3 Interpolation

3.1 What is Interpolation?

Imagine that there is an unknown function $f(x)$ for which someone supplies you with its (exact) values at $(n+1)$ distinct points $x_0 < x_1 < \cdots < x_n$, i.e., $f(x_0), \ldots, f(x_n)$ are given. The interpolation problem is to construct a function $Q(x)$ that passes through these points, i.e., to find a function $Q(x)$ such that the **interpolation requirements**

$$Q(x_j) = f(x_j), \qquad 0 \leqslant j \leqslant n, \tag{3.1}$$

are satisfied (see Figure 3.1). One easy way of obtaining such a function, is to connect the given points with straight lines. While this is a legitimate solution of the interpolation problem, usually (though not always) we are interested in a different kind of a solution, e.g., a smoother function. We therefore always specify a certain class of functions from which we would like to find one that solves the **interpolation problem**. For example, we may look for a function $Q(x)$ that is a polynomial, $Q(x)$. Alternatively, the function $Q(x)$ can be a trigonometric function or a piecewise-smooth polynomial, and so on.

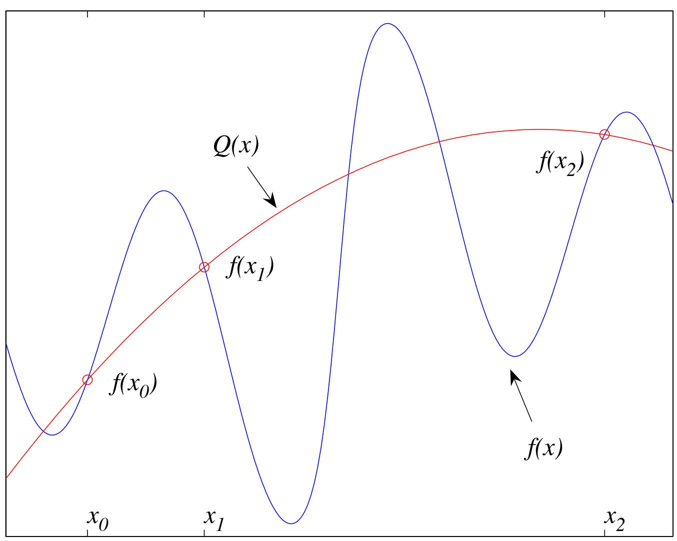

Figure 3.1: The function $f(x)$, the interpolation points x_0, x_1, x_2, and the interpolating polynomial $Q(x)$

As a simple example let's consider values of a function that are prescribed at two points: $(x_0, f(x_0))$ and $(x_1, f(x_1))$. There are infinitely many functions that pass through these two points. However, if we limit ourselves to polynomials of degree less than or equal to one, there is only one such function that passes through these two points: the line that connects them. A line, in general, is a polynomial of degree one, but if the two given values are equal, $f(x_0) = f(x_1)$, the line that connects them is the constant $Q_0(x) \equiv f(x_0)$, which is a polynomial of degree zero. This is why we say that there is a unique polynomial of degree $\leqslant 1$ that connects these two points (and not "a polynomial of degree 1").

The points $x_0, \ldots, x_n$ are called the **interpolation points**. The property of "passing through these points" is referred to as **interpolating the data**. The function that interpolates the data is **an interpolant** or **an interpolating polynomial** (or whatever function is being used).

There are cases were the interpolation problem has no solution, e.g., if we look for a linear polynomial that interpolates three points that do not lie on a straight line. When a solution exists, it can be unique (a linear polynomial and two points), or the problem can have more than one solution (a quadratic polynomial and two points). What we are going to study in this section is precisely how to distinguish between these cases. We are also going to present different approaches to constructing the interpolant.

Other than agreeing at the interpolation points, the interpolant $Q(x)$ and the underlying function $f(x)$ are generally different. The **interpolation error** is a measure on how different these two functions are. We will study ways of estimating the interpolation error. We will also discuss strategies on how to minimize this error.

It is important to note that it is possible to formulate the interpolation problem without referring to (or even assuming the existence of) any underlying function $f(x)$. For example, you may have a list of interpolation points $x_0, \ldots, x_n$, and data that is experimentally collected at these points, $y_0, y_1, \ldots, y_n$, which you would like to interpolate. The solution to this interpolation problem is identical to the one where the values are taken from an underlying function.

3.2 The Interpolation Problem

We begin our study with the problem of **polynomial interpolation**: Given $n + 1$ distinct points $x_0, \ldots, x_n$, we seek a polynomial $Q_n(x)$ of the lowest degree such that the following interpolation conditions are satisfied:

$$Q_n(x_j) = f(x_j), \qquad j = 0, \ldots, n. \tag{3.2}$$

Note that we do not assume any ordering between the points $x_0, \ldots, x_n$, as such an order will make no difference. If we do not limit the degree of the interpolation polynomial it is easy to see that there any infinitely many polynomials that interpolate the data. However, limiting the degree of $Q_n(x)$ to be $deg(Q_n(x)) \leqslant n$, singles out precisely one interpolant that will do the job. For example, if $n = 1$, there are infinitely many polynomials that interpolate $(x_0, f(x_0))$ and $(x_1, f(x_1))$. However, there is only one polynomial $Q_n(x)$ with $deg(Q_n(x)) \leqslant 1$ that does the job. This result is formally stated in the following theorem:

Theorem 3.1 *If $x_0, \ldots, x_n \in \mathbb{R}$ are distinct, then for any $f(x_0), \ldots f(x_n)$ there exists a unique polynomial $Q_n(x)$ of degree $\leqslant n$ such that the interpolation conditions (3.2) are satisfied.*

Proof. We start with the *existence* part and prove the result by induction. For $n = 0$, $Q_0 = f(x_0)$. Suppose that Q_{n-1} is a polynomial of degree $\leqslant n-1$, and suppose also that

$$Q_{n-1}(x_j) = f(x_j), \qquad 0 \leqslant j \leqslant n-1.$$

Let us now construct from $Q_{n-1}(x)$ a new polynomial, $Q_n(x)$, in the following way:

$$Q_n(x) = Q_{n-1}(x) + c(x - x_0) \cdot \ldots \cdot (x - x_{n-1}). \tag{3.3}$$

The constant c in (3.3) is yet to be determined. Clearly, the construction of $Q_n(x)$ implies that $\deg(Q_n(x)) \leqslant n$. (Since we might end up with $c = 0$, $Q_n(x)$ could actually be of degree that is less than n.) In addition, the polynomial $Q_n(x)$ satisfies the interpolation requirements $Q_n(x_j) = f(x_j)$ for $0 \leqslant j \leqslant n-1$. All that remains is to determine the constant c in such a way that the last interpolation condition, $Q_n(x_n) = f(x_n)$, is satisfied, i.e.,

$$Q_n(x_n) = Q_{n-1}(x_n) + c(x_n - x_0) \cdot \ldots \cdot (x_n - x_{n-1}). \tag{3.4}$$

The condition (3.4) implies that c should be defined as

$$c = \frac{f(x_n) - Q_{n-1}(x_n)}{\displaystyle\prod_{j=0}^{n-1}(x_n - x_j)}, \tag{3.5}$$

and we are done with the proof of existence.
As for *uniqueness*, suppose that there are two polynomials $Q_n(x), P_n(x)$ of degree $\leqslant n$ that satisfy the interpolation conditions (3.2). Define a polynomial $H_n(x)$ as the difference

$$H_n(x) = Q_n(x) - P_n(x).$$

The degree of $H_n(x)$ is at most n which means that it can have at most n zeros (unless it is identically zero). However, since both $Q_n(x)$ and $P_n(x)$ satisfy all the interpolation requirements (3.2), we have

$$H_n(x_j) = (Q_n - P_n)(x_j) = 0, \qquad 0 \leqslant j \leqslant n,$$

which means that $H_n(x)$ has $n+1$ distinct zeros. This contradiction can be resolved only if $H_n(x)$ is the zero polynomial, i.e.,

$$P_n(x) \equiv Q_n(x),$$

and uniqueness is established. ∎

3.3 Newton's Form of the Interpolation Polynomial

One good thing about the proof of Theorem 3.1 is that it is constructive. In other words, we can use the proof to write down a formula for the interpolation polynomial. We follow the procedure given by (3.4) for reconstructing the interpolation polynomial. We do it in the following way:

- Let
$$Q_0(x) = a_0,$$
where $a_0 = f(x_0)$.

- Let
$$Q_1(x) = a_0 + a_1(x - x_0).$$
Following (3.5) we have
$$a_1 = \frac{f(x_1) - Q_0(x_1)}{x_1 - x_0} = \frac{f(x_1) - f(x_0)}{x_1 - x_0}.$$
We note that $Q_1(x)$ is nothing but the straight line connecting the two points $(x_0, f(x_0))$ and $(x_1, f(x_1))$.

In general, let

$$\begin{aligned} Q_n(x) &= a_0 + a_1(x - x_0) + \ldots + a_n(x - x_0) \cdot \ldots \cdot (x - x_{n-1}) \\ &= a_0 + \sum_{j=1}^{n} a_j \prod_{k=0}^{j-1} (x - x_k). \end{aligned} \tag{3.6}$$

The coefficients a_j in (3.6) are given by

$$\begin{aligned} a_0 &= f(x_0), \\ a_j &= \frac{f(x_j) - Q_{j-1}(x_j)}{\prod_{k=0}^{j-1}(x_j - x_k)}, \qquad 1 \leqslant j \leqslant n. \end{aligned} \tag{3.7}$$

We refer to the interpolation polynomial when written in the form (3.6)–(3.7) as **the Newton form of the interpolation polynomial**. As we shall see below, there are various ways of writing the interpolation polynomial. The uniqueness of the interpolation polynomial as guaranteed by Theorem 3.1 implies that we will only be rewriting the same polynomial in different ways.

Example 3.2
The Newton form of the polynomial that interpolates $(x_0, f(x_0))$ and $(x_1, f(x_1))$ is

$$Q_1(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0).$$

Example 3.3
The Newton form of the polynomial that interpolates the three points $(x_0, f(x_0))$, $(x_1, f(x_1))$, and $(x_2, f(x_2))$ is

$$Q_2(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) + \frac{f(x_2) - \left[f(x_0) + \frac{f(x_1)-f(x_0)}{x_1-x_0}(x_2 - x_0)\right]}{(x_2 - x_0)(x_2 - x_1)}(x - x_0)(x - x_1).$$

3.4 The Interpolation Problem and the Vandermonde Determinant

An alternative approach to the interpolation problem is to consider directly a polynomial of the form

$$Q_n(x) = \sum_{k=0}^{n} b_k x^k, \tag{3.8}$$

and require that the following interpolation conditions are satisfied

$$Q_n(x_j) = f(x_j), \qquad 0 \leqslant j \leqslant n. \tag{3.9}$$

In view of Theorem 3.1 we already know that this problem has a unique solution, so we should be able to compute the coefficients of the polynomial directly from (3.8). Indeed, the interpolation conditions, (3.9), imply that the following equations should hold:

$$b_0 + b_1 x_j + \ldots + b_n x_j^n = f(x_j), \qquad j = 0, \ldots, n. \tag{3.10}$$

In matrix form, (3.10) can be rewritten as

$$\begin{pmatrix} 1 & x_0 & \ldots & x_0^n \\ 1 & x_1 & \ldots & x_1^n \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \ldots & x_n^n \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix}. \tag{3.11}$$

In order for the system (3.11) to have a *unique* solution, it has to be nonsingular. This means, e.g., that the determinant of its coefficients matrix must not vanish, i.e.

$$\begin{vmatrix} 1 & x_0 & \ldots & x_0^n \\ 1 & x_1 & \ldots & x_1^n \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \ldots & x_n^n \end{vmatrix} \neq 0. \tag{3.12}$$

The determinant (3.12), is known as **the Vandermonde determinant**. In Lemma 3.4 we will show that the Vandermonde determinant equals to the product of terms of the form $x_i - x_j$ for $i > j$. Since we assume that the points $x_0, \ldots, x_n$ are distinct, the determinant in (3.12) is indeed non zero. Hence, the system (3.11) has a solution that is also unique, which confirms what we already know according to Theorem 3.1.

