newton's form.
The interpolation conditions

$$P_n(x_i) = \sum_{k=0}^{n} c_k q_k(x_i) = y_i, \qquad i = 0, 1, \ldots, n,$$

give rise to $n+1$ equations with $n+1$ parameters $c_0, c_1, \ldots, c_n$ to be determined

$$\begin{aligned} c_0 &= y_0, \\ c_0 + c_1(x_1 - x_0) &= y_1, \\ c_0 + c_1(x_2 - x_0) + c_2(x_2 - x_0)(x_2 - x_1) &= y_2, \\ &\vdots \\ c_0 + c_1(x_n - x_0) + c_2(x_n - x_0)(x_n - x_1) + \cdots + c_n(x_n - x_0) \cdots (x_n - x_{n-1}) &= y_n. \end{aligned}$$

In matrix equation, we have a triangular linear system

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & x_1 - x_0 & 0 & \cdots & 0 \\ 1 & x_2 - x_0 & (x_2 - x_0)(x_2 - x_1) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n - x_0 & (x_n - x_0)(x_n - x_1) & \cdots & (x_n - x_0)\cdots(x_n - x_{n-1}) \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Use forward substitution, we can solve for $c_0, c_1, \ldots, c_n$. Computational work involved in this approach includes forming and solving the system.

Example *Given the following* 4 *points* ($n = 3$)

x_i	0	1	3	5
y_i	1	2	6	7

find a polynomial of degree 3 *in Newton's form to interpolate these data.*

Sol: Write

$$\begin{aligned} P(x) &= c_0 + c_1(x-0) + c_2(x-0)(x-1) + c_3(x-0)(x-1)(x-3) \\ &= c_0 + c_1 x + c_2 x(x-1) + c_3 x(x-1)(x-3). \end{aligned}$$

Plug in the interpolation conditions,

$$\begin{aligned} P(0) &= c_0 = 1. \\ P(1) &= c_0 + c_1 = 2. \\ P(3) &= c_0 + 3c_1 + 6c_2 = 6. \\ P(5) &= c_0 + 5c_1 + 20c_2 + 40c_3 = 7, \end{aligned}$$

which lead to

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 3 & 6 & 0 \\ 1 & 5 & 20 & 40 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 6 \\ 7 \end{bmatrix} \quad \Longrightarrow \quad \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \frac{1}{3} \\ -\frac{17}{120} \end{bmatrix}.$$

Thus

$$P(x) = 1 + x + \frac{1}{3}x(x-1) - \frac{17}{120}x(x-1)(x-3).$$

Divided Differences Scheme

With proper arrangement, the triangular linear system in the previous subsection can be solved with the formulations

$$
\begin{aligned}
c_0 &= f(x_0) \\
c_1 &= \frac{f(x_1) - f(x_0)}{x_1 - x_0} \\
c_2 &= \frac{1}{x_2 - x_0}\left[\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}\right] \\
&\vdots \\
&\text{and so on.}
\end{aligned}
$$

and so on. An efficient procedure, called the divided-difference method, can be derived for computing the $n + 1$ coefficients $c_0, c_1, \cdots, c_n$ in the Newton's form interpolating polynomial.

Divided Differences Scheme

First we introduce the divided-difference notation.

The zero divided difference of the function f with respect to x

$$f[x_i] = f(x_i),$$

is simply the value of f at x_i.

The first divided difference of f with respect to x_i and x_{i+1} is denoted and defined as

$$f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}.$$

The second divided difference of f is defined as

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

In general, the k-th divided difference relative to $x_i, x_{i+1}, \ldots, x_{i+k}$ is given by

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

As might be expected from the evaluation of $c_0, c_1, \ldots, c_n$ in the Newton form interpolating polynomial, the required coefficients are given by

$$c_k = f[x_0, x_1, \ldots, x_k], \qquad k = 0, 1, \ldots, n.$$

Therefore the interpolating polynomial in Newton's form can be expressed as

$$P_n(x) = f[x_0] + \sum_{k=1}^{n} f[x_0, x_1, \ldots, x_k](x - x_0)(x - x_1)\cdots(x - x_{k-1}).$$

This formula is known as the Newton's interpolatory divided-difference formula.

