Chapter 3

Interpolation and Polynomial Approximation

3.1 Interpolation and the Lagrange Polynomial

One of the most useful and well-known classes of functions mapping the set of real numbers into itself is the algebraic polynomials, the set of functions of the form

\[ P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0, \]

where \( n \) is a nonnegative integer and \( a_0, \ldots, a_n \) are real constants. One reason for their importance is that they uniformly approximate continuous functions. This result is expressed precisely in the Weierstrass Approximation Theorem.

**Theorem 3.1.1** (Weierstrass Approximation Theorem). Suppose that \( f \in C[a, b] \). For each \( \varepsilon > 0 \), there exists a polynomial \( P(x) \) such that

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

Another important reason for considering the class of polynomials in the approximation of functions is that the derivative 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.

**Lagrange Interpolating Polynomials**

The problem of determining a polynomial of degree one that passes through the distinct points \((x_0, y_0)\) and \((x_1, y_1)\) is the same as approximating a function for which \( f(x_0) = y_0 \) and \( f(x_1) = y_1 \) by means of a first-degree polynomial interpolating, or agreeing with the values of \( f \) at the given points. Using this polynomial for approximation within the interval given by the endpoints is called polynomial interpolation.

Define the functions

\[ L_0(x) = \frac{x - x_1}{x_0 - x_1}, \quad \text{and} \quad L_1(x) = \frac{x - x_0}{x_1 - x_0}. \]

The linear Lagrange interpolating polynomial through \((x_0, y_0)\) and \((x_1, y_1)\) is

\[ P(x) = L_0(x)f(x_0) + L_1(x)f(x_1). \]
Note that \( L_0(x_0) = 1, \) \( L_0(x_1) = 0, \) \( L_1(x_0) = 0, \) and \( L_1(x_1) = 1, \) which implies that
\[
P(x_0) = f(x_0) = y_0 \quad \text{and} \quad P(x_1) = f(x_1) = y_1.
\]
So \( P \) is the unique polynomial of degree at most one that passes through \((x_0, y_0)\) and \((x_1, y_1)\).

**Example 3.1.2.** We determine the linear Lagrange interpolating polynomial that passes through the points \((2, 4)\) and \((5, 1)\). In this case, we have
\[
L_0(x) = \frac{x - 5}{2 - 5} = -\frac{1}{3}(x - 5) \quad \text{and} \quad L_1(x) = \frac{x - 2}{5 - 2} = \frac{1}{3}(x - 2).
\]
Hence, we have
\[
P(x) = -\frac{1}{3}(x - 5) \cdot 4 + \frac{1}{3}(x - 2) \cdot 1 = -\frac{4}{3}x + \frac{20}{3} + \frac{1}{3}x - \frac{2}{3} = -x + 6.
\]
To generalize the concept of interpolation, consider the construction of a polynomial of degree at most \( n \) that passes through the \( n + 1 \) points \((x_i, f(x_i))\) for \( i = 0, \cdots, n \). In this case, we first construct, for each \( i = 0, 1, \cdots, n \), a function \( L_{n,k}(x) \) with the property that
\[
L_{n,k}(x_i) = 0 \quad \text{when} \quad i \neq k \quad \text{and} \quad L_{n,k}(x_k) = 1.
\]
To satisfy the first type of condition, it requires that the numerator of \( L_{n,k}(x) \) contains the term
\[
(x - x_0)(x - x_1) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_n).
\]
To satisfy \( L_{n,k}(x_k) = 1 \), the denominator of \( L_{n,k}(x) \) must be the same term but evaluated at \( x = x_k \). Thus,
\[
L_{n,k}(x) = \frac{(x - x_0)(x - x_1) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_n)}{(x_k - x_0)(x_k - x_1) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_n)}.
\]
The interpolating polynomial is easily described once the form of \( L_{n,k} \) is known. This polynomial, called \( n \)-th Lagrange interpolating polynomial, is defined in the following theorem.

**Theorem 3.1.3** (Lagrange Polynomial). If \( x_0, x_1, \cdots, x_n \) are \( n + 1 \) distinct numbers and \( f \) is a function whose values are given at these points, then a unique polynomial \( P(x) \) of degree at most \( n \) exists with
\[
f(x_k) = P(x_k) \quad \text{for} \quad k = 0, \cdots, n.
\]
The polynomial \( P(x) \) is given by
\[
P(x) = \sum_{k=0}^{n} f(x_k)L_{n,k}(x)
\]  \hspace{1cm} (3.1)
where
\[
L^{(n)}_k(x) = \frac{(x - x_0)(x - x_1) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_n)}{(x_k - x_0)(x_k - x_1) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_n)} = \prod_{i=0, i\neq k}^{n} \frac{x - x_i}{x_k - x_i}.
\]
We will write \( L_{n,k} \) simply as \( L_k \) when there is no confusion as to its degree.

**Example 3.1.4.** (a) Use the numbers (called nodes) \( x_0 = 2, x_1 = 2.75, \) and \( x_2 = 4 \) to find the second Lagrange interpolating polynomial for \( f(x) = 1/x \).
3.1. INTERPOLATION AND THE LAGRANGE POLYNOMIAL

(b) Use this polynomial to approximate \( f(3) = 1/3 \).