Lemma 3.4

$$\begin{vmatrix} 1 & x_0 & \dots & x_0^n \\ 1 & x_1 & \dots & x_1^n \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \dots & x_n^n \end{vmatrix} = \prod_{i>j}(x_i - x_j). \tag{3.13}$$

Proof. We will prove (3.13) by induction. First we verify that the result holds in the 2×2 case. Indeed,

$$\begin{vmatrix} 1 & x_0 \\ 1 & x_1 \end{vmatrix} = x_1 - x_0.$$

We now assume that the result holds for $n-1$ and consider n. We note that the index n corresponds to a matrix of dimensions $(n+1) \times (n+1)$, hence our induction assumption is that (3.13) holds for any Vandermonde determinant of dimension $n \times n$. We subtract the first row from all other rows, and expand the determinant along the first column:

$$\begin{vmatrix} 1 & x_0 & \dots & x_0^n \\ 1 & x_1 & \dots & x_1^n \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \dots & x_n^n \end{vmatrix} = \begin{vmatrix} 1 & x_0 & \dots & x_0^n \\ 0 & x_1 - x_0 & \dots & x_1^n - x_0^n \\ \vdots & \vdots & & \vdots \\ 0 & x_n - x_0 & \dots & x_n^n - x_0^n \end{vmatrix} = \begin{vmatrix} x_1 - x_0 & \dots & x_1^n - x_0^n \\ \vdots & & \vdots \\ x_n - x_0 & \dots & x_n^n - x_0^n \end{vmatrix}$$

For every row k we factor out a term $x_k - x_0$:

$$\begin{vmatrix} x_1 - x_0 & \dots & x_1^n - x_0^n \\ \vdots & & \vdots \\ x_n - x_0 & \dots & x_n^n - x_0^n \end{vmatrix} = \prod_{k=1}^{n}(x_k - x_0) \begin{vmatrix} 1 & x_1 + x_0 & \dots & \sum_{i=0}^{n-1} x_1^{n-1-i} x_0^i \\ 1 & x_2 + x_0 & \dots & \sum_{i=0}^{n-1} x_2^{n-1-i} x_0^i \\ \vdots & \vdots & & \vdots \\ 1 & x_n + x_0 & \dots & \sum_{i=0}^{n-1} x_n^{n-1-i} x_0^i \end{vmatrix}$$

Here, we used the expansion

$$x_1^n - x_0^n = (x_1 - x_0)(x_1^{n-1} + x_1^{n-2}x_0 + x_1^{n-3}x_0^2 + \ldots + x_0^{n-1}),$$

for the first row, and similar expansions for all other rows. For every column l, starting from the second one, subtracting the sum of x_0^i times column i (summing only over "previous" columns, i.e., columns i with $i < l$), we end up with

$$\prod_{k=1}^{n}(x_k - x_0) \begin{vmatrix} 1 & x_1 & \dots & x_1^{n-1} \\ 1 & x_2 & \dots & x_2^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \dots & x_n^{n-1} \end{vmatrix}. \tag{3.14}$$

Since now we have on the RHS of (3.14) a Vandermonde determinant of dimension $n \times n$, we can use the induction to conclude with the desired result. ■

3.5 The Lagrange Form of the Interpolation Polynomial

The form of the interpolation polynomial that we used in (3.8) assumed a linear combination of polynomials of degrees $0, \ldots, n$, in which the coefficients were unknown. In this section we take a different approach and assume that the interpolation polynomial is given as a linear combination of $n+1$ polynomials of degree n. This time, we set the coefficients as the interpolated values, $\{f(x_j)\}_{j=0}^n$, while the unknowns are the polynomials. We thus let

$$Q_n(x) = \sum_{j=0}^n f(x_j) l_j^n(x), \tag{3.15}$$

where $l_j^n(x)$ are $n+1$ polynomials of degree $\leqslant n$. We use two indices in these polynomials: the subscript j enumerates $l_j^n(x)$ from 0 to n and the superscript n is used to remind us that the degree of $l_j^n(x)$ is n. Note that in this particular case, the polynomials $l_j^n(x)$ are precisely of degree n (and not $\leqslant n$). However, $Q_n(x)$, given by (3.15) may have a lower degree. In either case, the degree of $Q_n(x)$ is n at the most. We now require that $Q_n(x)$ satisfies the interpolation conditions

$$Q_n(x_i) = f(x_i), \qquad 0 \leqslant i \leqslant n. \tag{3.16}$$

By substituting x_i for x in (3.15) we have

$$Q_n(x_i) = \sum_{j=0}^n f(x_j) l_j^n(x_i), \qquad 0 \leqslant i \leqslant n.$$

In view of (3.16) we may conclude that $l_j^n(x)$ must satisfy

$$l_j^n(x_i) = \delta_{ij}, \qquad i, j = 0, \ldots, n, \tag{3.17}$$

where δ_{ij} is the Krönecker delta, defined as

$$\delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

Each polynomial $l_j^n(x)$ has $n+1$ unknown coefficients. The conditions (3.17) provide exactly $n+1$ equations that the polynomials $l_j^n(x)$ must satisfy and these equations can be solved in order to determine all $l_j^n(x)$'s. Fortunately there is a shortcut. An obvious way of constructing polynomials $l_j^n(x)$ of degree $\leqslant n$ that satisfy (3.17) is the following:

$$l_j^n(x) = \frac{(x-x_0) \cdot \ldots \cdot (x-x_{j-1})(x-x_{j+1}) \cdot \ldots \cdot (x-x_n)}{(x_j-x_0) \cdot \ldots \cdot (x_j-x_{j-1})(x_j-x_{j+1}) \cdot \ldots \cdot (x_j-x_n)}, \quad 0 \leqslant j \leqslant n. \tag{3.18}$$

The uniqueness of the interpolating polynomial of degree $\leqslant n$ given $n+1$ distinct interpolation points implies that the polynomials $l_j^n(x)$ given by (3.17) are the only polynomials of degree $\leqslant n$ that satisfy (3.17).

Note that the denominator in (3.18) does not vanish since we assume that all interpolation points are distinct. **The Lagrange form of the interpolation polynomial** is the polynomial $Q_n(x)$ given by (3.15), where the polynomials $l_j^n(x)$ of degree $\leqslant n$ are given by (3.18). A compact form of rewriting (3.18) using the product notation is

$$l_j^n(x) = \frac{\prod\limits_{\substack{i=0\\ i\neq j}}^{n}(x-x_i)}{\prod\limits_{\substack{i=0\\ i\neq j}}^{n}(x_j-x_i)}, \qquad j=0,\ldots,n. \tag{3.19}$$

Example 3.5

We are interested in finding the Lagrange form of the interpolation polynomial that interpolates two points: $(x_0, f(x_0))$ and $(x_1, f(x_1))$. We know that the unique interpolation polynomial through these two points is the line that connects the two points. Such a line can be written in many different forms. In order to obtain the Lagrange form we let

$$l_0^1(x) = \frac{x-x_1}{x_0-x_1}, \qquad l_1^1(x) = \frac{x-x_0}{x_1-x_0}.$$

The desired polynomial is therefore given by the familiar formula

$$Q_1(x) = f(x_0)l_0^1(x) + f(x_1)l_1^1(x) = f(x_0)\frac{x-x_1}{x_0-x_1} + f(x_1)\frac{x-x_0}{x_1-x_0}.$$

Example 3.6

This time we are looking for the Lagrange form of the interpolation polynomial, $Q_2(x)$, that interpolates three points: $(x_0, f(x_0))$, $(x_1, f(x_1))$, $(x_2, f(x_2))$. Unfortunately, the Lagrange form of the interpolation polynomial does not let us use the interpolation polynomial through the first two points, $Q_1(x)$, as a building block for $Q_2(x)$. This means that we have to compute all the polynomials $l_j^n(x)$ from scratch. We start with

$$\begin{aligned}
l_0^2(x) &= \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)},\\
l_1^2(x) &= \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)},\\
l_2^2(x) &= \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}.
\end{aligned}$$

The interpolation polynomial is therefore given by

$$\begin{aligned}Q_2(x) &= f(x_0)l_0^2(x) + f(x_1)l_1^2(x) + f(x_2)l_2^2(x)\\ &= f(x_0)\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + f(x_1)\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + f(x_2)\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}.\end{aligned}$$

It is easy to verify that indeed $Q_2(x_j) = f(x_j)$ for $j = 0, 1, 2$, as desired.

Remarks.

1. One instance where the Lagrange form of the interpolation polynomial may seem to be advantageous when compared with the Newton form is when there is a need to solve several interpolation problems, all given at the same interpolation points $x_0, \ldots x_n$ but with different values $f(x_0), \ldots, f(x_n)$. In this case, the polynomials $l_j^n(x)$ are identical for all problems since they depend only on the points but not on the values of the function at these points. Therefore, they have to be constructed only once.