This formula is known as the Newton's interpolatory divided-difference formula. The determination of the divided differences is best described by using tabular form as shown in the following table as an example.

x_i	$k=0$	$k=1$	$k=2$	$k=3$
x_0	$f[x_0]$			
		$f[x_0, x_1]$		
x_1	$f[x_1]$		$f[x_0, x_1, x_2]$	
		$f[x_1, x_2]$		$f[x_0, x_1, x_2, x_3]$
x_2	$f[x_2]$		$f[x_1, x_2, x_3]$	
		$f[x_2, x_3]$		
x_3	$f[x_3]$			

Example *Given the following 4 points ($n = 3$)*

x_i	0	1	3	5
y_i	1	2	6	7

find a polynomial of degree 3 in Newton's form to interpolate these data.

Sol:

x_i	$k=0$	$k=1$	$k=2$	$k=3$
0	1			
		> 1		
1	2		$> \frac{1}{3}$	
		> 2		$> -\frac{17}{120}$
3	6		$> -\frac{3}{8}$	
		$> \frac{1}{2}$		
5	7			

So,

$$P(x) = 1 + x + \frac{1}{3}x(x-1) - \frac{17}{120}x(x-1)(x-3).$$

■

Note that x_i can be reordered, but must be distinct. When the order of x_i are changed, one obtains the same polynomial but in different form.

Example *Given the following 4 points ($n = 3$)*

x_i	3	1	5	0
y_i	6	2	7	1

find a polynomial of degree 3 in Newton's form to interpolate these data.

Sol:

x_i	$k=0$	$k=1$	$k=2$	$k=3$
3	6			
		> 2		
1	2		$> -\frac{3}{8}$	
		$> \frac{5}{4}$		$> -\frac{17}{120}$
5	7		$> \frac{1}{20}$	
		$> \frac{6}{5}$		
0	1			

So,

$$P(x) = 6 + 2(x-3) - \frac{3}{8}(x-3)(x-1) - \frac{17}{120}(x-3)(x-1)(x-5).$$

■

Note that $c_0, c_1, \ldots, c_{n-1}$ are changed, but c_n is unchanged, and the polynomial is expressed in different form.

Recall:

$$P_n(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \cdots + c_n(x - x_0)(x - x_1)\cdots(x - x_{n-1}).$$

$$= \sum_{k=0}^{n} c_k q_k(x) \quad , \qquad q_n(x) = (x - x_0)(x - x_1)\cdots(x - x_{n-1})$$

$$c_k = f[x_0, x_1, \ldots, x_k], \qquad k = 0, 1, \ldots, n.$$

$$P_n(x) = f[x_0] + \sum_{k=1}^{n} f[x_0, x_1, \ldots, x_k](x - x_0)(x - x_1)\cdots(x - x_{k-1}).$$

Algorithm (Divided-Difference Algorithm) *Given $n+1$ distinct points (x_i, y_i), $i = 0, 1, \ldots, n$ his algorithm computes the coefficients of the interpolating polynomial in Newton's form using an 1-d array of size $n + 1$..*

for $i = 0, 1, \ldots, n$ **do**
  $c(i) = y(i) = f(x(i))$
end for
for $k = 1, \ldots, n$ **do**
  for $i = n, n - 1, \ldots, k$ **do**
    $c(i) = \dfrac{c(i) - c(i-1)}{x(i) - x(i-k)}$
  end for
end for

$$P_n(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \cdots + c_n(x - x_0)(x - x_1)\cdots(x - x_{n-1}).$$

$$= \sum_{k=0}^{n} c_k q_k(x) \qquad q_n(x) = (x - x_0)(x - x_1)\cdots(x - x_{n-1})$$

Generalized Rolle's Theorem

Recall Rolle's Theorem: Let f be continuous on $[a,b]$ and differentiable on (a,b). If $f(a)=f(b)$ then there exists $c\in(a,b)$ such that $f'(c)=0$. The Generalized Rolle's Theorem extends this idea to higher order derivatives:

Generalized Rolle's Theorem: Let f be continuous on $[a,b]$ and n times differentiable on (a,b). If f is zero at the $n+1$ distinct points $x_0 < x_1 < \ldots < x_n$ in $[a,b]$, then there exists a number c in (a,b) such that $f^{(n)}(c)=0$.

