PART II

2 Interpolation

2.1 Problem Statement and Applications

Consider the following table:

x_0	f_0
x_1	f_1
x_2	f_2
$\vdots$	$\vdots$
x_k	f_k
$\vdots$	$\vdots$
x_n	f_n

In the above table, $f_k, k = 0, \cdots, n$ are assumed to be the values of a certain function $f(x)$, evaluated at $x_k, k = 0, \cdots, n$ in an interval containing these points. **Note that only the functional values are known, not the function $f(x)$ itself.** The problem is to find f_u corresponding to a nontabulated intermediate value $x = u$.

Such a problem is called an **Interpolation Problem**. The numbers $x_0,\ x_1, \cdots, x_n$ are called the **nodes**.

Interpolation Problem

Given $(n+1)$ points: $(x_0, f_0), (x_1, f_1), \cdots (x_n, f_n)$, find f_u corresponding to x_u, where $x_0 < x_u < x_n$; assuming that $f_0, f_1, \cdots, f_n$ are the values of a certain function $f(x)$ at $x = x_0, x_1, \cdots, x_n$, respectively.

The Interpolation problem is also a classical problem and dates back to the time of **Newton** and **Kepler**, who needed to solve such a problem in analyzing data on the positions of stars and planets. It is also of interest in numerous other practical applications. Here is an example.

2.2 Existence and Uniqueness

It is well-known that a continuous function $f(x)$ on $[a,b]$ can be approximated as close as possible by means of a polynomial. Specifically, for each $\epsilon > 0$, there exists a polynomial $P(x)$ such that $|f(x) - P(x)| < \epsilon$ for all x in $[a,b]$. This is a classical result, known as **Weierstrass Approximation Theorem**.

Knowing that $f_k, k = 0, \cdots, n$ are the values of a certain function at x_k, the most obvious thing then to do is to construct a polynomial $P_n(x)$ of degree at most n that passes through the $(n+1)$ points: $(x_0, f_0),\ (x_1, f_1),\ \cdots, (x_n, f_n)$.

Indeed, if the nodes $\boldsymbol{x_0, x_1, ..., x_n}$ are assumed to be distinct, then such a polynomial always does exist and is unique, as can be seen from the following.

Let $P_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$ be a polynomial of degree at most n. If $P_n(x)$ interpolates at $x_0, x_1, \cdots, x_n$, we must have, by definition

$$\begin{aligned}
&P_n(x_0) = f_0 = a_0 + a_1 x_0 + a_2 x_0^2 + \cdots + a_n x_0^n \\
&P_n(x_1) = f_1 = a_0 + a_1 x_1 + a_2 x_1^2 + \cdots + a_n x_1^n \\
&\vdots \\
&P_n(x_n) = f_n = a_0 + a_1 x_n + a_2 x_n^2 + \cdots + a_n x_2^n
\end{aligned} \tag{2.1}$$

These equations can be written in matrix form:

$$\begin{pmatrix} 1 & x_0 & x_0^2 \cdots x_0^n \\ 1 & x_1 & x_1^2 \cdots x_1^n \\ \vdots & & \\ 1 & x_n & x_n^2 \cdots x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{pmatrix}$$

Because $x_0,\ x_1, \cdots, x_n$ are distinct, it can be shown [**Exercise**] that the matrix of the above system is nonsingular. Thus, the linear system for the unknowns $a_0,\ a_1, \cdots, a_n$ has a unique solution, in view of the following well-known result, available in any linear algebra text book.

The $n \times n$ algebraic linear system $Ax = b$ has a unique solution for every b if and only if A is nonsingular.

This means that $P_n(x)$ exists and is unique.

Theorem 2.1 ***(Existence and Uniqueness Theorem for Polynomial Interpolation)***
Given $(n+1)$ distinct points $x_0,\ x_1, \cdots, x_n$ and the associated values $f_0,\ f_1, \cdots, f_n$ of a function $f(x)$ at these points (that is, $f(x_i) = f_i,\ i = 0, 1, \cdots, n$)), there is a **unique polynomial** *$P_n(x)$ of degree at most n such that $P_n(x_i) = f_i, i = 0, 1, \cdots, n$.*

The polynomial $P_n(x)$ in Theorem 3.1 is called the **interpolating polynomial**.