**Solution.** We first determine the coefficient polynomials \( L_0, L_1, \) and \( L_2 \). In nested form they are

\[
L_0(x) = \frac{(x - 2.75)(x - 4)}{(2 - 2.75)(2 - 4)} = \frac{2}{3}(x - 2.75)(x - 4),
\]

\[
L_1(x) = \frac{(x - 2)(x - 4)}{(2.5 - 2)(2.75 - 4)} = \frac{16}{15}(x - 2)(x - 4),
\]

and

\[
L_2(x) = \frac{(x - 2)(x - 2.75)}{(4 - 2)(4 - 2.75)} = \frac{2}{5}(x - 2)(x - 2.75).
\]

Also, \( f(x_0) = 1/2, f(x_1) = 4/11, f(x_2) = 1/4, \) so

\[
P(x) = \sum_{k=0}^{2} f(x_k)L_k(x) = \frac{1}{22}x^2 - \frac{35}{88}x + \frac{49}{44}.
\]

An approximation to \( f(3) = 1/3 \) is

\[
f(3) \approx P(3) = 0.32955.
\]

See Figure 3.1 for an illustration.

![Figure 3.1: The function \( f(x) = 1/x \) and its 2-nd Lagrange polynomial \( P(x) = \frac{1}{22}x^2 - \frac{35}{88}x + \frac{49}{44} \)](image)

The next step is to calculate a remainder term or bound for the error involved in approximating a function by an interpolating polynomial.

**Theorem 3.1.5.** Suppose \( x_0, x_1, \cdots, x_n \) are \( n + 1 \) distinct numbers in the interval \( [a, b] \) and \( f \in C^{n+1}[a, b] \). Then, for each \( x \in [a, b] \), a number \( \xi(x) \) (generally unknown) between \( x_0, \cdots, x_n \) and hence in \( (a, b) \) exists with

\[
f(x) = P(x) + \frac{f^{(n+1)}(\xi(x))}{(n + 1)!}(x - x_0)(x - x_1)\cdots(x - x_n),
\]

where \( P(x) \) is the interpolating polynomial given by (3.1).
Proof. Note that if we take \( x = x_k \), for any \( k = 0, 1, \cdots, n \), then we have
\[
f(x_k) = P(x_k).
\]
One may choose \( \xi(x) \) arbitrarily in the interval \( (a, b) \) and the equation (3.2) will hold.
If \( x \neq x_k \) for all \( k = 0, 1, \cdots, n \), we define the function \( g(t) \) for \( t \in [a, b] \) by
\[
g(t) = f(t) - P(t) - [f(x) - P(x)] \prod_{i=0}^{n} \frac{t - x_i}{x - x_i}.
\]
Since \( f \in C^{n+1}[a, b] \), and \( P \in C^{\infty}[a, b] \), it follows that \( g \in C^{n+1}[a, b] \). For \( t = x_k \), we have
\[
g(x_k) = f(x_k) - P(x_k) - [f(x) - P(x)] \prod_{i=0}^{n} \frac{x_k - x_i}{x - x_i} = 0 - [f(x) - P(x)] \cdot 0 = 0.
\]
Moreover, when \( t = x \), we have
\[
g(x) = f(x) - P(x) - [f(x) - P(x)] \prod_{i=0}^{n} \frac{x - x_i}{x - x_i} = f(x) - P(x) - [f(x) - P(x)] = 0.
\]
Thus, \( g \in C^{n+1}[a, b] \) and \( g \) is zero at the \( n+2 \) distinct numbers \( x, x_0, x_1, \cdots, x_n \). By Generalized Rolle’s Theorem \([1.1.16]\), there exists a number \( \xi \in (a, b) \) such that \( g^{(n+1)}(\xi) = 0 \). Hence, we have
\[
0 = g^{(n+1)}(\xi) = f^{(n+1)}(\xi) - P^{(n+1)}(\xi) - [f(x) - P(x)] \frac{d^{n+1}}{dt^{n+1}} \left[ \prod_{i=0}^{n} \frac{t - x_i}{x - x_i} \right]_{t=\xi}.
\]
(3.3)
However, \( P(x) \) is a polynomial of degree at most \( n \), so the \( n+1 \)-st derivative is identically zero.
Also, \( \prod_{i=0}^{n} (t - x_i)/(x - x_i) \) is a polynomial of degree \( n + 1 \), so we have
\[
\frac{d^{n+1}}{dt^{n+1}} \prod_{i=0}^{n} \frac{t - x_i}{x - x_i} = \frac{(n + 1)!}{\prod_{i=0}^{n}(x - x_i)}.
\]
Then, (3.3) now becomes
\[
0 = f^{(n+1)}(\xi) - 0 - [f(x) - P(x)] \prod_{i=0}^{n} \frac{(n + 1)!}{(x - x_i)} \implies f(x) = P(x) + \frac{f^{(n+1)}(\xi)}{(n + 1)!} \prod_{i=0}^{n}(x - x_i).
\]
This completes the proof. \( \square \)

The error formula is an important theoretical result because Lagrange polynomials are used extensively for deriving numerical differentiation and integration methods. Error bounds for these techniques are obtained from the Lagrange error formula. Note that the error form for the Lagrange polynomial is quite similar to that for the Taylor polynomial. The \( n \)-th Taylor polynomial about \( x_0 \) concentrates all the known information at \( x_0 \) and has an error term of the form
\[
\frac{f^{(n+1)}(\xi(x))}{(n + 1)!}(x - x_0)^{n+1}.
\]
The Lagrange polynomial of degree \( n \) uses information at the distinct numbers \( x_0, x_1, \cdots, x_n \) and, in place of \((x - x_0)^{n+1}\), its error formula uses a product of the \( n + 1 \) terms:
\[
\frac{f^{(n+1)}(\xi(x))}{(n + 1)!}(x - x_0)(x - x_1) \cdots (x - x_n).
\]
Example 3.1.6. In Example 3.1.4 we found the second Lagrange polynomial for \( f(x) = 1/x \) on \([2,4]\) using the nodes \(x_0 = 2, x_1 = 2.75, \) and \(x_2 = 4. \) Determine the error form for this polynomial, and the maximum error when the polynomial is used to approximate \( f(x) \) for \( x \in [2,4]. \)