2. An alternative form for $l_j^n(x)$ can be obtained in the following way. Define the polynomials $w_n(x)$ of degree $n+1$ by

$$w_n(x) = \prod_{i=0}^{n}(x - x_i).$$

   Then it its derivative is

$$w_n'(x) = \sum_{j=0}^{n}\prod_{\substack{i=0\\ i\neq j}}^{n}(x - x_i). \tag{3.20}$$

   When $w_x'(x)$ is evaluated at an interpolation point, x_j, there is only one term in the sum in (3.20) that does not vanish:

$$w_n'(x_j) = \prod_{\substack{i=0\\ i\neq j}}^{n}(x_j - x_i).$$

   Hence, in view of (3.19), $l_j^n(x)$ can be rewritten as

$$l_j^n(x) = \frac{w_n(x)}{(x-x_j)w_n'(x_j)}, \qquad 0 \leqslant j \leqslant n. \tag{3.21}$$

3. For future reference we note that the coefficient of x^n in the interpolation polynomial $Q_n(x)$ is

$$\sum_{j=0}^{n}\frac{f(x_j)}{\prod_{\substack{k=0\\ k\neq j}}^{n}(x_j - x_k)}. \tag{3.22}$$

For example, the coefficient of x in $Q_1(x)$ in Example 3.5 is

$$\frac{f(x_0)}{x_0 - x_1} + \frac{f(x_1)}{x_1 - x_0}.$$

3.6 Divided Differences

We recall that Newton's form of the interpolation polynomial is given by (see (3.6)–(3.7))

$$Q_n(x) = a_0 + a_1(x - x_0) + \ldots + a_n(x - x_0) \cdot \ldots \cdot (x - x_{n-1}),$$

with $a_0 = f(x_0)$ and

$$a_j = \frac{f(x_j) - Q_{j-1}(x_j)}{\prod_{k=0}^{j-1}(x_j - x_k)}, \qquad 1 \leqslant j \leqslant n.$$

From now on, we will refer to the coefficient, a_j, as **the j^{th}-order divided difference**. The j^{th}-order divided difference, a_j, is based on the points $x_0, \ldots, x_j$ and on the values of the function at these points $f(x_0), \ldots, f(x_j)$. To emphasize this dependence, we use the following notation:

$$a_j = f[x_0, \ldots, x_j], \qquad 1 \leqslant j \leqslant n. \tag{3.23}$$

We also denote the zeroth-order divided difference as

$$a_0 = f[x_0],$$

where

$$f[x_0] = f(x_0).$$

Using the divided differences notation (3.23), the Newton form of the interpolation polynomial becomes

$$Q_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + \ldots + f[x_0, \ldots x_n] \prod_{k=0}^{n-1}(x - x_k). \tag{3.24}$$

There is a simple recursive way of computing the j^{th}-order divided difference from divided differences of lower order, as shown by the following lemma:

Lemma 3.7 *The divided differences satisfy:*

$$f[x_0, \ldots x_n] = \frac{f[x_1, \ldots x_n] - f[x_0, \ldots x_{n-1}]}{x_n - x_0}. \tag{3.25}$$

Proof. For any k, we denote by $Q_k(x)$, a polynomial of degree $\leqslant k$, that interpolates $f(x)$ at $x_0, \ldots, x_k$, i.e.,

$$Q_k(x_j) = f(x_j), \qquad 0 \leqslant j \leqslant k.$$

We now consider the unique polynomial $P(x)$ of degree $\leqslant n-1$ that interpolates $f(x)$ at $x_1, \ldots, x_n$, and claim that

$$Q_n(x) = P(x) + \frac{x - x_n}{x_n - x_0}[P(x) - Q_{n-1}(x)]. \tag{3.26}$$

In order to verify this equality, we note that for $i = 1, \ldots, n-1$, $P(x_i) = Q_{n-1}(x_i)$ so that

$$RHS(x_i) = P(x_i) = f(x_i).$$

At x_n, $RHS(x_n) = P(x_n) = f(x_n)$. Finally, at x_0,

$$RHS(x_0) = P(x_0) + \frac{x_0 - x_n}{x_n - x_0}\,[P(x_0) - Q_{n-1}(x_0)] = Q_{n-1}(x_0) = f(x_0).$$

Hence, the RHS of (3.26) interpolates $f(x)$ at the $n+1$ points $x_0, \ldots, x_n$, which is also true for $Q_n(x)$ due to its definition. Since the RHS and the LHS in equation (3.26) are both polynomials of degree $\leqslant n$, the uniqueness of the interpolating polynomial (in this case through $n+1$ points) implies the equality in (3.26).
Once we established the equality in (3.26) we can compare the coefficients of the monomials on both sides of the equation. The coefficient of x^n on the left-hand-side of (3.26) is $f[x_0, \ldots, x_n]$. The coefficient of x^{n-1} in $P(x)$ is $f[x_1, \ldots, x_n]$ and the coefficient of x^{n-1} in $Q_{n-1}(x)$ is $f[x_0, \ldots, x_{n-1}]$. Hence, the coefficient of x^n on the right-hand-side of (3.26) is

$$\frac{1}{x_n - x_0}(f[x_1, \ldots, x_n] - f[x_0, \ldots, x_{n-1}]),$$

which means that

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

Remark. In some books, instead of defining the divided difference in such a way that they satisfy (3.25), the divided differences are defined by the formula

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

If this is the case, all our results on divided differences should be adjusted accordingly as to account for the missing factor in the denominator.

Example 3.8
The second-order divided difference is

$$f[x_0,x_1,x_2]=\frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0}=\frac{\frac{f(x_2)-f(x_1)}{x_2-x_1}-\frac{f(x_1)-f(x_0)}{x_1-x_0}}{x_2-x_0}.$$

Hence, the unique polynomial that interpolates $(x_0,f(x_0))$, $(x_1,f(x_1))$, and $(x_2,f(x_2))$ is

$$\begin{aligned}Q_2(x)&=f[x_0]+f[x_0,x_1](x-x_0)+f[x_0,x_1,x_2](x-x_0)(x-x_1)\\&=f(x_0)+\frac{f(x_1)-f(x_0)}{x_1-x_0}(x-x_0)+\frac{\frac{f(x_2)-f(x_1)}{x_2-x_1}-\frac{f(x_1)-f(x_0)}{x_1-x_0}}{x_2-x_0}(x-x_0)(x-x_1).\end{aligned}$$

For example, if we want to find the polynomial of degree $\leqslant 2$ that interpolates $(-1,9)$, $(0,5)$, and $(1,3)$, we have

$$f(-1)=9,$$

$$f[-1,0]=\frac{5-9}{0-(-1)}=-4,\qquad f[0,1]=\frac{3-5}{1-0}=-2,$$

$$f[-1,0,1]=\frac{f[0,1]-f[-1,0]}{1-(-1)}=\frac{-2+4}{2}=1.$$

so that

$$Q_2(x)=9-4(x+1)+(x+1)x=5-3x+x^2.$$

The relations between the divided differences are schematically portrayed in Table 3.1 (up to third-order). We note that the divided differences that are being used as the coefficients in the interpolation polynomial are those that are located in the top of every column. The recursive structure of the divided differences implies that it is required to compute all the low order coefficients in the table in order to get the high-order ones.

One important property of any divided difference is that it is a symmetric function of its arguments. This means that if we assume that $y_0,\ldots,y_n$ is any permutation of $x_0,\ldots,x_n$, then

$$f[y_0,\ldots,y_n]=f[x_0,\ldots,x_n].$$

This property can be clearly explained by recalling that $f[x_0,\ldots,x_n]$ plays the role of the coefficient of x^n in the polynomial that interpolates $f(x)$ at $x_0,\ldots,x_n$. At the same time, $f[y_0,\ldots,y_n]$ is the coefficient of x^n at the polynomial that interpolates $f(x)$ at the same points. Since the interpolation polynomial is unique for any given data set, the order of the points does not matter, and hence these two coefficients must be identical.

$$
\begin{array}{cccccccc}
x_0 & f(x_0) & & & & & & \\
 & & \searrow & & & & & \\
 & & & f[x_0,x_1] & & & & \\
 & & \nearrow & & \searrow & & & \\
x_1 & f(x_1) & & & & f[x_0,x_1,x_2] & & \\
 & & \searrow & & \nearrow & & \searrow & \\
 & & & f[x_1,x_2] & & & & f[x_0,x_1,x_2,x_3] \\
 & & \nearrow & & \searrow & & \nearrow & \\
x_2 & f(x_2) & & & & f[x_1,x_2,x_3] & & \\
 & & \searrow & & \nearrow & & & \\
 & & & f[x_2,x_3] & & & & \\
 & & \nearrow & & & & & \\
x_3 & f(x_3) & & & & & &
\end{array}
$$

Table 3.1: Divided Differences

3.7 The Error in Polynomial Interpolation

Our goal in this section is to provide estimates on the "error" we make when interpolating data that is taken from sampling an underlying function $f(x)$. While the interpolant and the function agree with each other at the interpolation points, there is, in general, no reason to expect them to be close to each other elsewhere. Nevertheless, we can estimate the difference between them, a difference which we refer to as the **interpolation error**. We let Π_n denote the space of polynomials of degree $\leqslant n$, and let $C^{n+1}[a,b]$ denote the space of functions that have $n+1$ continuous derivatives on the interval $[a,b]$.

Theorem 3.9 *Let $f(x) \in C^{n+1}[a,b]$. Let $Q_n(x) \in \Pi_n$ such that it interpolates $f(x)$ at the $n+1$ distinct points $x_0,\ldots,x_n \in [a,b]$. Then $\forall x \in [a,b]$, $\exists \xi_n \in (a,b)$ such that*

$$
f(x) - Q_n(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi_n) \prod_{j=0}^{n} (x - x_j). \tag{3.27}
$$

Proof. Fix a point $x \in [a,b]$. If x is one of the interpolation points $x_0,\ldots,x_n$, then the left-hand-side and the right-hand-side of (3.27) are both zero, and the result holds trivially. We therefore assume that $x \neq x_j$ $0 \leqslant j \leqslant n$, and let

$$
w(x) = \prod_{j=0}^{n} (x - x_j).
$$

We now let

$$
F(y) = f(y) - Q_n(y) - \lambda w(y),
$$

where λ is chosen as to guarantee that $F(x) = 0$, i.e.,

$$\lambda = \frac{f(x) - Q_n(x)}{w(x)}.$$

Since the interpolation points $x_0, \ldots, x_n$ and x are distinct, $w(x)$ does not vanish and λ is well defined. We now note that since $f \in C^{n+1}[a,b]$ and since Q_n and w are polynomials, then also $F \in C^{n+1}[a,b]$. In addition, F vanishes at $n+2$ points: $x_0, \ldots, x_n$ and x. According to Rolle's theorem, F' has at least $n+1$ distinct zeros in (a,b), F'' has at least n distinct zeros in (a,b), and similarly, $F^{(n+1)}$ has at least one zero in (a,b), which we denote by ξ_n. We have

$$\begin{aligned} 0 &= F^{(n+1)}(\xi_n) = f^{(n+1)}(\xi_n) - Q_n^{(n+1)}(\xi_n) - \lambda(x) w^{(n+1)}(\xi_n) \\ &= f^{(n+1)}(\xi_n) - \frac{f(x) - Q_n(x)}{w(x)}(n+1)! \end{aligned} \tag{3.28}$$

Here, we used the fact that the leading term of $w(x)$ is x^{n+1}, which guarantees that its $(n+1)^{\text{th}}$ derivative is

$$w^{(n+1)}(x) = (n+1)! \tag{3.29}$$

Reordering the terms in (3.28) we conclude with

$$f(x) - Q_n(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi_n) w(x). \quad \blacksquare$$

In addition to the interpretation of the divided difference of order n as the coefficient of x^n in some interpolation polynomial, it can also be characterized in another important way. Consider, e.g., the first-order divided difference

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

Since the order of the points does not change the value of the divided difference, we can assume, without any loss of generality, that $x_0 < x_1$. If we assume, in addition, that $f(x)$ is continuously differentiable in the interval $[x_0, x_1]$, then this divided difference equals to the derivative of $f(x)$ at an intermediate point, i.e.,

$$f[x_0, x_1] = f'(\xi), \qquad \xi \in (x_0, x_1).$$

In other words, the first-order divided difference can be viewed as an approximation of the first derivative of $f(x)$ in the interval. It is important to note that while this interpretation is based on additional smoothness requirements from $f(x)$ (i.e. its being differentiable), the divided differences are well defined also for non-differentiable functions.

This notion can be extended to divided differences of higher order as stated by the following lemma.