Proof: The argument uses mathematical induction. If $n=1$ then we have the original Rolle's Theorem. To see how the induction argument works, consider the next case, $n=2$. Then $f(x)=0$ at three points $x_0 < x_1 < x_2$. Applying Rolle's Theorem on the interval $[x_0,x_1]$, there exists $c_1\in(x_0,x_1)$ such that $f'(c_1)=0$. Similarly, there exists $c_2\in(x_1,x_2)$ such that $f'(c_2)=0$. Now, using Rolle's Theorem again, $f'(c_1)=f'(c_2)=0$ implies that there exists $c\in(a,b)$ such that $f''(c)=0$. The full mathematical induction argument follows.

Theorem *Suppose $f \in C^n[a,b]$ and $x_0, x_1, \ldots, x_n$ are distinct numbers in $[a,b]$. Then there exists $\xi \in (a,b)$ such that*

$$f[x_0, x_1, \ldots, x_n] = \frac{f^{(n)}(\xi)}{n!}.$$

Sol: Let

$$P_n(x) = f[x_0] + \sum_{k=1}^{n} f[x_0, x_1, \ldots, x_k](x - x_0)(x - x_1)\cdots(x - x_{k-1})$$

be the interpolating polynomial of f in Newton's form.

Define $$g(x) = f(x) - P_n(x).$$

Since $P_n(x_i) = f(x_i)$ for $i = 0, 1, \ldots, n$, the function g has $n+1$ distinct zeros in $[a,b]$. By the generalized Rolle's Theorem, there exists $\xi \in (a,b)$ such that

$$g^{(n)}(\xi) = f^{(n)}(\xi) - P_n^{(n)}(\xi) = 0.$$

Note that

$$P_n^{(n)}(x) = n! f[x_0, x_1, \ldots, x_n].$$

As a consequence

$$f[x_0, x_1, \ldots, x_n] = \frac{f^{(n)}(\xi)}{n!}.$$

■

Newton's Method for Equi-Spaced Points

Newton's interpolatory divided-difference formula can be expressed in a simplified form when $x_0, x_1, \ldots, x_n$ are arranged consecutively with equal spacing. Let

$$h = \frac{x_n - x_0}{n} = x_{i+1} - x_i, \qquad i = 0, 1, \ldots, n-1.$$

Then each $x_i = x_0 + i * h$, $i = 0, 1, \ldots, n$.

For any $x \in [a, b]$, write

$$x = x_0 + s * h, \qquad s \in \mathbb{R}.$$

Hence $x - x_i = (s - i) * h$ and

$$\begin{aligned} P_n(x) &= f[x_0] + \sum_{k=1}^{n} f[x_0, x_1, \ldots, x_k](x - x_0)(x - x_1)\cdots(x - x_{k-1}) \\ &= f(x_0) + \sum_{k=1}^{n} f[x_0, x_1, \ldots, x_k](s-0) * h * (s-1) * h * \cdots (s-k+1) * h \\ &= f(x_0) + \sum_{k=1}^{n} f[x_0, x_1, \ldots, x_k] s(s-1)\cdots(s-k+1)h^k \\ &= f(x_0) + \sum_{k=1}^{n} f[x_0, x_1, \ldots, x_k] k! \binom{s}{k} h^k, \end{aligned}$$

where the binomial formula

$$\binom{s}{k} = \frac{s(s-1)\cdots(s-k+1)}{k!}$$

This formula is called the Newton forward divided-difference formula.