**Solution.** Because \( f(x) = 1/x, \) we have

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

As a consequence, the second Lagrange polynomial has the error form

\[
\frac{f'''(\xi(x))}{3!} (x - x_0)(x - x_1)(x - x_2) = \frac{(x - 2)(x - 2.75)(x - 4)}{(\xi(x))^4} \quad \text{for} \quad \xi(x) \in (2,4).
\]

The maximum value of \((\xi(x))^4\) on the interval is \(2^{-4} = 1/16. \) We now need to determine the maximum value on this interval of the absolute value of the polynomial

\[
g(x) := (x - 2)(x - 2.75)(x - 4) = x^3 - \frac{35}{4} x^2 + \frac{49}{2} x - 22.
\]

Because

\[
g'(x) = 3x^2 - \frac{35}{2} x + \frac{49}{2} = \frac{1}{2}(3x - 7)(2x - 7),
\]

the critical points occur at

\[
x = \frac{7}{3} \quad \text{with} \quad g\left(\frac{7}{3}\right) = \frac{25}{108}, \quad \text{and} \quad x = \frac{7}{2} \quad \text{with} \quad g\left(\frac{7}{2}\right) = -\frac{9}{16}.
\]

Hence, the maximum error is

\[
\frac{f'''(\xi(x))}{3!} |(x - x_0)(x - x_1)(x - x_2)| \leq \frac{1}{16 \cdot 6} \left| -\frac{9}{16} \right| = \frac{3}{512} \approx 0.00586.
\]

\(\square\)

### 3.2 Data Approximation and Neville’s Method

A practical difficulty with Lagrange interpolation is that the error term is difficult to apply, so the degree of the polynomial needed for the desired accuracy is generally not known until computations have been performed. A common practice is to compute the results given from various polynomials until appropriate agreement is obtained. However, the work done in calculating the approximation by the second polynomial does not lessen the work needed to calculate the third approximation; nor is the fourth approximation easier to obtain once the third approximation is known, and so on. We now derive these approximating polynomials in a recursive manner.

**Definition 3.2.1.** Let \(f\) be a function defined at \(x_0, x_1, \ldots, x_n\) and suppose that \(m_1, m_2, \ldots, m_k\) are \(k\) distinct integers, with \(0 \leq m_i \leq n\) for each \(i = 1, \ldots, k.\) The Lagrange polynomial that agrees with \(f(x)\) at the \(k\) points \(x_{m_1}, \ldots, x_{m_k}\) is denoted \(P_{m_1,m_2,\ldots,m_k}(x).\)

**Example 3.2.2.** Let \(x_0 = 1, \ x_1 = 2, \ x_2 = 3, \ x_3 = 4, \ x_4 = 6, \) and \(f(x) = e^x.\) Determine the interpolating polynomial denoted \(P_{1,2,4}(x),\) and use this polynomial to approximate \(f(5).\)
Solution. This is the Lagrange polynomial that agrees with \( f(x) \) at \( x_1 = 2, \ x_2 = 3, \ x_4 = 6 \). Hence,
\[
P_{1,2,4}(x) = \frac{(x-3)(x-6)}{(2-3)(2-6)} e^2 + \frac{(x-2)(x-6)}{(3-2)(3-6)} e^3 + \frac{(x-2)(x-3)}{(6-2)(6-3)} e^6.
\]
Hence,
\[
f(5) \approx P(5) = \frac{(5-3)(5-6)}{(2-3)(2-6)} e^2 + \frac{(5-2)(5-6)}{(3-2)(3-6)} e^3 + \frac{(5-2)(5-3)}{(6-2)(6-3)} e^6 \approx 218.105.
\]

The next result describes a method for recursively generating Lagrange polynomial approximations.

**Theorem 3.2.3.** Let \( f \) be defined at \( x_0, x_1, \ldots, x_n \) and let \( x_j \) and \( x_i \) be two distinct numbers in this set. Then
\[
P(x) = \frac{(x - x_j)P_{0,1,\ldots,i-1,j+1,\ldots,k}(x) - (x - x_i)P_{0,1,\ldots,i-1,i+1,\ldots,k}(x)}{x_i - x_j}
\]
is the \( k \)-th Lagrange polynomial that interpolates at the \( k + 1 \) points \( x_0, x_1, \ldots, x_k \).