Lemma 3.10 *Let $x, x_0, \ldots, x_{n-1}$ be $n+1$ distinct points. Let $a = \min(x, x_0, \ldots, x_{n-1})$ and $b = \max(x, x_0, \ldots, x_{n-1})$. Assume that $f(y)$ has a continuous derivative of order n in the interval (a, b). Then*

$$f[x_0, \ldots, x_{n-1}, x] = \frac{f^{(n)}(\xi)}{n!}, \tag{3.30}$$

where $\xi \in (a, b)$.

Proof. Let $Q_n(y)$ interpolate $f(y)$ at $x_0, \ldots, x_{n-1}, x$. Then according to the construction of the Newton form of the interpolation polynomial (3.24), we know that

$$Q_n(y) = Q_{n-1}(y) + f[x_0, \ldots, x_{n-1}, x] \prod_{j=0}^{n-1} (y - x_j).$$

Since $Q_n(y)$ interpolated $f(y)$ at x, we have

$$f(x) = Q_{n-1}(x) + f[x_0, \ldots, x_{n-1}, x] \prod_{j=0}^{n-1} (x - x_j).$$

By Theorem 3.9 we know that the interpolation error is given by

$$f(x) - Q_{n-1}(x) = \frac{1}{n!} f^{(n)}(\xi_{n-1}) \prod_{j=0}^{n-1} (x - x_j),$$

which implies the result (3.30). ■

Remark. In equation (3.30), we could as well think of the interpolation point x as any other interpolation point, and name it, e.g., x_n. In this case, the equation (3.30) takes the somewhat more natural form of

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

In other words, the n^{th}-order divided difference is an n^{th}-derivative of the function $f(x)$ at an intermediate point, assuming that the function has n continuous derivatives. Similarly to the first-order divided difference, we would like to emphasize that the n^{th}-order divided difference is also well defined in cases where the function is not as smooth as required in the theorem, though if this is the case, we can no longer consider this divided difference to represent a n^{th}-order derivative of the function.

3.8 Interpolation at the Chebyshev Points

In the entire discussion up to now, we assumed that the interpolation points are given. There may be cases where one may have the flexibility of choosing the interpolation points. If this is the case, it would be reasonable to use this degree of freedom to minimize the interpolation error.

We recall that if we are interpolating values of a function $f(x)$ that has n continuous derivatives, the interpolation error is of the form

$$f(x) - Q_n(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi_n) \prod_{j=0}^{n} (x - x_j). \tag{3.31}$$

Here, $Q_n(x)$ is the interpolating polynomial and ξ_n is an intermediate point in the interval of interest (see (3.27)).

It is important to note that the interpolation points influence two terms on the right-hand-side of (3.31). The obvious one is the product

$$\prod_{j=0}^{n} (x - x_j). \tag{3.32}$$

The second term that depends on the interpolation points is $f^{(n+1)}(\xi_n)$ since the value of the intermediate point ξ_n depends on $\{x_j\}$. Due to the implicit dependence of ξ_n on the interpolation points, minimizing the interpolation error is not an easy task. We will return to this "full" problem later on in the context of the minimax approximation problem. For the time being, we are going to focus on a simpler problem, namely, how to choose the interpolation points $x_0, \ldots, x_n$ such that the product (3.32) is minimized. The solution of this problem is the topic of this section. Once again, we would like to emphasize that a solution of this problem does not (in general) provide an optimal choice of interpolation points that minimizes the interpolation error. All that it guarantees is that the product part of the interpolation error is minimal.

The tool that we are going to use is the Chebyshev polynomials. The solution of the problem will be to choose the interpolation points as the Chebyshev points. We will first introduce the Chebyshev polynomials and the Chebyshev points and then explain why interpolating at these points minimizes (3.32).

The **Chebyshev polynomials** can be defined using the following recursion relation:

$$\begin{cases} T_0(x) = 1, \\ T_1(x) = x, \\ T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad n \geqslant 1. \end{cases} \tag{3.33}$$

For example, $T_2(x) = 2xT_1(x) - T_0(x) = 2x^2 - 1$, and $T_3(x) = 4x^3 - 3x$. The polynomials $T_1(x), T_2(x)$ and $T_3(x)$ are plotted in Figure 3.2.

Instead of writing the recursion formula, (3.33), it is possible to write an explicit formula for the Chebyshev polynomials:

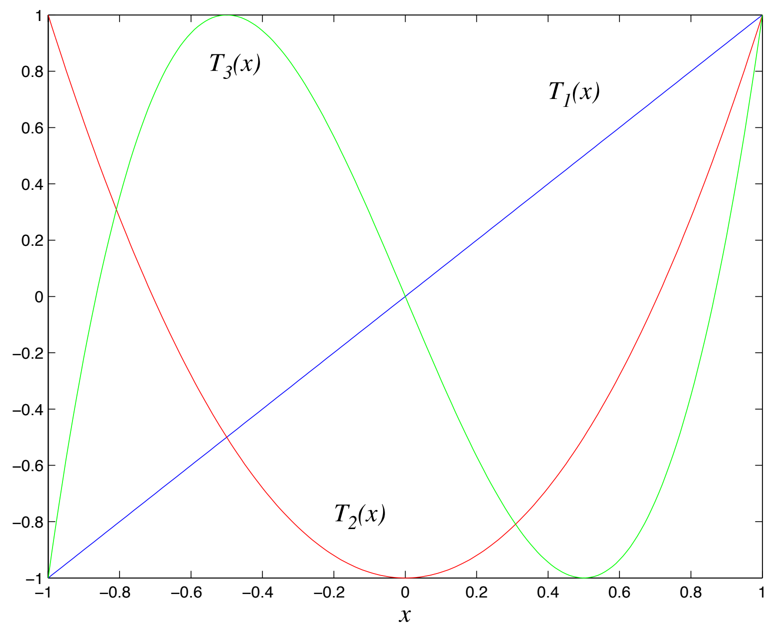

Figure 3.2: The Chebyshev polynomials $T_1(x), T_2(x)$ and $T_3(x)$

Lemma 3.11 *For $x \in [-1,1]$,*

$$T_n(x) = \cos(n \cos^{-1} x), \qquad n \geqslant 0. \tag{3.34}$$

Proof. Standard trigonometric identities imply that

$$\begin{aligned} \cos(n+1)\theta &= \cos\theta \cos n\theta - \sin\theta \sin n\theta, \\ \cos(n-1)\theta &= \cos\theta \cos n\theta + \sin\theta \sin n\theta. \end{aligned}$$

Hence

$$\cos(n+1)\theta = 2\cos\theta\cos n\theta - \cos(n-1)\theta. \tag{3.35}$$

We now let $\theta = \cos^{-1} x$, i.e., $x = \cos\theta$, and define

$$t_n(x) = \cos(n \cos^{-1} x) = \cos(n\theta).$$

Then by (3.35)

$$\begin{cases} t_0(x) = 1, \\ t_1(x) = x, \\ t_{n+1}(x) = 2x t_n(x) - t_{n-1}(x), \quad n \geqslant 1. \end{cases}$$

Hence $t_n(x) = T_n(x)$. ■

What is so special about the Chebyshev polynomials, and what is the connection between these polynomials and minimizing the interpolation error? We are about to answer these questions, but before doing so, there is one more issue that we must clarify.

We define a **monic polynomial** as a polynomial for which the coefficient of the leading term is one, i.e., a polynomial of degree n is monic, if it is of the form

$$x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0.$$

Note that Chebyshev polynomials are not monic: the definition (3.33) implies that the Chebyshev polynomial of degree n is of the form

$$T_n(x) = 2^{n-1}x^n + \ldots$$

This means that $T_n(x)$ divided by 2^{n-1} is monic, i.e.,

$$2^{1-n}T_n(x) = x^n + \ldots$$

A general result about monic polynomials is given by the following theorem

Theorem 3.12 *If $p_n(x)$ is a monic polynomial of degree n, then*

$$\max_{-1\leqslant x\leqslant 1} |p_n(x)| \geqslant 2^{1-n}. \tag{3.36}$$

Proof. We prove (3.36) by contradiction. Suppose that

$$|p_n(x)| < 2^{1-n}, \qquad |x| \leqslant 1.$$

Let

$$q_n(x) = 2^{1-n}T_n(x),$$

and let x_j be the following $n+1$ points

$$x_j = \cos\left(\frac{j\pi}{n}\right), \qquad 0 \leqslant j \leqslant n.$$

Since

$$T_n\left(\cos\frac{j\pi}{n}\right) = (-1)^j,$$

we have

$$(-1)^j q_n(x_j) = 2^{1-n}.$$

Hence

$$(-1)^j p_n(x_j) \leqslant |p_n(x_j)| < 2^{1-n} = (-1)^j q_n(x_j).$$

This means that

$$(-1)^j (q_n(x_j) - p_n(x_j)) > 0, \qquad 0 \leqslant j \leqslant n.$$

Hence, the polynomial $(q_n - p_n)(x)$ oscillates $(n+1)$ times in the interval $[-1,1]$, which means that $(q_n - p_n)(x)$ has at least n distinct roots in the interval. However, $p_n(x)$ and $q_n(x)$ are both monic polynomials which means that their difference is a polynomial of degree $n-1$ at most. Such a polynomial cannot have more than $n-1$ distinct roots, which leads to a contradiction. Note that $p_n - q_n$ cannot be the zero polynomial because that will imply that $p_n(x)$ and $q_n(x)$ are identical which again is not possible due to the assumptions on their maximum values. ■

We are now ready to use Theorem 3.12 to figure out how to reduce the interpolation error. We know by Theorem 3.9 that if the interpolation points $x_0, \ldots, x_n \in [-1,1]$, then there exists $\xi_n \in (-1,1)$ such that the distance between the function whose values we interpolate, $f(x)$, and the interpolation polynomial, $Q_n(x)$, is

$$\max_{|x|\leqslant 1} |f(x) - Q_n(x)| \leqslant \frac{1}{(n+1)!} \max_{|x|\leqslant 1} |f^{(n+1)}(\xi_n)| \max_{|x|\leqslant 1} \left| \prod_{j=0}^{n} (x - x_j) \right|.$$

We are interested in minimizing

$$\max_{|x|\leqslant 1} \left| \prod_{j=0}^{n} (x - x_j) \right|.$$

We note that $\prod_{j=0}^{n}(x - x_j)$ is a monic polynomial of degree $n+1$ and hence by Theorem 3.12

$$\max_{|x|\leqslant 1} \left| \prod_{j=0}^{n} (x - x_j) \right| \geqslant 2^{-n}.$$

The minimal value of 2^{-n} can be actually obtained if we set

$$2^{-n} T_{n+1}(x) = \prod_{j=0}^{n} (x - x_j),$$

which is equivalent to choosing x_j as the roots of the Chebyshev polynomial $T_{n+1}(x)$. Here, we have used the obvious fact that $|T_n(x)| \leqslant 1$.

