Sect.1 Representation of a Function by Polynomials

It is frequently convenient and sometimes imperative that a function $y = f(x)$ be replaced by a simpler function $y = app(x)$ so that the properties and values of $f(x)$ can be studied or obtained from the corresponding properties or values of $app(x)$. If we put

$$f(x) = app(x) + E(x)$$

we may regard $app(x)$ as an approximation to $f(x)$ and $E(x)$ as the error function. It is desirable, for the most part, that the approximating function $app(x)$ will be a polynomial

$$P(x) = a_0 + a_1x + \cdots + a_nx^n \tag{1}$$

whose degree does not exceed a preassigned n and which approximates $f(x)$ as well as possible.

Let us suppose first that $A(x_0, y_0)$ is a point on the graph of $y = f(x)$. We obtain in this section a polynomial whose graph approximates as well as possible in some intuitive sense the graph of $y = f(x)$ in the neighborhood of point A. It would seem natural to require that the graph of the polynomial pass through A, that its tangent coincide with the tangent to the graph of $y = f(x)$ at A, and that its radius of curvature coincide with the radius of curvature of $y = f(x)$ at A. These requirements will be satisfied if

$$P(x_0) = f(x_0), \qquad P'(x_0) = f'(x_0), \qquad P''(x_0) = f''(x_0).$$

Let us choose the a's in (1) so that

$$P(x_0) = f(x_0), \qquad P'(x_0) = f'(x_0), \quad \dots, \quad P^{(n)}(x_0) = f^{(n)}(x_0).$$

We assume that $f(x)$ possesses all the derivatives in question, but it remains to show that the conditions just imposed uniquely determine the a's.
It follows from (1) that to satisfy these conditions we must solve the linear equations

$$\begin{array}{rrrrrrl}
a_0+ & a_1x_0+ & a_2x_0^2+ & a_3x_0^3+ & \cdots + & a_nx_0^n & = f(x_0) \\
 & a_1+ & 2a_2x_0+ & 3a_3x_0^2+ & \cdots + & na_nx_0^{n-1} & = f'(x_0) \\
 & & 2a_2+ & 6a_3x_0+ & \cdots + & n(n-1)a_nx_0^{n-2} & = f''(x_0) \\
 & & & & \vdots & & \vdots \\
 & & & & & n!a_n & = f^{(n)}(x_0)
\end{array}$$

for a_0, a_1,..., a_n. We solve the last equation for a_n, then the preceding one for a_{n-1}, and so on. It develops that the a's are uniquely determined and are given by

$$\begin{aligned}
a_0 &= f(x_0) - x_0 f'(x_0) + \frac{x_0^2}{2!} f''(x_0) + \cdots + (-1)^n \frac{x_0^n}{n!} f^{(n)}(x_0) \\
a_1 &= \frac{1}{1!} f'(x_0) - \frac{2x_0}{2!} f''(x_0) + \cdots + (-1)^{(n-1)} \frac{n x_0^{n-1}}{n!} f^{(n)}(x_0) \\
\vdots & \qquad \vdots \\
a_{n-1} &= \frac{f^{(n-1)}(x_0)}{(n-1)!} - \frac{n x_0}{n!} f^{(n)}(x_0) \\
a_n &= \frac{f^{(n)}(x_0)}{n!}
\end{aligned}$$

If we multiply these equations by 1, x, x^2, ..., x^n, respectively, and then sum the right-hand members by diagonals that slope up and to the right, we find that

$$P(x) = f(x_0) + (x - x_0) f'(x_0) + \frac{(x - x_0)^2}{2!} f''(x_0) + \cdots + \frac{(x - x_0)^n}{n!} f^{(n)}(x_0)$$

is the required polynomial.
Therefore, by Taylor's theorem we merely deduce the expression of the error function made in replacing $f(x)$ by $P(x)$ as follows

$$E(x) = f(x) - P(x) = \frac{(x - x_0)^{n+1}}{(n+1)!} f^{(n+1)}\left(x_0 + \xi(x - x_0)\right), \qquad 0 < \xi < 1.$$

As a justification for the approximation of an unknown function by means of a polynomial, we state a famous theorem due to Weierstrass (1885):

Theorem 1: (Weierstrass Approximation Theorem)