2.3 The Lagrange Interpolation

Once we know that the interpolating polynomial exists and is unique, the problem then becomes how to construct an interpolating polynomial; that is, how to construct a polynomial $P_n(x)$ of degree at most n, such that

$$P_n(x_i) = f_i,\ i = 0, 1, \cdots, n.$$

It is natural to obtain the polynomial by solving the linear system (3.1) in the previous section. Unfortunately, the matrix of this linear system, known as the **Vandermonde Matrix**, is usually **highly ill-conditioned**, and the **solution of such an ill-conditioned system, even by the use of a stable method, may not be accurate.** There are, however, several other ways to construct such a polynomial, that do not require solution of a Vandermonde system. We describe one such in the following:

Suppose $n = 1$, that is, suppose that we have only two points $(x_0, f_0),\ (x_1, f_1)$, then it is easy to see that the linear polynomial

$$P_1(x) = \frac{x - x_1}{(x_0 - x_1)} f_0 + \frac{(x - x_0)}{(x_1 - x_0)} f_1$$

is an interpolating polynomial, because

$$P_1(x_0) = f_0,\ P_1(x_1) = f_1.$$

For convenience, we shall write the polynomial $P_1(x)$ in the form

$$P_1(x) = L_0(x)f_0 + L_1(x)f_1,$$

where, $L_1(x) = \dfrac{x - x_0}{x_1 - x_0}$, and $L_1(x) = \dfrac{x - x_0}{x_1 - x_0}$.

Note that both the polynomials $L_0(x)$ and $L_1(x)$ are polynomials of degree 1.

The concept can be generalized easily for polynomials of higher degrees.

To generate polynomials of higher degrees, let's define the set of polynomials $\{L_k(x)\}$ recursively, as follows:

$$L_k(x) = \frac{(x - x_0)(x - x_1)\cdots(x - x_{k-1})(x - x_{k+1})\cdots(x - x_n)}{(x_k - x_0)(x_k - x_1)\cdots(x_k - x_{k-1})(x_k - x_{k+1})\cdots(x_k - x_n)},\ k = 0, 1, 2, \cdots, n. \tag{2.2}$$

We will now show that the polynomial $P_n(x)$ defined by

$$P_n(x) = L_0(x)f_0 + L_1(x)f_1 + \cdots + L_n(x)f_n \tag{2.3}$$

is an interpolating polynomial.

To see this, note that

$$\begin{aligned}
L_0(x) &= \frac{(x - x_1)\cdots(x - x_n)}{(x_0 - x_1)\cdots(x_0 - x_n)} \\
L_1(x) &= \frac{(x - x_0)(x - x_2)\cdots(x - x_n)}{(x_1 - x_0)(x_1 - x_2)\cdots(x_1 - x_n)} \\
&\vdots \\
L_n(x) &= \frac{(x - x_0)(x - x_1)(x - x_2)\cdots(x - x_{n-1})}{(x_n - x_0)(x_n - x_1)(x_n - x_2)\cdots(x_n - x_{n-1})}
\end{aligned}$$

Also, note that

$$L_0(x_0) = 1,\ L_0(x_1) = L_0(x_2) = \cdots = L_0(x_n) = 0$$

$$L_1(x_1) = 1,\ L_1(x_0) = L_1(x_2) = \cdots = L_n(x_n) = 0$$

In general

$L_k(x_k) = 1$ and $L_k(x_i) = 0,\ i \neq k.$

Thus

$$
\begin{aligned}
P_n(x_0) &= L_0(x_0)f_0 + L_1(x_0)f_1 + \cdots + L_n(x_0)f_n = f_0 \\
P_n(x_1) &= L_0(x_1)f_0 + L_1(x_1)f_1 + \cdots + L_n(x_1)f_n = 0 + f_1 + \cdots + 0 = f_1 \\
&\vdots \\
P_n(x_n) &= L_0(x_n)f_0 + L_1(x_n)f_1 + \cdots + L_n(x_n)f_n = 0 + 0 + \cdots + 0 + f_n = f_n
\end{aligned}
$$

That is, the polynomial $P_n(x)$ has the property that $P_n(x_k) = f_k,\ k = 0, 1, \cdots, n$.

The polynomial $P_n(x)$ defined by (3.3) is known as the **Lagrange Interpolating Polynomial**.

Example 2.1 *Interpolate $f(x)$ from the following table:*

0	7
1	13
2	21
4	43

and find and approximation to $f(3)$.

$$
\begin{aligned}
L_0(x) &= \frac{(x-1)(x-2)(x-4)}{(-1)(-2)(-4)} \\
L_1(x) &= \frac{(x-0)(x-2)(x-4)}{1\cdot(-1)(-3)} \\
L_2(x) &= \frac{(x-0)(x-1)(x-4)}{2\cdot 1\cdot(-2)} \\
L_3(x) &= \frac{(x-0)(x-1)(x-2)}{4\cdot 3\cdot 2}
\end{aligned}
$$

$P_3(x) = 7L_0(x) + 13L_1(x) + 21L_2(x) + 43L_3(x)$

Thus, $P_3(3) = 7L_0(3) + 13L_1(3) + 21L_2(3) + 43L_3(3) = 31.$

Now, $L_0(3) = \frac{1}{4},\ L_1(3) = -1,\ L_2(3) = \frac{3}{2},\ L_3(3) = \frac{1}{4}.$

So, $P_3(3) = 7L_0(3) + 13L_1(3) + 21L_2(3) + 43L_3(3) = 31.$

Verify: Note that $f(x)$ in this case is $f(x) = x^2 + 5x + 7$, and the exact value of $f(x)$ at $x = 3$ is 31.