What are the roots of the Chebyshev polynomial $T_{n+1}(x)$? By Lemma 3.11

$$T_{n+1}(x) = \cos((n+1)\cos^{-1} x).$$

The roots of $T_{n+1}(x)$, $x_0, \ldots, x_n$, are therefore obtained if

$$(n+1)\cos^{-1}(x_j) = \left(j + \frac{1}{2}\right)\pi, \qquad 0 \leqslant j \leqslant n,$$

i.e., the $(n+1)$ roots of $T_{n+1}(x)$ are

$$x_j = \cos\left(\frac{2j+1}{2n+2}\pi\right), \qquad 0 \leqslant j \leqslant n. \tag{3.37}$$

The roots of the Chebyshev polynomials are sometimes referred to as **the Chebyshev points**. The formula (3.37) for the roots of the Chebyshev polynomial has the following geometrical interpretation. In order to find the roots of $T_n(x)$, define $\alpha = \pi/n$. Divide the upper half of the unit circle into $n+1$ parts such that the two side angles are $\alpha/2$ and the other angles are α. The Chebyshev points are then obtained by projecting these points on the x-axis. This procedure is demonstrated in Figure 3.3 for $T_4(x)$.

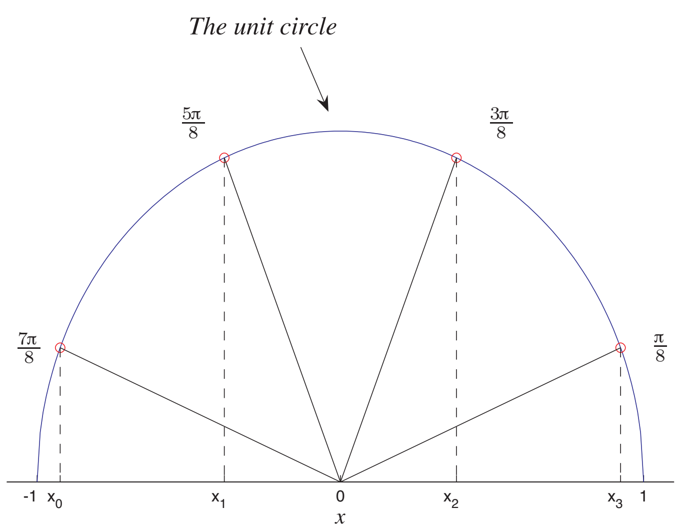

Figure 3.3: The roots of the Chebyshev polynomial $T_4(x)$, $x_0, \ldots, x_3$. Note that they become dense next to the boundary of the interval

The following theorem summarizes the discussion on interpolation at the Chebyshev points. It also provides an estimate of the error for this case.

Theorem 3.13 *Assume that $Q_n(x)$ interpolates $f(x)$ at $x_0, \ldots, x_n$. Assume also that these $(n+1)$ interpolation points are the $(n+1)$ roots of the Chebyshev polynomial of degree $n+1$, $T_{n+1}(x)$, i.e.,*

$$x_j = \cos\left(\frac{2j+1}{2n+2}\pi\right), \qquad 0 \leqslant j \leqslant n.$$

Then $\forall |x| \leqslant 1$,

$$|f(x) - Q_n(x)| \leqslant \frac{1}{2^n(n+1)!} \max_{|\xi|\leqslant 1} \left|f^{(n+1)}(\xi)\right|. \tag{3.38}$$

Example 3.14
Problem: Let $f(x) = \sin(\pi x)$ in the interval $[-1, 1]$. Find $Q_2(x)$ which interpolates $f(x)$ in the Chebyshev points. Estimate the error.

Solution: Since we are asked to find an interpolation polynomial of degree $\leqslant 2$, we need 3 interpolation points. We are also asked to interpolate at the Chebyshev points, and hence we first need to compute the 3 roots of the Chebyshev polynomial of degree 3,

$$T_3(x) = 4x^3 - 3x.$$

The roots of $T_3(x)$ can be easily found from $x(4x^2 - 3) = 0$, i.e.,

$$x_0 = -\frac{\sqrt{3}}{2}, \quad , x_1 = 0, \quad x_2 = \frac{\sqrt{3}}{2}.$$

The corresponding values of $f(x)$ at these interpolation points are

$$\begin{aligned} f(x_0) &= \sin\left(-\frac{\sqrt{3}}{2}\pi\right) \approx -0.4086, \\ f(x_1) &= 0, \\ f(x_2) &= \sin\left(\frac{\sqrt{3}}{2}\pi\right) \approx 0.4086. \end{aligned}$$

The first-order divided differences are

$$\begin{aligned} f[x_0, x_1] &= \frac{f(x_1) - f(x_0)}{x_1 - x_0} \approx 0.4718, \\ f[x_1, x_2] &= \frac{f(x_2) - f(x_1)}{x_2 - x_1} \approx 0.4718, \end{aligned}$$

and the second-order divided difference is

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = 0.$$

The interpolation polynomial is

$$Q_2(x) = f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \approx 0.4718x.$$

The original function $f(x)$ and the interpolant at the Chebyshev points, $Q_2(x)$, are plotted in Figure 3.4.
As of the error estimate, $\forall |x| \leqslant 1$,

$$|\sin \pi x - Q_2(x)| \leqslant \frac{1}{2^2 3!} \max_{|\xi| \leqslant 1} |(\sin \pi t)^{(3)}| \leqslant \frac{\pi^3}{2^2 3!} \leqslant 1.292$$

A brief examination of Figure 3.4 reveals that while this error estimate is correct, it is far from being sharp.

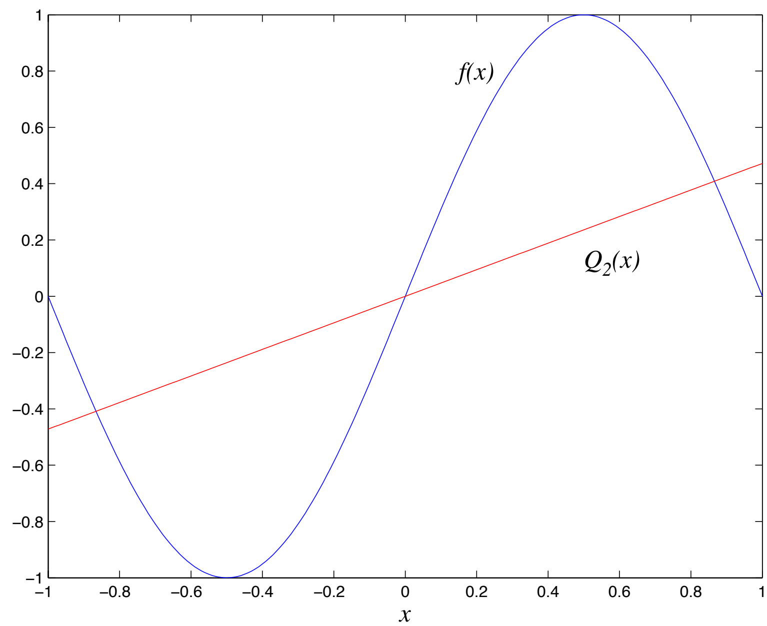

Figure 3.4: The function $f(x) = \sin(\pi(x))$ and the interpolation polynomial $Q_2(x)$ that interpolates $f(x)$ at the Chebyshev points. See Example 3.14.

Remark. In the more general case where the interpolation interval for the function $f(x)$ is $x \in [a, b]$, it is still possible to use the previous results by following the following steps: Start by converting the interpolation interval to $y \in [-1, 1]$:

$$x = \frac{(b - a)y + (a + b)}{2}.$$

This converts the interpolation problem for $f(x)$ on $[a, b]$ into an interpolation problem for $f(x) = g(x(y))$ in $y \in [-1, 1]$. The Chebyshev points in the interval $y \in [-1, 1]$ are the roots of the Chebyshev polynomial $T_{n+1}(x)$, i.e.,

$$y_j = \cos\left(\frac{2j+1}{2n+2}\pi\right), \qquad 0 \leqslant j \leqslant n.$$

The corresponding $n+1$ interpolation points in the interval $[a,b]$ are

$$x_j = \frac{(b-a)y_j + (a+b)}{2}, \qquad 0 \leqslant j \leqslant n.$$

In this case, the product term in the interpolation error is

$$\max_{x\in[a,b]}\left|\prod_{j=0}^{n}(x-x_j)\right| = \left|\frac{b-a}{2}\right|^{n+1}\max_{|y|\leqslant 1}\left|\prod_{j=0}^{n}(x-y_j)\right|,$$

and the interpolation error is given by

$$|f(x)-Q_n(x)| \leqslant \frac{1}{2^n(n+1)!}\left|\frac{b-a}{2}\right|^{n+1}\max_{\xi\in[a,b]}\left|f^{(n+1)}(\xi)\right|. \tag{3.39}$$

3.9 Hermite Interpolation

We now turn to a slightly different interpolation problem in which we assume that in addition to interpolating the values of the function at certain points, we are also interested in interpolating its derivatives. Interpolation that involves the derivatives is called **Hermite interpolation**. Such an interpolation problem is demonstrated in the following example:

Example 3.15
Problem: Find a polynomials $p(x)$ such that $p(1) = -1$, $p'(1) = -1$, and $p(0) = 1$.

Solution: Since three conditions have to be satisfied, we can use these conditions to determine three degrees of freedom, which means that it is reasonable to expect that these conditions uniquely determine a polynomial of degree $\leqslant 2$. We therefore let

$$p(x) = a_0 + a_1x + a_2x^2.$$

The conditions of the problem then imply that

$$\begin{cases} a_0 + a_1 + a_2 = -1, \\ a_1 + 2a_2 = -1, \\ a_0 = 1. \end{cases}$$

Hence, there is indeed a unique polynomial that satisfies the interpolation conditions and it is

$$p(x) = x^2 - 3x + 1.$$

In general, we may have to interpolate high-order derivatives and not only first-order derivatives. Also, we assume that for any point x_j in which we have to satisfy an interpolation condition of the form

$$p^{(l)}(x_j) = f(x_j),$$

(with $p^{(l)}$ being the l^{th}-order derivative of $p(x)$), we are also given all the values of the lower-order derivatives up to l as part of the interpolation requirements, i.e.,

$$p^{(i)}(x_j) = f^{(i)}(x_j), \qquad 0 \leqslant i \leqslant l.$$

If this is not the case, it may not be possible to find a unique interpolant as demonstrated in the following example.

Example 3.16
Problem: Find $p(x)$ such that $p'(0) = 1$ and $p'(1) = -1$.

Solution: Since we are asked to interpolate two conditions, we may expect them to uniquely determine a linear function, say

$$p(x) = a_0 + a_1 x.$$

However, both conditions specify the derivative of $p(x)$ at two distinct points to be of different values, which amounts to a contradicting information on the value of a_1. Hence, a linear polynomial cannot interpolate the data and we must consider higher-order polynomials. Unfortunately, a polynomial of order $\geqslant 2$ will no longer be unique because not enough information is given. Note that even if the prescribed values of the derivatives were identical, we will not have problems with the coefficient of the linear term a_1, but we will still not have enough information to determine the constant a_0.

A simple case that you are probably already familiar with is the **Taylor series**. When viewed from the point of view that we advocate in this section, one can consider the Taylor series as an interpolation problem in which one has to interpolate the value of the function and its first n derivatives at a given point, say x_0, i.e., the interpolation conditions are:

$$p^{(j)}(x_0) = f^{(j)}(x_0), \qquad 0 \leqslant j \leqslant n.$$

The unique solution of this problem in terms of a polynomial of degree $\leqslant n$ is

$$p(x) = f(x_0) + f'(x_0)(x - x_0) + \ldots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n = \sum_{j=0}^{n} \frac{f^{(j)}(x_0)}{j!}(x - x_0)^j,$$

which is the Taylor series of $f(x)$ expanded about $x = x_0$.