**Proof.** For ease of notation, let \( Q = P_{0,1,\ldots,i-1,i+1,\ldots,k} \) and \( \hat{Q} = P_{0,1,\ldots,j-1,j+1,\ldots,k} \). Since \( Q(x) \) and \( \hat{Q}(x) \) are polynomials of degree \( k - 1 \) or less, \( P(x) \) is of degree at most \( k \). First note that \( \hat{Q}(x_i) = f(x_i) \), implies that
\[
P(x_i) = \frac{(x_i - x_j)\hat{Q}(x_i) - (x_i - x_i)Q(x_i)}{x_i - x_j} = \frac{(x_i - x_j)}{(x_i - x_j)} f(x_i) = f(x_i).
\]
Similarly, since \( Q(x_j) = f(x_j) \), we have \( P(x_j) = f(x_j) \). In addition, if \( 0 \leq r \leq k \) and \( r \) is neither \( i \) nor \( j \), then \( Q(x_r) = \hat{Q}(x_r) = f(x_r) \). Hence, we have
\[
P(x_r) = \frac{(x_r - x_j)\hat{Q}(x_r) - (x_r - x_i)Q(x_r)}{x_i - x_j} = \frac{(x_i - x_j)}{(x_i - x_j)} f(x_r) = f(x_r).
\]
But, by definition, \( P_{0,1,\ldots,k}(x) \) is the unique polynomial of degree at most \( k \) that agrees with \( f \) at \( x_0, x_1, \ldots, x_k \). Thus, \( P = P_{0,1,\ldots,k} \). \( \square \)

The theorem above implies that the interpolating polynomials can be generated recursively. For example, we have
\[
P_{0,1} = \frac{1}{x_1 - x_0}[(x - x_0)P_1 - (x - x_1)P_0], \quad P_{1,2} = \frac{1}{x_2 - x_1}[(x - x_1)P_2 - (x - x_2)P_1],
\]
\[
P_{0,1,2} = \frac{1}{x_2 - x_0}[(x - x_0)P_{1,2} - (x - x_2)P_{0,1}],
\]
and so on. They are generated in the manner shown in the following table, which each row is completed before the succeeding rows are begun.

The procedure that uses the result of the theorem above to recursively generate interpolating polynomial approximations is called **Neville’s method**. To avoid the multiple subscripts, we let \( Q_{i,j}(x) \) for \( 0 \leq j \leq i \), denote the interpolating polynomial of degree \( j \) on the \( j + 1 \) numbers \( x_{i-j}, x_{i-j+1}, \ldots, x_{i-1}, x_i \); that is,
\[
Q_{i,j} = P_{i-j,i-j+1,\ldots,i-1,i}.
\]
Using this notation provides the \( Q \) notation array in Table 3.2.
Then, if the latest approximation was not sufficiently accurate, another node $x_5$, could be added, and another row added to the table:

<table>
<thead>
<tr>
<th>$x_5$</th>
<th>$Q_{5,0}$</th>
<th>$Q_{5,1}$</th>
<th>$Q_{5,2}$</th>
<th>$Q_{5,3}$</th>
<th>$Q_{5,4}$</th>
<th>$Q_{5,5}$</th>
</tr>
</thead>
</table>

Then, $Q_{4,4}$, $Q_{5,4}$, and $Q_{5,5}$ could be compared to determine further accuracy.
3.3 Divided Differences

Iterated interpolation was used in the previous section to generate successively higher-degree polynomial approximation at a specific point. Divided-difference method introduced in this section are used to successively generate the polynomials themselves.

Suppose that \( P_n(x) \) is the \( n \)-th Lagrange polynomial that agrees with the function \( f \) at the distinct numbers \( x_0, x_1, \ldots, x_n \). Although this polynomial is unique, there are alternate algebraic representations that are useful in certain situations. The divided differences of \( f \) with respect to \( x_0, x_1, \ldots, x_n \) are used to express \( P_n(x) \) in the form

\[
P_n(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \cdots + a_n(x-x_0)(x-x_1) \cdots (x-x_{n-1}),
\]

(3.4)

for appropriate constants \( a_0, a_1, \ldots, a_n \). To determine the first of these constants, \( a_0 \), note that if \( P_n(x) \) is written in the form of (3.4), then evaluating \( P_n(x_0) \) leaves only the constant term \( a_0 \); that is

\[
a_0 = P_n(x_0) = f(x_0).
\]

Similarly, when \( P_n(x) \) is evaluated at \( x_1 \), the only nonzero terms in the evaluation of \( P_n(x) \) are the constant and linear terms,

\[
f(x_0) + a_1(x_1-x_0) = P_n(x_1) = f(x_1) \implies a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}.
\]

We now introduce the divided-difference notation. The zeroth divided difference of the function \( f \) with respect to \( x_i \), denoted \( f[x_i] \), is simply the value of \( f \) at \( x_i \):

\[
f[x_i] = f(x_i).
\]

(3.5)

The remaining divided differences are defined recursively; the first divided difference of \( f \) with respect to \( x_i \) and \( x_{i+1} \) is denoted \( f[x_i, x_{i+1}] \) and defined as

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

(3.6)

The second divided difference, \( f[x_i, x_{i+1}, x_{i+2}] \) is defined as

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

(3.7)

Similarly, after the \( (k-1) \)-st divided differences,

\[
f[x_i, x_{i+1}, x_{i+2}, \ldots, x_{i+k-1}] \quad \text{and} \quad f[x_{i+1}, x_{i+2}, \ldots, x_{i+k-1}, x_{i+k}],
\]

have been determined, the \( k \)-th divided difference relative to \( x_i, x_{i+1}, \ldots, x_{i+k} \) is defined as

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

(3.8)

The process ends with the single \( n \)-th divided difference

\[
f[x_0, x_1, \ldots, x_n] = \frac{f[x_1, x_2, \ldots, x_n] - f[x_0, x_1, \ldots, x_{n-1}]}{x_n - x_0}.
\]