Example 2.2 *Given*

i	x_i	$f(x_i)$
0	2	$\frac{1}{2}$
1	2.5	$\frac{1}{2.5}$
2	4	$\frac{1}{4}$

We want to find $f(3)$.

$$L_0(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} = \frac{(x-2.5)(x-4)}{(-0.5)(-2)} = (x-2.5)(x-4)$$

$$L_1(x) = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} = \frac{(x-2)(x-4)}{(0.5)(-1.5)} = -\frac{1}{0.75}(x-2)(x-4)$$

$$L_2(x) = \frac{(x-x_1)(x-x_0)}{(x_2-x_1)(x_2-x_0)} = \frac{1}{3}(x-2.5)(x-2)$$

$$P_2(x) = f(x_0)L_0(x) + f(x_1)L_1(x) + f(x_2)L_2(x) = \frac{1}{2}L_0(x) + \frac{1}{2.5}L_1(x) + \frac{1}{4}L_2(x)$$

$$P_2(3) = \frac{1}{2}L_0(3) + \frac{1}{2.5}L_1(3) + \frac{1}{4}L_2(3) = \frac{1}{2}(-0.5) + \frac{1}{2.5}\left(\frac{1}{0.75}\right) + \frac{1}{4}\left(\frac{.5}{3}\right) = 0.3250$$

Verify: (The value of $f(x)$ at $x = 3$ is $\frac{1}{3} = 0.3333$).

2.4 Error in Interpolation

If $f(x)$ is approximated by an interpolating polynomial $P_n(x)$, we would like to obtain an expression for the error of interpolation at a give intermediate point, say, $\bar{x}$.

That is, we would like to calculate $E(\bar{x}) = f(\bar{x}) - P_n(\bar{x})$.

Note that, since $P_n(x_i) = f(x_i),\ E(x_i) = 0,\ i = 0, 1, 2, \cdots, n)$, that is, **there are no errors of interpolating at a tabulated point.**

Here is a result for the expression of $E(\bar{x})$.

Theorem 2.2 ***(Interpolation-Error Theorem)*** *Let $P_n(x)$ be the interpolating polynomial that interpolates at $(n+1)$ distinct numbers in $[a, b]$, and let $f(x)$ be $(n+1)$ times continuously differentiable on $[a, b]$. Then for every $\bar{x}$ in $[a, b]$, there exists a number $\xi = \xi(\bar{x})$ (depending on $\bar{x}$) such that*

$$E_n(\bar{x}) = f(\bar{x}) - P_n(\bar{x}) = \frac{f^{(n+1)}(\xi(\bar{x}))}{(n+1)!} \prod_{i=0}^{n} (\bar{x} - x_i). \tag{2.4}$$

Proof: If $\bar{x}$ is one of the numbers $x_0, x_1, \cdots, x_n$: then the result follows trivially. Because, the error in this case is zero, and the result will hold for any arbitrary ξ.

Next, assume that $\bar{x}$ is not one of the numbers $x_0, x_1, \cdots, x_n$.

Define a function $g(t)$ in variable t in $[a, b]$:

$$g(t) = f(t) - P_n(t) - [f(\bar{x}) - P_n(\bar{x})] * \left[\frac{(t-x_0)(t-x_1)\cdots(t-x_n)}{(\bar{x}-x_0)(\bar{x}-x_1)\cdots(\bar{x}-x_n)}\right]. \tag{2.5}$$

Then, noting that $f(x_k) = P_n(x)$, for $k = 0, 1, 2, \cdots, n$, we have

$$\begin{aligned} g(x_k) &= f(x_k) - P_n(x_k) - [f(\bar{x}) - P_n(\bar{x})] \left[\frac{(x_k-x_0)\cdots(x_k-x_n)}{(\bar{x}-x_0)\cdots(\bar{x}-x_n)}\right] \\ &= P_n(x_k) - P_n(x_k) - [f(\bar{x}) - P_n(\bar{x})] \ \times 0 \\ &= 0 \end{aligned} \tag{2.6}$$

(Note that the numerator of the fraction appearing above contains the term $(x_k - x_k) = 0$).