3.9.1 Divided differences with repetitions

We are now ready to consider the Hermite interpolation problem. The first form we study is the Newton form of the Hermite interpolation polynomial. We start by extending the definition of divided differences in such a way that they can handle derivatives. We already know that the first derivative is connected with the first-order divided difference by

$$f'(x_0) = \lim_{x \to x_0} \frac{f(x) - f(x_0)}{x - x_0} = \lim_{x \to x_0} f[x, x_0].$$

Hence, it is natural to extend the notion of divided differences by the following definition.

Definition 3.17 The first-order **divided difference with repetitions** is defined as

$$f[x_0, x_0] = f'(x_0). \tag{3.40}$$

In a similar way, we can extend the notion of divided differences to high-order derivatives as stated in the following lemma (which we leave without a proof).

Lemma 3.18 *Let $x_0 \leqslant x_1 \leqslant \ldots \leqslant x_n$. Then the divided differences satisfy*

$$f[x_0, \ldots x_n] = \begin{cases} \dfrac{f[x_1, \ldots, x_n] - f[x_0, \ldots, x_{n-1}]}{x_n - x_0}, & x_n \neq x_0, \\ \dfrac{f^{(n)}(x_0)}{n!}, & x_n = x_0. \end{cases} \tag{3.41}$$

We now consider the following Hermite interpolation problem: The interpolation points are $x_0, \ldots, x_l$ (which we assume are ordered from small to large). At each interpolation point x_j, we have to satisfy the interpolation conditions:

$$p^{(i)}(x_j) = f^{(i)}(x_j), \qquad 0 \leqslant i \leqslant m_j.$$

Here, m_j denotes the number of derivatives that we have to interpolate for each point x_j (with the standard notation that zero derivatives refers to the value of the function only). In general, the number of derivatives that we have to interpolate may change from point to point. The extended notion of divided differences allows us to write the solution to this problem in the following way:

We let n denote the total number of points including their multiplicities (that correspond to the number of derivatives we have to interpolate at each point), i.e.,

$$n = m_1 + m_2 + \ldots + m_l.$$

We then list all the points including their multiplicities (that correspond to the number of derivatives we have to interpolate). To simplify the notations we identify these points with a new ordered list of points y_i:

$$\{y_0, \ldots, y_{n-1}\} = \{\underbrace{x_0, \ldots, x_0}_{m_1}, \underbrace{x_1, \ldots, x_1}_{m_2}, \ldots, \underbrace{x_l, \ldots, x_l}_{m_l}\}.$$

The interpolation polynomial $p_{n-1}(x)$ is given by

$$p_{n-1}(x) = f[y_0] + \sum_{j=1}^{n-1} f[y_0, \ldots, y_j] \prod_{k=0}^{j-1} (x - y_k). \tag{3.42}$$

Whenever a point repeats in $f[y_0, \ldots, y_j]$, we interpret this divided difference in terms of the extended definition (3.41). In practice, there is no need to shift the notations to y's and we work directly with the original points. We demonstrate this interpolation procedure in the following example.

Example 3.19
Problem: Find an interpolation polynomial $p(x)$ that satisfies

$$\begin{cases} p(x_0) = f(x_0), \\ p(x_1) = f(x_1), \\ p'(x_1) = f'(x_1). \end{cases}$$

Solution: The interpolation polynomial $p(x)$ is

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

The divided differences:

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

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

Hence

$$p(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) + \frac{(x_1 - x_0) f'(x_1) - [f(x_1) - f(x_0)]}{(x_1 - x_0)^2}(x - x_0)(x - x_1).$$

3.9.2 The Lagrange form of the Hermite interpolant

In this section we are interested in writing the Lagrange form of the Hermite interpolant in the special case in which the nodes are $x_0, \ldots, x_n$ and the interpolation conditions are

$$p(x_i) = f(x_i), \quad p'(x_i) = f'(x_i), \quad 0 \leqslant i \leqslant n. \tag{3.43}$$

We look for an interpolant of the form

$$p(x) = \sum_{i=0}^{n} f(x_i) A_i(x) + \sum_{i=0}^{n} f'(x_i) B_i(x). \tag{3.44}$$

In order to satisfy the interpolation conditions (3.43), the polynomials $p(x)$ in (3.44) must satisfy the $2n+2$ conditions:

$$\begin{cases} A_i(x_j) = \delta_{ij}, \quad B_i(x_j) = 0, \\ \\ A_i'(x_j) = 0, \quad\;\; B_i'(x_j) = \delta_{ij}, \end{cases} \qquad i,j = 0,\ldots,n. \tag{3.45}$$

We thus expect to have a unique polynomial $p(x)$ that satisfies the constraints (3.45) assuming that we limit its degree to be $\leqslant 2n+1$.

It is convenient to start the construction with the functions we have used in the Lagrange form of the standard interpolation problem (Section 3.5). We already know that

$$l_i(x) = \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j},$$

satisfy $l_i(x_j) = \delta_{ij}$. In addition, for $i \neq j$,

$$l_i^2(x_j) = 0, \qquad (l_i^2(x_j))' = 0.$$

The degree of $l_i(x)$ is n, which means that the degree of $l_i^2(x)$ is $2n$. We will thus assume that the unknown polynomials $A_i(x)$ and $B_i(x)$ in (3.45) can be written as

$$\begin{cases} A_i(x) = r_i(x) l_i^2(x), \\ B_i(x) = s_i(x) l_i^2(x). \end{cases}$$

The functions $r_i(x)$ and $s_i(x)$ are both assumed to be linear, which implies that $\deg(A_i) = \deg(B_i) = 2n+1$, as desired. Now, according to (3.45)

$$\delta_{ij} = A_i(x_j) = r_i(x_j) l_i^2(x_j) = r_i(x_j)\delta_{ij}.$$

Hence

$$r_i(x_i) = 1. \tag{3.46}$$

Also,

$$0 = A_i'(x_j) = r_i'(x_j)[l_i(x_j)]^2 + 2r_i(x_j) l_i(x_J) l_i'(x_j) = r_i'(x_j)\delta_{ij} + 2r_i(x_j)\delta_{ij} l_i'(x_j),$$

and thus

$$r_i'(x_i) + 2l_i'(x_i) = 0. \tag{3.47}$$

Assuming that $r_i(x)$ is linear, $r_i(x) = ax + b$, equations (3.46),(3.47), imply that

$$a = -2l_i'(x_i), \qquad b = 1 + 2l_i'(x_i)x_i.$$

Therefore

$$A_i(x) = [1 + 2l_i'(x_i)(x_i - x)]l_i^2(x).$$

As of $B_i(x)$ in (3.44), the conditions (3.45) imply that

$$0 = B_i(x_j) = s_i(x_j)l_i^2(x_j) \quad \Longrightarrow \quad s_i(x_i) = 0, \tag{3.48}$$

and

$$\delta_{ij} = B_i'(x_j) = s_i'(x_j)l_i^2(x_j) + 2s_i(x_j)(l_i^2(x_j))' \quad \Longrightarrow \quad s_i'(x_i) = 1. \tag{3.49}$$

Combining (3.48) and (3.49), we obtain

$$s_i(x) = x - x_i,$$

so that

$$B_i(x) = (x - x_i)l_i^2(x).$$

To summarize, the Lagrange form of the Hermite interpolation polynomial is given by

$$p(x) = \sum_{i=0}^{n} f(x_i)[1 + 2l_i'(x_i)(x_i - x)]l_i^2(x) + \sum_{i=0}^{n} f'(x_i)(x - x_i)l_i^2(x). \tag{3.50}$$

The error in the Hermite interpolation (3.50) is given by the following theorem.

Theorem 3.20 *Let $x_0, \ldots, x_n$ be distinct nodes in $[a, b]$ and $f \in C^{2n+2}[a, b]$. If $p \in \Pi_{2n+1}$, such that $\forall 0 \leqslant i \leqslant n$,*

$$p(x_i) = f(x_i), \qquad p'(x_i) = f'(x_i),$$

then $\forall x \in [a, b]$, there exists $\xi \in (a, b)$ such that

$$f(x) - p(x) = \frac{f^{(2n+2)}(\xi)}{(2n + 2)!} \prod_{i=0}^{n} (x - x_i)^2. \tag{3.51}$$

Proof. The proof follows the same techniques we used in proving Theorem 3.9. If x is one of the interpolation points, the result trivially holds. We thus fix x as a non-interpolation point and define

$$w(y) = \prod_{i=0}^{n} (y - x_i)^2.$$

We also have

$$\phi(y) = f(y) - p(y) - \lambda w(y),$$

and select λ such that $\phi(x) = 0$, i.e.,

$$\lambda = \frac{f(x) - p(x)}{w(x)}.$$

ϕ has (at least) $n+2$ zeros in $[a, b]$: $(x, x_0, \ldots, x_n)$. By Rolle's theorem, we know that ϕ' has (at least) $n+1$ zeros that are different than $(x, x_0, \ldots, x_n)$. Also, ϕ' vanishes at $x_0, \ldots, x_n$, which means that ϕ' has at least $2n+2$ zeros in $[a, b]$.
Similarly, Rolle's theorem implies that ϕ'' has at least $2n+1$ zeros in (a, b), and by induction, $\phi^{(2n+2)}$ has at least one zero in (a, b), say ξ.
Hence

$$0 = \phi^{(2n+2)}(\xi) = f^{(2n+2)}(\xi) - p^{(2n+2)}(\xi) - \lambda w^{(2n+2)}(\xi).$$

Since the leading term in $w(y)$ is x^{2n+2}, $w^{(2n+2)}(\xi) = (2n+2)!$. Also, since $p(x) \in \Pi_{2n+1}$, $p^{(2n+2)}(\xi) = 0$. We recall that x was an arbitrary (non-interpolation) point and hence we have

$$f(x) - p(x) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \prod_{i=0}^{n} (x - x_i)^2. \quad \blacksquare$$

Example 3.21
Assume that we would like to find the Hermite interpolation polynomial that satisfies:

$$p(x_0) = y_0, \quad p'(x_0) = d_0, \quad p(x_1) = y_1, \quad p'(x_1) = d_1.$$

In this case $n = 1$, and

$$l_0(x) = \frac{x - x_1}{x_0 - x_1}, \quad l_0'(x) = \frac{1}{x_0 - x_1}, \quad l_1(x) = \frac{x - x_0}{x_1 - x_0}, \quad l_1'(x) = \frac{1}{x_1 - x_0}.$$

According to (3.50), the desired polynomial is given by (check!)

$$\begin{aligned} p(x) &= y_0 \left[1 + \frac{2}{x_0 - x_1}(x_0 - x)\right] \left(\frac{x - x_1}{x_0 - x_1}\right)^2 + y_1 \left[1 + \frac{2}{x_1 - x_0}(x_1 - x)\right] \left(\frac{x - x_0}{x_1 - x_0}\right)^2 \\ &\quad + d_0 (x - x_0) \left(\frac{x - x_1}{x_0 - x_1}\right)^2 + d_1 (x - x_1) \left(\frac{x - x_0}{x_1 - x_0}\right)^2. \end{aligned}$$