Next we introduce the forward difference notation $\triangle$

$$\triangle f(x_i) = f(x_{i+1}) - f(x_i)$$

and

$$\triangle^k f(x_i) = \triangle^{k-1} f(x_{i+1}) - \triangle^{k-1} f(x_i) = \triangle\left(\triangle^{k-1} f(x_i)\right),$$

for $i = 0, 1, \ldots, n-1$. With this notation,

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

and, in general

$$f[x_0, x_1, \ldots, x_k] = \frac{1}{k!\, h^k}\triangle^k f(x_0).$$

Using the forward difference $\triangle$ notation, the Newton's forward divided-difference formula can be expressed as

$$P_n(x) = f(x_0) + \sum_{k=1}^{n} \binom{s}{k} \triangle^k f(x_0).$$

This formula is called the Newton forward-difference formula.

Newton backward divided-difference formula.

If the interpolation nodes are arranged as $x_n, x_{n-1}, \ldots, x_0$, a formula for the interpolating polynomial similar to results

$$P_n(x) = f[x_n] + \sum_{k=1}^{n} f[x_n, x_{n-1}, \ldots, x_{n-k}](x - x_n)(x - x_{n-1}) \cdots (x - x_{n-k+1}).$$

If the nodes are equally spaced with

$$h = \frac{x_n - x_0}{n}, \qquad x_i = x_n - (n - i) * h, \qquad x = x_n + s * h,$$

then

$$P_n(x) = f[x_n] + \sum_{k=1}^{n} f[x_n, x_{n-1}, \ldots, x_{n-k}] s * h * (s+1) * h * \cdots * (s+k-1) * h$$

$$\begin{aligned} P_n(x) &= f[x_n] + \sum_{k=1}^{n} f[x_n, x_{n-1}, \ldots, x_{n-k}] s * h * (s+1) * h * \cdots * (s+k-1) * h \\ &= f(x_n) + \sum_{k=1}^{n} f[x_n, x_{n-1}, \ldots, x_{n-k}] (-1)^k k! \binom{-s}{k} h^k \end{aligned}$$

where the binomial formula is extended to include all real value s

$$\binom{-s}{k} = \frac{-s(-s-1)\cdots(-s-k+1)}{k!} = (-1)^k \frac{s(s+1)\cdots(s+k-1)}{k!}.$$

This form is called the Newton backward divided-difference formula.

introduce the backward-difference notation

$$\nabla^k f(x_i) = \nabla^{k-1} f(x_i) - \nabla^{k-1} f(x_{i-1}) = \nabla\left(\nabla^{k-1} f(x_i)\right),$$

then

$$f[x_n, x_{n-1}] = \frac{1}{h}\nabla f(x_n), \qquad f[x_n, x_{n-1}, x_{n-2}] = \frac{1}{2h^2}\nabla^2 f(x_n),$$

and, in general,

$$f[x_n, x_{n-1}, \ldots, x_{n-k}] = \frac{1}{k!\,h^k}\nabla^k f(x_n).$$

Using this backward-difference ∇ notation, the Newton backward divided-difference formula can be written as

$$P_n(x) = f(x_0) + \sum_{k=1}^{n} (-1)^k \binom{-s}{k} \nabla^k f(x_k).$$

Newton backward divided-difference formula.

Example 1:

Given the following data, estimate $f(1.83)$ using Newton-Gregory forward difference interpolation polynomial:

i	0	1	2	3	4
x_i	1.0	3.0	5.0	7.0	9.0
f_i	0	1.0986	1.6094	1.9459	2.1972

Solution:

Here we have five data points i.e $i = 0, 1, 2, 3, 4$. Let us first generate the forward difference table.

i	x_i	f_i	Δf_i	$\Delta^2 f_i$	$\Delta^3 f_i$	$\Delta^4 f_i$
0	1.0	.0				
			1.0986			
1	3.0	1.0986		-0.5878		
			0.5108		0.4135	
2	5.0	1.6094		-0.1743		−0.3244
			0.3365		0.0891	
3	7.0	1.9459		-0.0852		
			0.2513			
4	9.0	2.1972				

$$h = 2, \quad x = 1.83, \quad x_0 = 1.0, \quad s = \frac{x - x_0}{h} = 0.415$$

$\therefore$ Newton Gregory forward difference interpolation polynomial is given by:

$$P_n(s) = \sum_{k=0}^{n} \binom{s}{k} \Delta^k f_0$$

$$P_1(0.415) = \binom{0.415}{0} f_0 + \binom{0.415}{1} \Delta f_0$$
$$= 0 + (0.415)(1.0986) = 0.455919$$

$$P_2(0.415) = P_1(0.415) + \binom{0.415}{2} \Delta^2 f_0$$
$$= 0.455919 + \frac{0.415(0.415-1)}{2}(-0.5878)$$
$$= 0.455919 + 0.071352$$
$$= 0.527271$$

$$P_3(0.415) = P_2(0.415) + \binom{0.415}{3} \Delta^3 f_0$$
$$= 0.527271 + \frac{0.415(0.415-1)(0.415.2)}{6}(0.4135)$$
$$= 0.527271 + 0.026519$$
$$= 0.554157$$

$$P_4(0.415) = P_3(0.415) + \binom{0.415}{4} \Delta^4 f_0$$
$$= 0.554157 + \frac{0.415(0.415-1)(0.415-2)(0.415-3)}{24}(-0.3244)$$
$$= 0.554157 + 0.013445$$
$$= 0.567602$$

$$\therefore f(1.83) = 0.567602$$

Error Analysis for Polynomial Interpolation

Theorem *Suppose $f(x) \in C^{n+1}[a,b]$, and $P(x)$ is a polynomial of degree $\leq n$ that interpolates f at $n+1$ given distinct points $x_0, x_1 \ldots, x_n$ in $[a,b]$. Then for each $x \in [a,b]$, there exists a point $\xi_x \in (a,b)$ such that*

$$f(x) - P(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi_x) \prod_{k=0}^{n} (x - x_k).$$

Proof: Note first that if $x = x_i$ for some $0 \leq i \leq n$, then $P(x) = f(x)$, the theorem holds for any arbitrary $\xi_x \in (a,b)$. Suppose that $x \in [a,b]$ but $x \neq x_i$ for all $i = 0, 1, \ldots, n$.

Define a new function

$$g(t) = f(t) - P(t) - [f(x) - P(x)] \prod_{k=0}^{n} \frac{(t - x_k)}{(x - x_k)}, \qquad t \in [a,b].$$

Since $f \in C^{n+1}[a,b]$, $P \in C^{\infty}[a,b]$, and $x \neq x_i$, $\forall\ i$, it follows that $g(t)$ is well-defined for all $t \in [a,b]$, and $g \in C^{n+1}[a,b]$. Then

$$g(x_i) = f(x_i) - P(x_i) - [f(x) - P(x)] \prod_{k=0}^{n} \frac{(x_i - x_k)}{(x - x_k)} = 0, \qquad i = 0, 1, \ldots, n.$$

Moreover

$$g(x) = f(x) - P(x) - [f(x) - P(x)] \prod_{k=0}^{n} \frac{(x - x_k)}{(x - x_k)} = 0.$$

Thus $g \in C^{n+1}[a,b]$ and g has $n+2$ zeros in $[a,b]$.

By the generalized Rolle's theorem, there exists $\xi_x \in (a,b)$ for which $g^{(n+1)}(\xi_x) = 0$.

Thus $g \in C^{n+1}[a,b]$ and g has $n+2$ zeros in $[a,b]$. By the generalized Rolle's theorem, there exists $\xi_x \in (a,b)$ for which $g^{(n+1)}(\xi_x) = 0$. Hence

$$\begin{aligned} 0 = g^{(n+1)}(\xi_x) &= f^{(n+1)}(\xi_x) - P^{(n+1)}(\xi_x) - [f(x) - P(x)] \left[\frac{d^{n+1}}{dt^{n+1}} \prod_{k=0}^{n} \frac{(t - x_k)}{(x - x_k)} \right] \Bigg|_{t=\xi_x} \\ &= f^{(n+1)}(\xi_x) - 0 - [f(x) - P(x)] \cdot \frac{(n+1)!}{\prod (x - x_k)}. \end{aligned}$$

Therefore

$$f(x) - P(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi_x) \prod_{k=0}^{n} (x - x_k).$$

■