Now we can write \( a_1 = f[x_0, x_1] \), just as \( a_0 \) can be expressed as \( a_0 = f[x_0] \). Hence, the interpolating polynomial in (3.3) can be written as

\[
P_n(x) = f[x_0] + f[x_0, x_1](x-x_1) + a_2(x-x_0)(x-x_1) + \cdots + a_n(x-x_0)(x-x_1) \cdots (x-x_{n-1}),
\]
3.3. DIVIDED DIFFERENCES

\begin{tabular}{|c|c|c|}
\hline
\textbf{x} & \textbf{f(x)} & \textbf{First divided differences} & \textbf{Second divided differences} \\
\hline
\textbf{x}_0 & \textbf{f(x)} & f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0} & \\
\hline
\textbf{x}_1 & \textbf{f(x)} & f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} & \\
\hline
\textbf{x}_2 & \textbf{f(x)} & f[x_1, x_2, x_3] = \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1} & \\
\hline
\textbf{x}_3 & \textbf{f(x)} & f[x_2, x_3, x_4] = \frac{f[x_3, x_4] - f[x_2, x_3]}{x_4 - x_2} & \\
\hline
\textbf{x}_4 & \textbf{f(x)} & f[x_3, x_4, x_5] = \frac{f[x_4, x_5] - f[x_3, x_4]}{x_5 - x_3} & \\
\hline
\textbf{x}_5 & \textbf{f(x)} & f[x_4, x_5] = \frac{f[x_5] - f[x_4]}{x_5 - x_4} & \\
\hline
\end{tabular}

Table 3.3: The divided difference table.

where \(a_k = f[x_0, x_1, \ldots, x_k]\), for each \(k = 0, \ldots, n\). Hence, \(P_n(x)\) can be rewritten in a form called Newton’s Divided-Difference:

\[
P_n(x) = f(x_0) + \sum_{k=1}^{n} f[x_0, x_1, \ldots, x_k] (x - x_0) \cdots (x - x_{k-1}).
\]  \hfill (3.9)

The value of \(f[x_0, \ldots, x_k]\) is independent of the order of the number \(x_0, \ldots, x_k\).

The generation of the divided differences is outlined in the following table. Two fourth and one fifth difference can also be determined from these data.

**Example 3.3.1.** Complete the divided difference table for the following data, and construct the interpolating polynomial that uses all this data.

<table>
<thead>
<tr>
<th>(x)</th>
<th>1.0</th>
<th>1.3</th>
<th>1.6</th>
<th>1.9</th>
<th>2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(f(x))</td>
<td>0.7651977</td>
<td>0.6200860</td>
<td>0.4554022</td>
<td>0.2818186</td>
<td>0.1103623</td>
</tr>
</tbody>
</table>

**Solution.** The first divided difference involving \(x_0\) and \(x_1\) is

\[
f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{0.6200860 - 0.7651977}{1.3 - 1.0} = -0.4837057.
\]

The remaining first divided differences are found in a similar manner and are shown in the fourth column in the following table.
The Mean Value Theorem \(1.1.14\) applied to (3.6) when \(i = 0\),

\[ f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}, \]

implies that when \(f'\) exists, \(f[x_0, x_1] = f'(\xi)\) for some number \(\xi\) between \(x_0\) and \(x_1\). The following theorem generalizes this result.

**Theorem 3.3.2.** Suppose that \(f \in C^n[a, b]\) and \(x_0, x_1, \ldots, x_n\) are distinct numbers in \([a, b]\). Then, a number \(\xi\) exists in \((a, b)\) with

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

**Proof.** Let \(g(x) := f(x) - P_n(x)\). Since \(f(x_i) = P_n(x_i)\) for each \(i = 0, 1, \ldots, n\), the function \(g\) has \(n + 1\) distinct zeros in \([a, b]\). Generalized Rolle’s Theorem \(1.1.16\) implies that a number \(\xi \in (a, b)\) exists with \(g^{(n)}(\xi) = 0\), so

\[ 0 = f^{(n)}(\xi) - P_n^{(n)}(\xi). \]

Since \(P_n\) is a polynomial of degree \(n\) whose leading coefficient is \(f[x_0, x_1, \ldots, x_n]\), then

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

for all values of \(x \in [a, b]\). As a consequence, we have

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

\[ \square \]

Newton’s divided-difference formula can be expressed in a simplified form when the nodes are arranged consecutively with equal spacing. In this case, we introduce the notation \(h := x_{i+1} - x_i\) for each \(i = 0, 1, \ldots, n - 1\) and let \(x = x_0 + sh\). Then, the difference \(x - x_i = (s - i)h\). Hence, (3.9) becomes

\[
P_n(x) = P_n(x_0 + sh) = f[x_0] + shf[x_0, x_1] + s(s-1)h^2f[x_0, x_1, x_2] \\
+ \cdots + s(s-1)\cdots(s-n+1)h^nf[x_0, x_1, \ldots, x_n] \\
= f[x_0] + \sum_{k=1}^{n} s(s-1)\cdots(s-k+1)h^kf[x_0, x_1, \ldots, x_k].
\]
Using binomial-coefficient notation,
\[ \binom{s}{k} = \frac{s(s-1) \cdots (s-k+1)}{k!}, \]
we can express \( P_n(x) \) compactly as
\[ P_n(x) = P_n(x_0 + sh) = f[x_0] + \sum_{k=1}^{n} \binom{s}{k} k!h^k f[x_0, x_1, \cdots, x_k]. \]

**Remark:** One can use the notation of forward difference (or backward difference) to rewrite the Newton’s divided-difference formula above. This is just a book keeping and leave as an exercise.

### 3.4 Hermite Interpolation

In this section, we introduce the concept of Hermite polynomial. For a given function \( f \), the Hermite polynomial \( H(x) \) agree with \( f \) at \( x_0, x_1, \cdots, x_n \). Moreover, it requires that the first derivatives of the Hermite polynomial \( H(x) \) agree with those of \( f \).