Furthermore,

$$\begin{aligned} g(\bar{x}) &= f(\bar{x}) - P_n(\bar{x}) - [f(\bar{x}) - P_n(\bar{x})] * \left[\frac{(\bar{x}-x_0)\cdots(\bar{x}-x_n)}{(\bar{x}-x_0)\cdots(\bar{x}-x_n)}\right] \\ &= f(\bar{x}) - P_n(\bar{x}) - f(\bar{x}) + P_n(\bar{x}) \\ &= 0 \end{aligned} \tag{2.7}$$

Thus, $g(t)$ becomes identically zero at $(n+2)$ distinct points: $x_0, x_1, \cdots, x_n$, and $\bar{x}$. Furthermore, $g(t)$ is $(n+1)$ times continuously differentiable, since $f(x)$ is so.

Therefore, by **generalized Rolle's theorem**, [ ], there exists a number $\xi(\bar{x})$ in (a, b) such that $g^{(n+1)}(\xi) = 0$.

Let's compute $g^{(n+1)}(t)$ now. ¿From (2.5) we have

$$g^{(n+1)}(t) = f^{(n+1)}(t) - P_n^{(n+1)}(t) - [f(\bar{x}) - P_n(\bar{x})] \frac{d^{n+1}}{d^{n+1}} \left[\frac{(t-x_0)(t-x_1)\cdots(t-x_n)}{((\bar{x}-x_0)(\bar{x}-x_1)\cdots(\bar{x}x_n)}\right]$$

Then

$$\begin{aligned} \frac{d^{n+1}}{dt^{n+1}} & \left[\frac{(t-x_0)(t-x_1)\cdots(t-x_n)}{(\bar{x}-x_0)(\bar{x}-x_1)\cdots(\bar{x}x_n)}\right] \\ &= \frac{1}{(\bar{x}-x_0)(\bar{x}-x_1)\cdots(\bar{x}-x_n)} \cdot \frac{d^{n+1}}{dt^{n+1}}[(t-x_0)(t-x_1)\cdots(t-x_n)] \\ &= \frac{1}{(\bar{x}-x_0)(\bar{x}-x_1)\cdots(\bar{x}-x_n)}(n+1)! \end{aligned}$$

(note that the expression within [ ] is a polynomial of degree $n+1$).

Also, $P_n^{(n+1)}(t) = 0$, because P_n is a polynomial of degree at most n. Thus, $P_n^{(n+1)}(\xi) = 0$.

So,

$$g^{(n+1)}(\xi) = f^{(n+1)}(\xi) - \frac{(f(\bar{x}) - P_n(\bar{x}))}{(\bar{x} - x_0)\cdots(\bar{x} - x_n)}(n+1)! \tag{2.8}$$

Since $g^{(n+1)}(\xi) = 0$, from (2.8), we have $E_n(\bar{x}) = f(\bar{x}) - P_n(\bar{x}) = \dfrac{f^{(n+1)}(\xi)}{(n+1)!}(\bar{x} - x_0)\cdots(\bar{x} - x_n)$.

Remark: To obtain the error of interpolation using the above theorem, we need to know the $(n+1)$th derivative of the $f(x)$ or its absolute maximum value on the interval $[a,b]$. Since in practice this value is hardly known, this error formula is of limited use only.

Example 2.3 *Let's compute the maximum absolute error for Example 3.2.*

Here $n = 2$.

$$\begin{aligned} E_2(\bar{x}) &= f(\bar{x}) - P_2(\bar{x}) \\ &= \frac{f^{(3)}(\xi)}{3!}(\bar{x} - x_0)(\bar{x} - x_1)(\bar{x} - x_2) \end{aligned}$$

To know the maximum value of $E_2(\bar{x})$, *we need to know* $f^{(3)}(x)$.

Let's compute this now:

$$f(x) = \frac{1}{x}, \quad f'(x) = -\frac{1}{x^2}, \quad f''(x)\frac{2}{x^3}, \quad f^{(3)}(x) = -\frac{6}{x^4}.$$

So, $|f^{(3)}(\xi)| < \dfrac{6}{2^4} = \dfrac{6}{16}$ *for* $0 < x \le 2$.