Suppose that f is defined and continuous on $[a, b]$. For each $\epsilon > 0$, there exists a polynomial $P(x)$ defined on $[a, b]$, with the property that

$$\left| f(x) - P(x) \right| < \epsilon, \qquad \forall\ x \in [a, b].$$

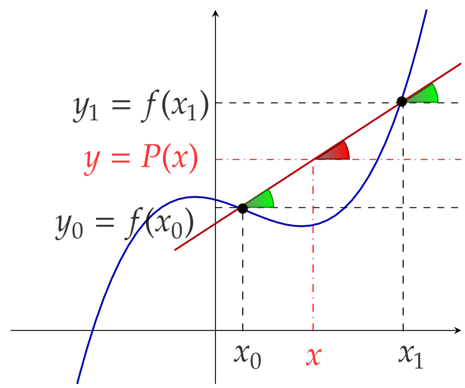
Sect.2 Polynomial Interpolation

Assume we are given a single-valued and continuous function $f(x)$, known explicitly, then the values of $f(x)$ corresponding to certain given values of x, say x_0, x_1, ..., x_n can easily be computed.

The central problem of numerical analysis is the converse one. We shall next consider the problem where we know the values of a function at a certain set of predetermined points, $\big(x_1, f(x_1)\big)$, $\big(x_2, f(x_2)\big)$, ...$\big(x_n, f(x_n)\big)$. The question to be answered in this chapter is what values does the function take at intermediate values of x (which is called interpolation), or alternatively what values does the function take external to this range (which is called extrapolation).
It is required to find a simpler function, say ϕ, which agrees with f at the set of tabulated points. If ϕ is a polynomial, then the process is called polynomial interpolation and ϕ is called the interpolating polynomial. Similarly, different types of interpolation arise depending on whether ϕ is a finite trigonometric series, series of Legendre functions, etc.

Sect.3 Lagrange Polynomials

Let us start with the determination of a polynomial of degree 1 that passes through the distinct points $\big(x_0, y_0\big)$ and $\big(x_1, y_1\big)$, which is the same as approximating a function f for which $f(x_0) = y_0$ and $f(x_1) = y_1$ by means of a first-degree polynomial agreeing with the values of f at these points.



From the graph, by remarking the equality

$$\frac{y_1 - P(x)}{x_1 - x} = \frac{P(x) - y_0}{x - x_0}$$

we merely deduce the expression of the polynomial as follows

$$P(x) = \left(\frac{x - x_0}{x_1 - x_0}\right) f(x_1) + \left(\frac{x - x_1}{x_0 - x_1}\right) f(x_0) = L_1(x)\, f(x_1) + L_0(x)\, f(x_0). \tag{2}$$

Note that these definitions imply that

$$L_i(x_i) = 1, \qquad L_i(x_j) = 0, \quad \text{and} \quad P(x_i) = f(x_i).$$

To generalize the concept of linear interpolation to higher-degree polynomials, consider the construction of a polynomial of degree at most n that passes through the $n+1$ distinct points

$$\big(x_0, f(x_0)\big),\ \big(x_1, f(x_1)\big),\ \dots\ ,\big(x_n, f(x_n)\big).$$

We could write

$$P_n(x) = a_0 + a_1 x + \cdots + a_n x^n$$

and, on putting $P(x_i) = f(x_i)$, $0 \leqslant i \leqslant n$, we obtain $n+1$ linear equations with $a_0, a_1, \dots, a_n$ unknown, i.e.

$$\left\{\begin{array}{rcl} a_0 + a_1 x_0 + a_2 x_0^2 + \cdots + a_n x_0^n &=& y_0 \\ a_0 + a_1 x_1 + a_2 x_1^2 + \cdots + a_n x_1^n &=& y_1 \\ \vdots & & \vdots \\ a_0 + a_1 x_n + a_2 x_n^2 + \cdots + a_n x_n^n &=& y_n \end{array}\right. \iff \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} \tag{3}$$

We know the expression of Vandermonde determinant

$$D_n = \begin{vmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{vmatrix} = \prod_{\substack{i=0 \\ i>j}}^{n} \left(x_i - x_j\right) \tag{4}$$