More precisely, the Hermite polynomial \( H(x) \) corresponding to the function \( f \) at the given points \( x_0, x_1, \cdots, x_n \) satisfies the following conditions:
\[ H(x_i) = f(x_i) \quad \text{and} \quad H'(x_i) = f'(x_i) \quad \text{for} \quad i = 0, 1, \cdots, n. \]

In total, there are \( 2n + 2 \) constraints for the Hermite polynomial \( H(x) \) and therefore it is a polynomial with degree at most \( 2n + 1 \). In order to find out the basis functions for the Hermite polynomial, we make use of the Lagrange basis functions introduced in Section 3.1.

To simplify the notation, we define \( \delta_{jk} \) satisfying the following conditions:
\[ \delta_{jk} = \begin{cases} 1 & \text{if } j = k \\ 0 & \text{if } j \neq k. \end{cases} \]

Suppose that we have developed two sets of basis functions with degree \( 2n + 1 \), denoted \( \{H_k^{(n)}\} \) and \( \{\hat{H}_k^{(n)}\} \), for Hermite polynomial such that the following conditions are satisfied: for \( j = 0, 1, \cdots, n, \)
\[
H_k^{(n)}(x_j) = \delta_{jk}, \quad \left(\frac{d}{dx}\right)^j H_k^{(n)}(x_j) = 0,
\]
\[
\hat{H}_k^{(n)}(x_j) = 0, \quad \left(\frac{d}{dx}\right)^j \hat{H}_k^{(n)}(x_j) = \delta_{jk}.
\]

Then, the desired Hermite polynomial \( H(x) \) (with degree at most \( 2n + 1 \)) can be written as the linear combination of those basis functions:
\[ H(x) = \sum_{j=0}^{n} f(x_j) H_k^{(n)}(x) + f'(x_j) \hat{H}_k^{(n)}(x). \]

It remains to find out the formula for \( H_k^{(n)} \) and \( \hat{H}_k^{(n)} \). Recall that the Lagrange basis functions with degree \( n \) are defined as follows: for \( k = 0, 1, \cdots, n \)
\[
L_k^{(n)}(x) = \frac{(x - x_0)(x - x_1) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_n)}{(x_k - x_0)(x_k - x_1) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_n)} = \prod_{i=0, i \neq k}^{n} \frac{x - x_i}{x_k - x_i}.
\]
We make use of the Lagrange basis functions to derive the formula for \( H_k^{(n)} \) and \( \hat{H}_k^{(n)} \). Since for each \( k = 0, 1, \cdots, n \) the polynomial \( H_k^{(n)} \) is of degree \( 2n + 1 \), one may assume that \( H_k^{(n)} \) has the following form
\[
H_k^{(n)}(x) = (ax + b) \left( L_k^{(n)}(x) \right)^2.
\]

Using the conditions in the first line of (3.10) for \( H_k^{(n)} \), we have
\[
a x_k + b = 1 \quad \text{and} \quad a + 2(ax_k + b) \left( L_k^{(n)} \right)'(x_k) = 0.
\]
It implies that
\[
a = -2 \left( L_k^{(n)} \right)'(x_k) \quad \text{and} \quad b = 1 + 2x_k \left( L_k^{(n)} \right)'(x_k).
\]
Hence, we have
\[
H_k^{(n)}(x) = \left( 1 - 2(x - x_k) \left( L_k^{(n)} \right)'(x_k) \right) \left( L_k^{(n)}(x) \right)^2 \quad \text{for} \quad k = 0, 1, \cdots, n.
\]
Similarly, we assume that
\[
\hat{H}_k^{(n)}(x) = (cx + d) \left( L_k^{(n)}(x) \right)^2.
\]
Using the conditions in the second line of (3.10) for \( \hat{H}_k^{(n)} \), we have
\[
c x_k + d = 0 \quad \text{and} \quad c + 2(cx_k + d) \left( L_k^{(n)} \right)'(x_k) = 1.
\]
It implies that
\[
c = 1 \quad \text{and} \quad d = -x_k.
\]
Hence, we have
\[
\hat{H}_k^{(n)}(x) = (x - x_k) \left( L_k^{(n)}(x) \right)^2.
\]
We summarize the results above as a theorem.

**Theorem 3.4.1.** If \( f \in C^1[a, b] \) and \( x_0, \cdots, x_n \in [a, b] \) are distinct, the unique polynomial of least degree agreeing with \( f \) and \( f' \) at \( x_0, \cdots, x_n \) is the Hermite polynomial of degree at most \( 2n + 1 \), given by
\[
H_{2n+1}(x) = \sum_{j=0}^{n} f(x_j) H_{n,j}(x) + f'(x_j) \hat{H}_{n,j}(x),
\]
where, for \( L_{n,j}(x) \) denoting \( j \)-th Lagrange coefficient polynomial of degree \( n \), we have
\[
H_{n,j}(x) = \left[ 1 - 2(x - x_j)L_j'(x_j) \right] L_{n,j}^2(x) \quad \text{and} \quad \hat{H}_{n,j}(x) = (x - x_j)L_{n,j}^2(x).
\]
Moreover, if \( f \in C^{2n+2}[a, b] \), then
\[
f(x) = H_{2n+1}(x) + \frac{(x - x_0)^2 \cdots (x - x_n)^2}{(2n + 2)!} f^{(2n+2)}(\xi(x)),
\]
for some (generally unknown) \( \xi(x) \) in the interval \( (a, b) \).
When \( n = 1 \), there are two points \( x_0 \) and \( x_1 \) given and we may explicitly write down the complete formulas for basis functions for Hermite polynomial:

\[
H_0^{(1)}(x) = \left(1 - 2 \frac{x - x_0}{x_0 - x_1}\right) \left(\frac{x - x_1}{x_0 - x_1}\right)^2,
\]

\[
H_1^{(1)}(x) = \left(1 - 2 \frac{x - x_1}{x_1 - x_0}\right) \left(\frac{x - x_0}{x_1 - x_0}\right)^2,
\]

\[
\tilde{H}_0^{(1)}(x) = (x - x_0) \left(\frac{x - x_1}{x_0 - x_1}\right)^2,
\]

\[
\tilde{H}_1^{(1)}(x) = (x - x_1) \left(\frac{x - x_0}{x_1 - x_0}\right)^2.
\]

**Example 3.4.2.** Let \( f(x) = \ln x \). Given

\[
f(1.0) = 0.0, \quad f(2.0) = 0.693147, \quad f'(1.0) = 1.0, \quad \text{and} \quad f'(2.0) = 0.5,
\]

find the Hermite polynomial \( H(x) \) corresponding to the function \( f \) about \( x_0 = 1.0 \) and \( x_1 = 2.0 \). Use \( H(1.5) \) as an approximation of \( f(1.5) \) and compute the absolute error.

**Solution.** Using the formula above, we have

\[
H(x) = 0.693147(5 - 2x)(x - 1)^2 + (x - 1)(x - 2)^2 + \frac{1}{2}(x - 2)(x - 1)^2
\]

and we have \( H(1.5) = 0.409074 \). One may also obtain \( f(1.5) = 0.405465 \) and

\[
|f(1.5) - H(1.5)| \approx 0.0036089.
\]

See the figure below for an illustration. They look like almost the same.

**Remark:** More generally, the so-called osculating polynomial is generalizing both the Taylor polynomial and Lagrange polynomial. Suppose that we are given \( n + 1 \) distinct numbers \( x_0, x_1, \cdots, x_n \) in \([a, b]\) and nonnegative integers \( m_0, m_1, \cdots, m_n \), and \( m = \max\{m_0, m_1, \cdots, m_n\} \).

The osculating polynomial approximating a function \( f \in C^m[a, b] \) at \( x_i \), for each \( i = 0, \cdots, n \), is the polynomial of least degree that has the same values as the function \( f \) and all its derivatives of order less than or equal to \( m_i \) at each \( x_i \). The degree of this osculating polynomial is at most

\[
M = n + \sum_{i=0}^{n} m_i
\]
because the number of condition to be satisfied is $n + 1 + \sum_{i=0}^{n} m_i$, and a polynomial of degree $M$ has $M + 1$ coefficients that can be used to satisfy these conditions.

**Definition 3.4.3** (Osculating Polynomial). Let $x_0, x_1, \cdots, x_n$ be $n + 1$ distinct numbers in $[a, b]$ and for $i = 0, 1, \cdots, n$ let $m_i$ be a nonnegative integer. Suppose that $f \in C^m[a, b]$, where $m = \max_{0 \leq i \leq n} m_i$. The osculating polynomial approximating $f$ is the polynomial $P$ of least degree such that

$$\frac{d^k P(x_i)}{dx^k} = \frac{d^k f(x_i)}{dx^k} \quad \text{for each } i = 0, 1, \cdots, n \quad \text{and} \quad k = 0, 1, \cdots, m_i.$$  

Note that when $n = 0$, the osculating polynomial approximating $f$ is the $m_0$-th Taylor Polynomial for $f$ at $x_0$. When $m_i = 0$ for each $i$, the osculating polynomial is the $n$-th Lagrange polynomial interpolating $f$ on $x_0, x_1, \cdots, x_n$.

The case when $m_i = 1$, for each $i = 0, 1, \cdots, n$, gives the Hermite polynomials. For a given function $f$, these polynomials agree with $f$ at $x_0, x_1, \cdots, x_n$. In addition, since their first derivatives agree with those of $f$, they have the same shape as the function at $(x_i, f(x_i))$ in the sense that the tangent lines to be the polynomial and the function agree.

Although Theorem 3.4.1 provides a complete description of the Hermite polynomials, it is clear from the above example that the need to determine and evaluate the Lagrange polynomials and their derivatives makes the procedure tedious even for small values of $n$.

There is an alternative method for generating Hermite approximations that has as its basis the Newton interpolatory divided-difference formula at $x_0, x_1, \cdots, x_n$, that is

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

The alternative method uses the connection between the $n$-th divided difference and the $n$-th derivative of $f$. Suppose that the distinct numbers $x_0, x_1, \cdots, x_n$ are given together with the values of $f$ and $f'$ at these numbers. Define a new sequence $z_0, z_1, \cdots, z_{2n+1}$ by

$$z_{2i} = z_{2i+1} = x_i, \quad \text{for each } i = 0, 1, \cdots, n$$

and construct the divided difference table in the form of Table 3.3 that uses $z_0, z_1, \cdots, z_{2n+1}$. Since $z_{2i} = z_{2i+1} = x_i$ for each $i$, we cannot define $f[z_{2i}, z_{2i+1}]$ by the divided difference formula. However, if we assume that the reasonable substitution in this situation is $f[z_{2i}, z_{2i+1}] = f'(z_{2i}) = f'(x_i)$, we can use the entries $f'(x_0), f'(x_1), \cdots, f'(x_n)$ in place of the undefined first divided differences