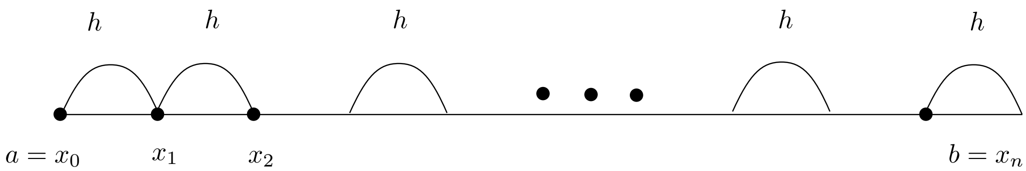
Since $\bar{x} = 3,\ x_0 = 2,\ x_1 = 2.5,\ x_2 = 4,$ *we have*

$$|E_2(\bar{x})| \le |\ \frac{6}{16} \times \frac{1}{6}\ (3-2)(3-2.5)(3-4)| = 0.0625.$$

Note that in four-digit arithmetic, the difference between the value obtained by interpolation and the exact value is $0.3333 - 0.3250 = 0.0083$.

2.5 Simplification of the Error Bound for Equidistant Nodes

The error formula in Theorem 3.2 can be simplified in case the tabulated points (nodes) are equally spaced; because, in this case it is possible to obtain a nice bound for the expression: $(\bar{x}-x_0)(\bar{x}-x_1)\cdots(\bar{x}-x_n)$. Suppose the nodes are equally spaced with spacing h; that is $x_{i+1}-x_i = h$.



Then it can be shown **[Exercise]** that

$$|(\bar{x}-x_0)(\bar{x}-x_1)\cdots(\bar{x}-x_n)| \leq \frac{h^{n+1}}{4}n!$$

If we also assume that $|f^{(n+1)}(x)| \leq M$, then we have

$$|E_n(\bar{x})| = |f(\bar{x}) - P_n(\bar{x})| \leq \frac{M}{(n+1)!}\ \frac{h^{n+1}}{4}\ n! = \frac{Mh^{n+1}}{4(n+1)}. \qquad (2.9)$$

Example 2.4 *Suppose a table of values for $f(x) = \cos x$ has to be prepared in $[0, 2\pi]$ with equal spacing nodes of spacing h, using* **linear interpolation**, *with an error of interpolation of at most 5×10^{-8}. How small should h be?*

Here $n = 1$.

$f(x) = \cos x,\ f'(x) = -\sin x,\ f^2(x) = f''(x) = -\cos x$

$\max|f^{(2)}(x)| = 1,$ *for* $0 \leq x \leq 2\pi$

Thus $M = 1$.

So, by (3.9) above we have

$$|E_1(\bar{x})| = |f(\bar{x}) - P_1(\bar{x})| \leq \frac{h^2}{8}.$$

Since the maximum error has to be 5×10^{-7}, we must have:
$\frac{h^2}{8} \leq 5 \times 10^{-7} = \frac{1}{2} \times 10^{-6}$. That is, $h \ \leq 6.3246 \times 10^{-4}$.

Example 2.5 *Suppose a table is to be prepared for the function $f(x) = \sqrt{x}$ on $[1, 2]$. Determine the spaceing h in a table such that the interpolation with a polynomial of degree 2 will give accuracy $\epsilon = 5 \times 10^{-8}$.*

We first compute the maximum absolute error.

Since $f^{(3)}(x) = \frac{3}{8} x \frac{-5}{2}$,

$$M = \left| f^{(3)}(x) \right| \leq \frac{3}{8} \quad \textit{for} \ \ 1 \leq x \leq 2.$$

Thus, taking $n = 2$ in (2.9) the maximum (absolute) error is $\frac{3}{8} \times \frac{h^3}{4x^3} = \frac{1}{32} h^3$.

Thus, to have an accuracy of $\epsilon = 5 \times 10^{-8}$, we must have $\frac{1}{32} h^3 < 5 \times 10^{-8}$ or $h^3 < 160 \times 10^{-8}$.

This means that a spacing h of about $h = \sqrt[3]{160 \times 10^{-8}} = 0.0117$ will be needed in the Table to guarantee the accuracy of 5×10^{-8}.

2.6 Divided Differences and the Newton-Interpolation Formula

A major difficulty with the Lagrange Interpolation is that one is not sure about the degree of interpolating polynomial needed to achieve a certain accuracy. Thus, if the accuracy is not good enough with polynomial of a certain degree, one needs to increase the degree of polynomial, and **computations need to be started all over again.**

Furthermore, computing various Lagrangian polynomials is an expensive procedure. **It is, indeed, desirable to have a formula which makes use of $P_{k-1}(x)$ in computing $P_k(x)$.**

The following form of interpolation, known as **Newton's interpolation** allows us to do so.

The idea is to obtain the interpolating polynomial $P_n(x)$ in the following form:

$$P_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_n(x - x_0)(x - x_1) \cdots (x - x_{n-1}) \tag{2.10}$$

The constants a_0 through a_n can be determined as follows:

For $x = x_0$, $\quad P_n(x_0) = a_0 = f_0$

For $x = x_1$, $\quad P_n(x_1) = a_0 + a_1(x_1 - x_0) = f_1$, which gives

$$a_1 = \frac{f_1 - a_0}{x_1 - x_0} = \frac{f_1 - f_0}{x_1 - x_0}.$$

The other numbers $a_i, i = 2, ..., n$ can similarly be obtained.