3.10 Spline Interpolation

So far, the only type of interpolation we were dealing with was polynomial interpolation. In this section we discuss a different type of interpolation: piecewise-polynomial interpolation. A simple example of such interpolants will be the function we get by connecting data with straight lines (see Figure 3.5). Of course, we would like to generate functions

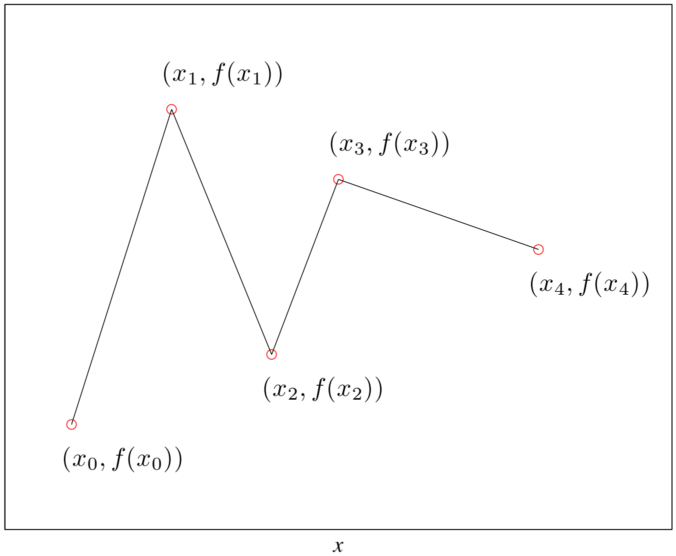

Figure 3.5: A piecewise-linear spline. In every subinterval the function is linear. Overall it is continuous where the regularity is lost at the knots

that are somewhat smoother than piecewise-linear functions, and still interpolate the data. The functions we will discuss in this section are **splines**.

You may still wonder why are we interested in such functions at all? It is easy to motivate this discussion by looking at Figure 3.6. In this figure we demonstrate what a high-order interpolant looks like. Even though the data that we interpolate has only one extrema in the domain, we have no control over the oscillatory nature of the high-order interpolating polynomial. In general, high-order polynomials are oscillatory, which rules them as non-practical for many applications. That is why we focus our attention in this section on splines.

Splines, should be thought of as polynomials on subintervals that are connected in a "smooth way". We will be more rigorous when we define precisely what we mean by smooth. First, we pick $n+1$ points which we refer to as the **knots**: $t_0 < t_1 < \cdots < t_n$. A **spline of degree** k having knots $t_0, \ldots, t_n$ is a function $s(x)$ that satisfies the following two properties:

1. On $[t_{i-1}, t_i)$ $s(x)$ is a polynomial of degree $\leqslant k$, i.e., $s(x)$ is a polynomial on every subinterval that is defined by the knots.

2. Smoothness: $s(x)$ has a continuous $(k-1)^{\text{th}}$ derivative on the interval $[t_0, t_n]$.

A spline of degree 0 is a piecewise-constant function (see Figure 3.7). A spline of

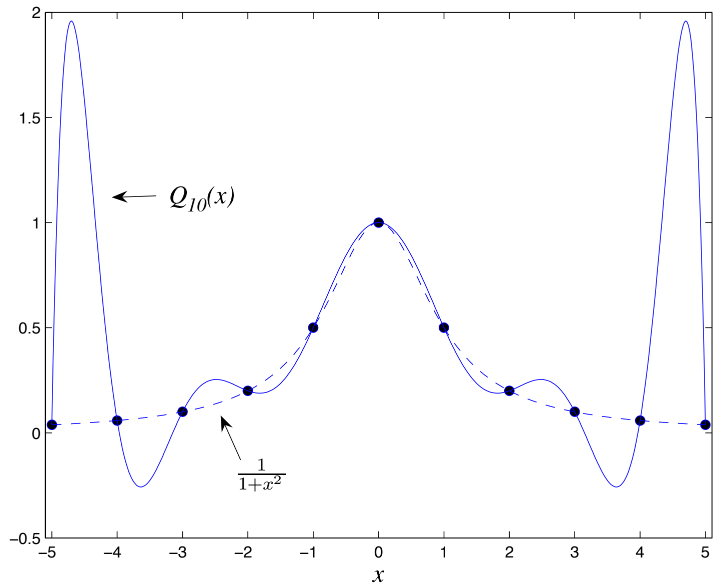

Figure 3.6: An interpolant "goes bad". In this example we interpolate 11 equally spaced samples of $f(x) = \frac{1}{1+x^2}$ with a polynomial of degree 10, $Q_{10}(x)$

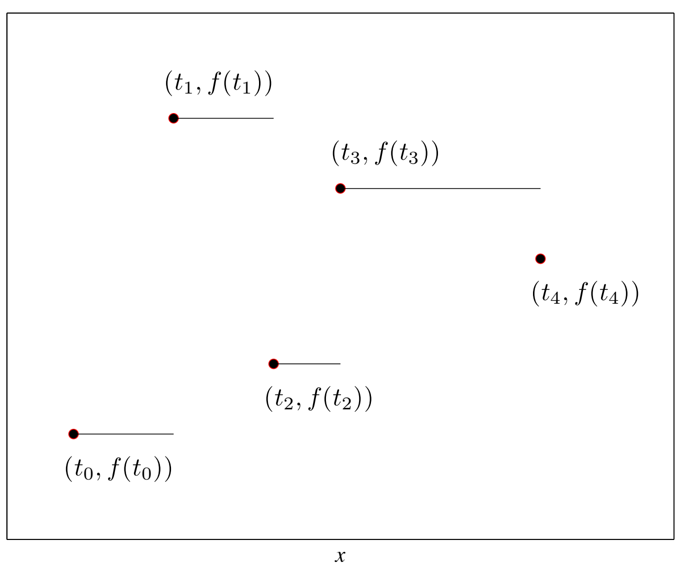

Figure 3.7: A zeroth-order (piecewise-constant) spline. The knots are at the interpolation points. Since the spline is of degree zero, the function is not even continuous

degree 1 is a piecewise-linear function that can be explicitly written as

$$s(x) = \begin{cases} s_0(x) = a_0x + b_0, & x \in [t_0, t_1), \\ s_1(x) = a_1x + b_1, & x \in [t_1, t_2), \\ \quad \vdots & \quad \vdots \\ s_{n-1}(x) = a_{n-1}x + b_{n-1}, & x \in [t_{n-1}, t_n], \end{cases}$$

(see Figure 3.5 where the knots $\{t_i\}$ and the interpolation points $\{x_i\}$ are assumed to be identical). It is now obvious why the points $t_0, \ldots, t_n$ are called knots: these are the points that connect the different polynomials with each other. To qualify as an interpolating function, $s(x)$ will have to satisfy interpolation conditions that we will discuss below. We would like to comment already at this point that knots should not be confused with the interpolation points. Sometimes it is convenient to choose the knots to coincide with the interpolation points but this is only optional, and other choices can be made.

3.10.1 Cubic splines

A special case (which is the most common spline function that is used in practice) is the cubic spline. A cubic spline is a spline for which the function is a polynomial of degree $\leqslant 3$ on every subinterval, and a function with two continuous derivatives overall (see Figure 3.8).

Let's denote such a function by $s(x)$, i.e.,

$$s(x) = \begin{cases} s_0(x), & x \in [t_0, t_1), \\ s_1(x), & x \in [t_1, t_2), \\ \quad \vdots & \quad \vdots \\ s_{n-1}(x), & x \in [t_{n-1}, t_n], \end{cases}$$

where $\forall i$, the degree of $s_i(x)$ is $\leqslant 3$.

We now assume that some data (that $s(x)$ should interpolate) is given at the knots, i.e.,

$$s(t_i) = y_i, \qquad 0 \leqslant i \leqslant n. \tag{3.52}$$

The interpolation conditions (3.52) in addition to requiring that $s(x)$ is continuous, imply that

$$s_{i-1}(t_i) = y_i = s_i(t_i), \qquad 1 \leqslant i \leqslant n-1. \tag{3.53}$$

We also require the continuity of the first and the second derivatives, i.e.,

$$\begin{aligned} s_i'(t_{i+1}) &= s_{i+1}'(t_{i+1}), \qquad 0 \leqslant i \leqslant n-2, \\ s_i''(t_{i+1}) &= s_{i+1}''(t_{i+1}), \qquad 0 \leqslant i \leqslant n-2. \end{aligned} \tag{3.54}$$

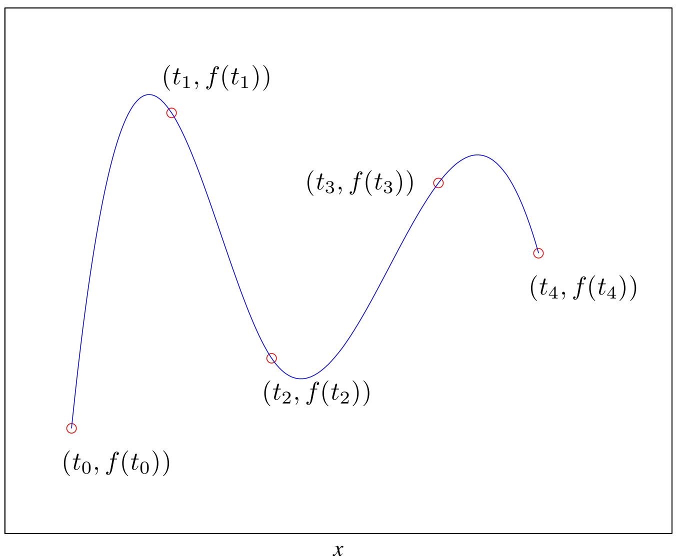

Figure 3.8: A cubic spline. In every subinterval $[t_{i-1}, t_i$, the function is a polynomial of degree $\leqslant 2$. The polynomials on the different subintervals are connected to each other in such a way that the spline has a second-order continuous derivative. In this example we use the not-a-knot condition.

Before actually computing the spline, let's check if we have enough equations to determine a unique solution for the problem. There are n subintervals, and in each subinterval we have to determine a polynomial of degree $\leqslant 3$. Each such polynomial has 4 coefficients, which leaves us with $4n$ coefficients to determine. The interpolation and continuity conditions (3.53) for $s_i(t_i)$ and $s_i(t_{i+1})$ amount to $2n$ equations. The continuity of the first and the second derivatives (3.54) add $2(n-1) = 2n-2$ equations. Altogether we have $4n-2$ equations but $4n$ unknowns which leaves us with 2 degrees of freedom. These indeed are two degrees of freedom that can be determined in various ways as we shall see below.