$$f[z_0, z_1], \ f[z_2, z_3], \cdots, f[z_{2n}, z_{2n+1}].$$

The remaining divided differences are produced as usual, and the appropriate divided difference are employed in Newton’s interpolatory divided-difference formula. Table 3.4 shows the entries that are used for the first three divided-difference columns when determining the Hermite polynomial $H_5(x)$ for $x_0, x_1,$ and $x_2$. The remaining entries are generated in the same manner. The Hermite polynomial is then given by

$$H_{2n+1}(x) = f[z_0] + \sum_{k=1}^{2n+1} f[z_0, \cdots, z_k](x - z_0)(x - z_1) \cdots (x - z_{k-1}).$$
3.5 PIECEWISE POLYNOMIALS APPROXIMATION

The previous sections concerned the approximation of arbitrary functions on closed intervals using a single polynomial. However, high-degree polynomials can oscillate erratically, that is, a minor fluctuation over a small portion of the interval can induce large fluctuations over the entire range.

**Example 3.5.1** (Runge’s Phenomenon). Let \( f(x) = \frac{1}{1 + x^2} \) and \( x \in [-5, 5] \). We partition the interval using the nodes \( x_k = -5 + kh \) with \( k = 0, 1, \ldots, 10 \). Find the 10-th Lagrange Interpolation Polynomial of \( f(x) \).

**Solution.** The 10-th Lagrange polynomial \( P \) can be written as follows:

\[
P(x) = \sum_{k=0}^{10} f(x_k)L_k(x).
\]

The shapes of the function \( f \) and \( P \) can be found in the following figure. One can see from the graph that when \( x \) is between \(-3 \) and \( 3 \), the difference \( |f(x) - P(x)| \) is quite small. However, when \( |x| > 3 \), the difference is large, especially when \( x \) is near the endpoints, the difference becomes dramatically large. In general, one may not use a higher-order Lagrange interpolating polynomial to approximate a function.

An alternative approach is to divide the approximation interval into a collection of subintervals and construct a (generally) different approximating polynomial on each subinterval. This is called **piecewise-polynomial approximation**. The simplest piecewise-polynomial approximation is **piecewise-linear** interpolation, which consists of joining a set of data points

\[
\{(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))\}
\]

by a series of straight lines. See Figure 3.3 for illustration.
To be more specific, in each subinterval \([x_i, x_{i+1}]\) for \(i = 0, 1, \cdots, n - 1\), we use a first-order Lagrange Interpolation polynomial to approximate the function \(f(x)\). The Lagrange basis functions in this subinterval \([x_i, x_{i+1}]\) are given by the following formulas

\[
L_i^{(1)}(x) = \frac{x - x_{i+1}}{x_i - x_{i+1}} \quad \text{and} \quad L_{i+1}^{(1)}(x) = \frac{x - x_i}{x_{i+1} - x_i}.
\]

Then, the corresponding (first-order) Lagrange Interpolation polynomial \(P_i\) is

\[
P_i(x) = f(x_i)L_i^{(1)}(x) + f(x_{i+1})L_{i+1}^{(1)}(x) \quad \text{for} \quad x \in [x_i, x_{i+1}].
\]

Eventually, the piecewise linear function \(Q(x)\) is defined to be

\[
Q(x) = P_i(x), \quad \text{for} \quad x \in [x_i, x_{i+1}], \quad i = 0, 1, \cdots, n - 1.
\]

If \(f\) is at least twice differentiable, then the remainder can be written as follows: for \(x \in [x_i, x_{i+1}]\),

\[
R_i(x) = f(x) - P_i(x) = \frac{1}{2}f''(\xi(x))(x - x_i)(x - x_{i+1})
\]
and we have the following error estimate

\[ |R_i(x)| \leq \frac{1}{8} (x_{i+1} - x_i)^2 \max_{x \in [x_i, x_{i+1}]} |f''(x)|, \quad \text{for all } x \in [x_i, x_{i+1}]. \]

If the points \( x_0, x_1, \ldots, x_n \) (assuming \( x_0 \leq x_1 \leq \cdots \leq x_n \)) are equally distributed and let \( h = \frac{x_n - x_0}{n} \), then we have

\[ |f(x) - Q(x)| \leq \frac{h^2}{8} \max_{x \in [x_0, x_n]} |f''(x)|, \quad \text{for all } x \in [x_0, x_n]. \]

A disadvantage of linear function approximation is that there is likely no differentiability at the endpoints of the subintervals, which, in a geometrical context, means that the interpolating function is not \textbf{smooth}. Often it is clear from physical condition that smoothness is required, so the approximating function must be continuously differentiable.

An alternative is to use a piecewise polynomial of Hermite type. For example, if the values of \( f \) and of \( f' \) are known at each of the points \( x_0 < x_1 < \cdots < x_n \), a cubic Hermite polynomial can be used on each of the subintervals \([x_0, x_1], [x_1, x_2], \cdots, [x_{n-1}, x_n]\) to obtain a function that has a continuous derivative on the interval \([x_0, x_n]\).

To determine the appropriate Hermite cubic polynomial on a given interval is simply a matter of computing \( H_3(x) \) for that interval. The Lagrange interpolating polynomials needed to determine \( H_3 \) are of first degree, so this can be accomplished without great difficulty. However, to use Hermite piecewise polynomials for general interpolation, we need to know the derivative of the function being approximated, and this is frequently unavailable.