It is convenient to introduce the following notation, because we will show how the numbers $a_0, ..., a_n$ can be obtained using these notations.

$$f(x_i) = f[x_i] \text{ and } f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}$$

Similarly,

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

With these notations, we then have

$$a_0 = f_0 = f(x_0) = f[x_0]$$

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

Continuing this, it can be shown that **[Exercise]**

$$a_k = f[x_0, x_1, \cdots, x_k]. \tag{2.11}$$

The number $f[x_0, x_1, \cdots, x_k]$ is called the k-th **divided difference**.

Substituting these expressions of a_k in (2.11), the interpolating polynomial $P_n(x)$ now can be written in terms of the **divided differences:**

$$\begin{aligned} P_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \cdots + \\ f[x_0, x_1, \cdots, x_n](x - x_0)(x - x_1) \cdots (x - x_{n-1}). \end{aligned} \tag{2.12}$$

Notes:

(i) Each divided difference can be obtained from two previous ones of lower orders.

For example, $f[x_0, x_1, x_2]$ can be computed from $f[x_0, x_1]$, and $f[x_1, x_2]$, and so on. Indeed, they can be arranged in form of a table as shown below:

Table of Divided Differences ($n = 4$)

x	$f(x)$	1^{st} Divide Difference	2^{nd} Divided Difference	3^{rd} Divided Difference
x_0	f_0			
x_1	f_1	$f[x_0, x_1]$	$f[x_0, x_1, x_2]$	
x_2	f_2	$f[x_1, x_2]$	$f[x_1, x_2, x_3]$	$f[x_0, x_1, x_2, x_3]$
x_3	f_3	$f[x_2, x_3]$	$f[x_2, x_3, x_4]$	$f[x_1, x_2, x_3, x_4]$
x_4	f_4	$f[x_3, x_4]$		

(ii) Note that in computing $P_n(x)$ we need only the diagonal entries of the above table; that is, we need only $f[x_0], f[x_0, x_1], \cdots, f[x_0, x_1, \cdots, x_n]$.

(iii) Since the divided differences are generated recursively, the interpolating polynomials of successively higher degrees can also be generated recursively. **Thus the work done previously can be used gainfully.**

For example,

$P_1(x) = f[x_0] + f[x_1, x_0](x - x_0)$

$$\begin{aligned} P_2(x) &= f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\ &= P_1(x) + f[x_0, x_1, x_2](x - x_0)(x - x_1). \end{aligned}$$

Similarly, $P_3(x) = P_2(x) + f[x_0, x_1, x_2, x_3](x - x_0)(x - x_1)(x - x_2)$

$$P_4(x) = P_3(x) + f[x_0, x_1, x_2, x_3, x_4](x - x_0)(x - x_1)(x - x_2)(x - x_3)$$

and so on.

Thus, in computing $P_2(x)$, $P_1(x)$ has been gainfully used; in computing $P_3(x)$, $P_2(x)$ has been gainfully used, etc.

Example 2.6 *Interpolate at $x = 2.5$ using the following Table, with polynomials of degree 3 and 4.*

n	x	f	1st diff.	2nd diff.	3rd diff.	4th diff	5th diff
0	1.0	0	0.35218	-0.1023	.0265533	$-.006408$	.001412
1	1.5	0.17609	0.24988	-0.0491933	.0105333	$-.002172$	
2	2.0	0.30103	0.17609	-0.0281267	.0051033		
3	3.0	0.47712	0.1339	-0.01792			
4	3.5	0.54407	0.11598				
5	4.0	0.60206					

¿From (2.12), with $n = 3$, we have

$$\begin{aligned} P_3(2.5) &= 0 + (2.5-1.0)(.35218) + (2.5-1.0)(2.5-1.5)(-.1023) \\ &\quad +(2.5-1.0)(2.5-1.5)(2.5-2.0)(.0265533) \\ &= .52827 - .15345 + .019915 \\ &= \boxed{0.394735} \end{aligned}$$

Similarly, with $n = 4$, we have

$$\begin{aligned} P_4(2.5) &= P_3(2.5) + (2.5-1.0)(2.5-1.5)(2.5-2.0)(2.5-3.0)(-.006408) \\ &= .394735 + .002403 = \boxed{0.397138}. \end{aligned}$$

Note that $P_4(2.5) - P_3(2.5) = \boxed{0.002403}$.

Verification. The above is a table for $\log(x)$.