We are now ready to compute the spline. We will use the following notation:

$$h_i = t_{i+1} - t_i.$$

We also set

$$z_i = s''(t_i).$$

Since the second derivative of a cubic function is linear, we observe that $s_i''(x)$ is the line connecting (t_i, z_i) and (t_{i+1}, z_{i+1}), i.e.,

$$s_i''(x) = \frac{x-t_i}{h_i} z_{i+1} - \frac{x-t_{i+1}}{h_i} z_i. \tag{3.55}$$

Integrating (3.55) once, we have

$$s_i'(x) = \frac{1}{2}(x-t_i)^2 \frac{z_{i+1}}{h_i} - \frac{1}{2}(x-t_{i+1})^2 \frac{z_i}{h_i} + \tilde{c}.$$

Integrating again

$$s_i(x) = \frac{z_{i+1}}{6h_i}(x-t_i)^3 + \frac{z_i}{6h_i}(t_{i+1}-x)^3 + C(x-t_i) + D(t_{i+1}-x).$$

The interpolation condition, $s(t_i) = y_i$, implies that

$$y_i = \frac{z_i}{6h_i} h_i^3 + Dh_i,$$

i.e.,

$$D = \frac{y_i}{h_i} - \frac{z_i h_i}{6}.$$

Similarly, $s_i(t_{i+1}) = y_{i+1}$, implies that

$$y_{i+1} = \frac{z_{i+1}}{6h_i} h_i^3 + Ch_i,$$

i.e.,

$$C = \frac{y_{i+1}}{h_i} - \frac{z_{i+1}}{6}h_i.$$

This means that we can rewrite $s_i(x)$ as

$$s_i(x) = \frac{z_{i+1}}{6h_i}(x-t_i)^3 + \frac{z_i}{6h_i}(t_{i+1}-x)^3 + \left(\frac{y_{i+1}}{h_i} - \frac{z_{i+1}}{6}h_i\right)(x-t_i) + \left(\frac{y_i}{h_i} - \frac{z_i}{6}h_i\right)(t_{i+1}-x).$$

All that remains to determine is the second derivatives of $s(x)$, $z_0, \ldots, z_n$. We can set $z_1, \ldots, z_{n-1}$ using the continuity conditions on $s'(x)$, i.e., $s_i'(t_i) = s_{i-1}'(t_i)$. We first compute $s_i'(x)$ and $s_{i-1}'(x)$:

$$\begin{aligned}
s_i'(x) &= \frac{z_{i+1}}{2h_i}(x-t_i)^2 - \frac{z_i}{2h_i}(t_{i+1}-x)^2 + \frac{y_{i+1}}{h_i} - \frac{z_{i+1}}{6}h_i - \frac{y_i}{h_i} + \frac{z_i h_i}{6}.\\
s_{i-1}'(x) &= \frac{z_i}{2h_{i-1}}(x-t_{i-1})^2 - \frac{z_{i-1}}{2h_{i-1}}(t_i-x)^2 + \frac{y_i}{h_{i-1}} - \frac{z_i}{6}h_{i-1} - \frac{y_{i-1}}{h_{i-1}} + \frac{z_{i-1}h_{i-1}}{6}.
\end{aligned}$$

So that

$$\begin{aligned}
s_i'(t_i) &= -\frac{z_i}{2h_i}h_i^2 + \frac{y_{i+1}}{h_i} - \frac{z_{i+1}}{6}h_i - \frac{y_i}{h_i} + \frac{z_i h_i}{6}\\
&= -\frac{h_i}{3}z_i - \frac{h_i}{6}z_{i+1} - \frac{y_i}{h_i} + \frac{y_{i+1}}{h_i},\\
s_{i-1}'(t_i) &= \frac{z_i}{2h_{i-1}}h_{i-1}^2 + \frac{y_i}{h_{i-1}} - \frac{z_i}{6}h_{i-1} - \frac{y_{i-1}}{h_{i-1}} + \frac{z_{i-1}h_{i-1}}{6}\\
&= \frac{h_{i-1}}{6}z_{i-1} + \frac{h_{i-1}}{3}z_i - \frac{y_{i-1}}{h_{i-1}} + \frac{y_i}{h_{i-1}}.
\end{aligned}$$

Hence, for $1 \leqslant i \leqslant n-1$, we obtain the system of equations

$$\frac{h_{i-1}}{6}z_{i-1} + \frac{h_i + h_{i-1}}{3}z_i + \frac{h_i}{6}z_{i+1} = \frac{1}{h_i}(y_{i+1}-y_i) - \frac{1}{h_{i-1}}(y_i - y_{i-1}). \tag{3.56}$$

These are $n-1$ equations for the $n+1$ unknowns, $z_0, \ldots, z_n$, which means that we have 2 degrees of freedom. Without any additional information about the problem, the only way to proceed is by making an arbitrary choice. There are several standard ways to proceed. One option is to set the end values to zero, i.e.,

$$z_0 = z_n = 0. \tag{3.57}$$

This choice of the second derivative at the end points leads to the so-called, **natural cubic spline**. We will explain later in what sense this spline is "natural". In this case, we end up with the following linear system of equations

$$\begin{pmatrix}
\frac{h_0+h_1}{3} & \frac{h_1}{6} & & & \\
\frac{h_1}{6} & \frac{h_1+h_2}{3} & \frac{h_2}{6} & & \\
 & \ddots & \ddots & \ddots & \\
 & & \frac{h_{n-3}}{6} & \frac{h_{n-3}+h_{n-2}}{3} & \frac{h_{n-2}}{6}\\
 & & & \frac{h_{n-2}}{6} & \frac{h_{n-2}+h_{n-1}}{3}
\end{pmatrix}
\begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_{n-2} \\ z_{n-1} \end{pmatrix}
=
\begin{pmatrix}
\frac{y_2-y_1}{h_1} - \frac{y_1-y_0}{h_0}\\
\frac{y_3-y_2}{h_2} - \frac{y_2-y_1}{h_1}\\
\vdots\\
\frac{y_{n-1}-y_{n-2}}{h_{n-2}} - \frac{y_{n-2}-y_{n-3}}{h_{n-3}}\\
\frac{y_n-y_{n-1}}{h_{n-1}} - \frac{y_{n-1}-y_{n-2}}{h_{n-2}}
\end{pmatrix}$$

The coefficients matrix is symmetric, tridiagonal, and diagonally dominant (i.e., $|a_{ii}| > \sum_{j=1,j\neq i}^{n} |a_{ij}|$, $\forall i$), which means that it can always be (efficiently) inverted.

In the special case where the points are equally spaced, i.e., $h_i = h$, $\forall i$, the system becomes

$$\begin{pmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & 1 & 4 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_{n-2} \\ z_{n-1} \end{pmatrix} = \frac{6}{h^2} \begin{pmatrix} y_2 - 2y_1 + y_0 \\ y_3 - 2y_2 + y_1 \\ \vdots \\ y_{n-1} - 2y_{n-2} + y_{n-3} \\ y_n - 2y_{n-1} + y_{n-2} \end{pmatrix} \tag{3.58}$$

In addition to the natural spline (3.57), there are other standard options:

1. If the values of the derivatives at the endpoints are known, one can specify them

   $$s'(t_0) = y_0', \qquad s'(t_n) = y_n'.$$

2. The **not-a-knot** condition. Here, we require the third-derivative $s^{(3)}(x)$ to be continuous at the points t_1, t_{n-1}. In this case we end up with a cubic spline with knots $t_0, t_2, t_3, \ldots, t_{n-2}, t_n$. The points t_1 and t_{n-1} no longer function as knots. The interpolation requirements are still satisfied at $t_0, t_1, \ldots, t_{n-1}, t_n$. Figure 3.9 shows two different cubic splines that interpolate the same initial data. The spline that is plotted with a solid line is the not-a-knot spline. The spline that is plotted with a dashed line is obtained by setting the derivatives at both end-points to zero.

3.10.2 What is natural about the natural spline?

The following theorem states that the natural spline cannot have a larger L^2-norm of the second-derivative than the function it interpolates (assuming that that function has a continuous second-derivative). In fact, we are minimizing the L^2-norm of the second-derivative not only with respect to the "original" function which we are interpolating, but with respect to any function that interpolates the data (and has a continuous second-derivative). In that sense, we refer to the natural spline as "natural".

Theorem 3.22 *Assume that $f''(x)$ is continuous in $[a,b]$, and let $a = t_0 < t_1 < \cdots < t_n = b$. If $s(x)$ is the natural cubic spline interpolating $f(x)$ at the knots $\{t_i\}$ then*

$$\int_a^b (s''(x))^2 dx \leqslant \int_a^b (f''(x))^2 dx.$$

Proof. Define $g(x) = f(x) - s(x)$. Then since $s(x)$ interpolates $f(x)$ at the knots $\{t_i\}$ their difference vanishes at these points, i.e.,

$$g(t_i) = 0, \qquad 0 \leqslant i \leqslant n.$$

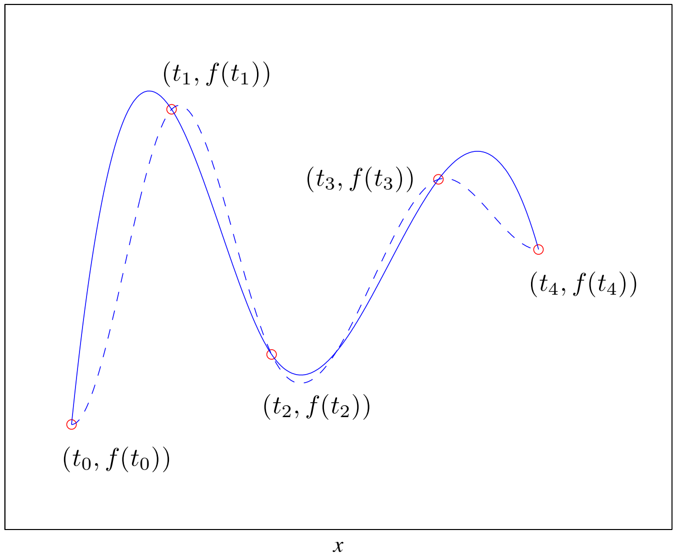

Figure 3.9: Two cubic splines that interpolate the same data. *Solid line:* a not-a-knot spline; *Dashed line:* the derivative is set to zero at both end-points

Now

$$\int_a^b (f'')^2 dx = \int_a^b (s'')^2 dx + \int_a^b (g'')^2 dx + 2\int_a^b s''g''dx. \tag{3.59}$$

We will show that the last term on the right-hand-side of (3.59) is zero, which will conclude the proof as the other two terms on the right-hand-side of (3.59) are non-negative. Splitting that term into a sum of integrals on the subintervals and integrating by parts on every subinterval, we have

$$\int_a^b s''g''dx = \sum_{i=1}^n \int_{t_{i-1}}^{t_i} s''g''dx = \sum_{i=1}^n \left[(s''g')\Big|_{t_{i-1}}^{t_i} - \int_{t_{i-1}}^{t_i} s'''g'dx \right].$$

Since we are dealing with the "natural" choice $s''(t_0) = s''(t_n) = 0$, and since $s'''(x)$ is constant on $[t_{i-1}, t_i]$ (say c_i), we end up with

$$\int_a^b s''g''dx = -\sum_{i=1}^n \int_{t_{i-1}}^{t_i} s'''g'dx = -\sum_{i=1}^n c_i \int_{t_{i-1}}^{t_i} g'dx = -\sum_{i=1}^n c_i(g(t_i) - g(t_{i-1})) = 0. \quad \blacksquare$$

We note that $f''(x)$ can be viewed as a linear approximation of the curvature

$$\frac{|f''(x)|}{(1 + (f'(x))^2)^{\frac{3}{2}}}.$$

From that point of view, minimizing $\int_a^b (f''(x))^2 dx$, can be viewed as finding the curve with a minimal $|f''(x)|$ over an interval.