The exact value of $\log(2.5)$ (correct up to 5 decimal places) is 0.39794.

(Note that in computing $P_4(2.5)$, $P_3(2.5)$ computed previously has been gainfully used).

Algorithm 2.1 ***Algorithm for Generating Divided Differences***

Inputs:

The definite numbers $x_0, x_1, \cdots, x_n$ and the values $f_0, f_1, \cdots, f_n$.

Outputs:

The Divided Differenced $D_{00}, D_{11}, \cdots D_{nn}$.

Step 1: (Initialization). Set

$d_{i,0} = 0 = f_i,\ i = 0, 1, 2, \cdots, n.$

For $i = 1, 2, \cdots, n$ do

$i = 1, 2, \cdots i$ do

$D_{ij} - \dfrac{D_{i,j-1} - D_{i-1,j-1}}{x_i - x_{i-j}}$

End

A Relationship Between nth Divided Difference and the n^{th} Derivative

The following theorem shows how the nth derivative of a function $f(x)$ is related to the nth divided difference. The proof is omitted. It can be found in any advanced numerical analysis text book (e.g., Atkins on (1978), p. 144).

Theorem 2.3 *Suppose f is n times continuously differentiable and $x_0, x_1, \cdots, x_n$ are $(n+1)$ distinct numbers in $[a,b]$. Then there exists a number ξ in (a,b) such that*

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

The Newton Interpolation with Equally Spaced Nodes

Suppose again that $x_0, \cdots, x_n$ are equally spaced with spacing h;

that is $x_{i+1} - x_i = h,\ i = 0, 1, 2, ..., n-1$. Let $x - x_0 = sh$.

Then $x - x_0 = sh$

$$x - x_1 = x - x_0 + x_0 - x_1 = (x - x_0) - (x_1 - x_0) = sh - h = (s-1)h$$

In general, $x - x_i = (s - i)h$.

So,

$$
\begin{aligned}
P_n(x) \quad &= P_n(x_0+sh) = f[x_0] + f[x_0,x_1](x-x_0) + \cdots + f[x_0,x_1,\cdots,x_n](x-x_0)\cdots(x-x_{n-1}) \\
&= f[x_0] + sh\, f[x_0,x_1] + s(s-1)h^2 f[x_0,x_1,x_2] + \cdots + s(s-1)\cdots(s-h+1)h^n f[x_0,\cdots,x_h] \\
&\sum_{k=0}^{n} s(s-1)\cdots(s-k+1)h^k f[x_0,\cdots,x_k].
\end{aligned} \tag{2.13}
$$

Invoke now the notation:

$$
\left(\begin{array}{c} s \\ k \end{array}\right) = \frac{(s)(s-1)\cdots(s-k+1)}{k!}
$$

We can then write

$$
P_n(x) = P_n(x_0+sh) = \sum_{k=0}^{n} \left(\begin{array}{c} s \\ k \end{array}\right) h^k k! f[x_0,\cdots,x_n] \tag{2.14}
$$

The Newton Forward-Difference Formula

Let's introduce the notations:

$\Delta f_i = f(x_{i+1}) - f(x_i)$.

Then, $\Delta f_0 = f(x_1) - f(x_0)$

So,

$$
f[x_0,x_1] = \frac{f(x_1)-f(x_0)}{x_1-x_0} = \frac{\Delta f_0}{h}. \tag{2.15}
$$

Also, note that

$$
\begin{aligned}
\Delta^2 f_0 \quad &= \Delta(\Delta f_0) = \Delta(f(x_1) - f(x_0)) \\
&= \Delta f(x_1) - \Delta f(x_0) \\
&= f(x_2) - f(x_1) - f(x_1) + f(x_0) \\
&= f(x_2) - 2f(x_1) + f(x_0)
\end{aligned} \tag{2.16}
$$

So,

$$
\begin{aligned}
f[x_0,x_1,x_2] \quad &= \frac{f[x_1,x_2] - f[x_0,x_1]}{(x_2-x_0)} \\
&= \frac{\frac{f(x_2)-f(x_1)}{x_2-x_1} - \frac{f(x_1)-f(x_0)}{x_1-x_0}}{(x_2-x_0)} \\
&= \frac{f(x_2) - 2f(x_1) - f(x_0)}{h \times 2h} = \frac{\Delta^2 f_0}{2h^2}
\end{aligned} \tag{2.17}
$$

In general, we have

$$f[x_0, x_1, \ldots, x_k] \\ = \frac{1}{k!h^k}\Delta^k f_0. \tag{2.18}$$

Proof is by induction on k.

For $k = 0$, the result is trivially true. We have also proved the result for $k = 1$, and $k = 2$.

Assume now that the result is true $k = m$. Then we need to show that the result is also true for $k = m + 1$.

Now, $\quad f[x_0, \cdots, x_{m+1}] = f[x_1, \cdots, x_{m+1}] - \dfrac{f[x_0, \cdots, x_m]}{x_{m+1} - x_0}$

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

The numbers $\Delta^k f_i$ are called k^{th} order forward differences of f at $x = i$.

We now show how the interpolating polynomial $P_n(x)$ given by (2.14) can be written using forward differences.

$$\begin{aligned} P_n(x) = P_n(x_0 + sh) \quad &= f[x_0] + shf[x_0, x_1] + s(s-1)h^2 f[x_0, x_1, x_2] \\ &+ \cdots + s(s-1)\cdots(s-n+1)h^n f[x_0, x_1 \cdot \cdots, x_n] \\ &= \sum_{k=0}^{n} s(s-1)\cdots(s-k+1)h^k f[x_0, x_1, \cdots, x_k] \\ &= \sum_{k=0}^{n} \begin{pmatrix} s \\ k \end{pmatrix} k!h^k f[x_0, x_1, \cdots, x_n] = \sum_{k=0}^{n} \binom{s}{k} \Delta^k f_0 \text{ using (2.18).} \end{aligned}$$

Newton's Forward-Difference Interpolation Formula

Let $x_0, x_1, ..., x_n$ be $(n+1)$ equidistant points with distance h; that is $x_{i+1} - x_i = h$. Then Newton's interpolating polynomial $P_n(x)$ of degree at most n, using forward-differences $\Delta^k f_0,\ k = 0, 1, 2, ..., n$ is given by

$$P_n(x) = \sum_{k=0}^{n} \begin{pmatrix} s \\ k \end{pmatrix} \Delta^k f_0,$$

where $s = \dfrac{x - x_0}{h}$.

Illustration: The Forward Difference Table with $n = 4$

x	$f(x)$	Δf	$\Delta^2 f$	$\Delta^3 f$	$\Delta^4 f$
x_0	f_0				
x_1	f_1	Δf_0			
x_2	f_2	Δf_1	$\Delta^2 f_0$		
x_3	f_3	Δf_2	$\Delta^2 f_1$	$\Delta^3 f_0$	
x_4	f_4	Δf_3	$\Delta^2 f_2$	$\Delta^3 f_1$	$\Delta^3 f_0$

Example 2.7 *Let $f(x) = e^x$.*

x	f	Δf	$\Delta^2 f$	$\Delta^3 f$	
0	1				
1	2.7183	1.7183			
2	7.3891	4.6709	2.8817	5.2147	
3	20.0855	12.6964	8.0964	13.7199	8.5052
		34.5127	21.8163		
4	54.5982				

Let $x = 1.5$

Then $s = \dfrac{x - x_0}{h} = \dfrac{1.5 - 0}{1} = 1.5$

$$P_4(1.5) = 1.7183 \times 1 + (1.5)(1.7183) + (1.5)\frac{(1.5-1)(1.5-2)}{2}$$
$$\times 2.8817 + (1.5)\frac{(1.5-1)(1.5-2)(1.5-3)}{6} \times 5.0734 = 4.0852.$$

The correct answer up to 4 *decimal digits is* 4.4817.

Interpolation using Newton-Backward Differences

While interpolating at some value of x near the end of the difference table, it is logical to reorder the nodes so that the end-differences can be used in computation. The backward differences allow us to do so.

The backward differences are defined by

$\nabla f_n = f_n - f_{n-1}, n = 1, 2, 3, \ldots,$

and

$\nabla^k f_n = \nabla(\nabla^{k-1} f_n), k = 2, 3, \ldots$

Thus, $\nabla^2 f_n = \nabla(\nabla f_n) = \nabla(f_n - f_{n-1})$

$$= f_n - f_{n-1} - (f_{n-1} - f_{n-2})$$

$$= f_n - 2f_{n-1} + f_{n-2},$$

and so on.

The following a relationship between the backward-differences and the divided differences can be obtained.

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

Using these backward-differences, we can write the Newton interpolation formula as:

$$P_n(x) = f_n + s\nabla f_n + \frac{s(s+1)}{2!}\nabla^2 f_n + ... + \frac{s(s+1)...(s+h-1)}{n!}\nabla^n f_n.$$

Again, using the notation $\binom{-s}{k} = \dfrac{(-s)(-s-1)\cdots(-s-k+1)}{k!}$

we can rewrite the above formula as:

Newton's Backward-Difference
Interpolations Formula

$$P_n(x) = \sum_{k=0}^{n} (-1)^k \binom{-s}{k} \bigtriangledown^k f_n.$$