Since these equations are difficult to solve unless n is small, we prefer a different approach based on a generalization of (2). We write P_n in the form

$$P_n(x) = L_0(x) f(x_0) + L_1(x) f(x_1) + \cdots + L_n(x) f(x_n)$$

where each polynomial L_i of degree n. The polynomial P, will have the same values as f at $x = x_0,\ x_1,\ \dots, x_n$ if

$$L_i(x_j) = \delta_{ij}$$

So, for example, L_0 has the value zero at $x = x_1,\ x_2,\ \dots, x_n$ and has the value 1 at $x = x_0$. Therefore, if we put

$$L_0(x) = C_0 \left(x - x_1\right)\left(x - x_2\right) \dots \left(x - x_n\right) \tag{5}$$

where C_0 is a constant, L_0 will indeed be zero at $x = x_1,\ x_2,\ \dots, x_n$ and L_0 is polynomial of degree n. Putting $x = x_1$ in (5) the condition $L_0(x_0) = 1$ gives

$$1 = C_0 \left(x_0 - x_1\right)\left(x_0 - x_2\right) \dots \left(x_0 - x_n\right)$$

which fixes the value of C_0. On substituting this value in (5) we obtain

$$L_0(x) = \frac{\left(x - x_1\right)\left(x - x_2\right) \dots \left(x - x_n\right)}{\left(x_0 - x_1\right)\left(x_0 - x_2\right) \dots \left(x_0 - x_n\right)} = \prod_{k=1}^{n} \left(\frac{x - x_k}{x_0 - x_k}\right) \tag{6}$$

Similarly, for a general value of i, $1 \leqslant i \leqslant n$, we obtain

$$L_i(x) = \frac{\left(x - x_0\right) \dots \left(x - x_{i-1}\right)\left(x - x_{i+1}\right) \dots \left(x - x_n\right)}{\left(x_i - x_0\right) \dots \left(x_i - x_{i-1}\right)\left(x_i - x_{i+1}\right) \dots \left(x_i - x_n\right)} = \prod_{\substack{k=0 \\ k \neq i}}^{n} \left(\frac{x - x_k}{x_i - x_k}\right) \tag{7}$$

In (7) the product is taken over all values of j from 0 to n except for $j = i$. From (7) we see immediately that L_i is zero at all values $x = x_0,\ x_1,\ \ldots, x_n$, except for $x = x_i$ when L_i takes the value 1. This agrees with the requirement $L_i(x_j) = \delta_{ij}$ and the interpolating polynomial can be written in the form

$$P_n(x) = \sum_{i=0}^{n} L_i(x) f(x_i). \tag{8}$$

This is known as the Lagrange form of the interpolating polynomial.
Now, since the determinant (4) is not zero as long as the x_i are distinct, then the system (3) has a unique solution, whence (8) is uniquely determined. We summarize in the

Theorem 2

If $\left(x_0, y_0\right),\ \left(x_1, y_1\right),\ \ldots\ ,\left(x_n, y_n\right)$ are $n+1$ points with distinct abscissas, there is one and only one polynomial $y = P_n(x)$ of degree at most n, whose graph passes through these points, that is, such that $y_i = P_n(x_i)$, $i = \overline{0, n}$.

Proof. Let us prove the unicity first. Suppose there were two such polynomials, p_n and q_n. Then the polynomial $p_n - q_n$ would have the property that $(p_n - q_n)(x_i) = 0$ for $0 \leqslant i \leqslant n$. Since the degree of $p_n - q_n$ can be at most n, this polynomial can have at most n zeros if it is not the zero polynomial. Since the x_i are distinct, $p_n - q_n$ has $n+1$ zeros, it must therefore be zero. Hence, $p_n = q_n$.
For the existence part, we proceed inductively. For $n = 0$, the existence is obvious since a constant function p_0 (polynomial of degree at most 0) can be chosen so that $p_0(x_0) = y_0$. Now suppose that we have obtained a polynomial p_{k-1} of degree at most $k-1$ with $p_{k-1}(x_i) = y_i$, for $0 \leqslant i \leqslant k-1$. We try to construct p_k in the form

$$p_k = p_{k-1} + c\left(x - x_0\right)\left(x - x_1\right) \ldots \left(x - x_{k-1}\right).$$

Note that this is unquestionably a polynomial of degree at most k. Furthermore, p_k interpolates the data that p_{k-1} interpolates, because

$$p_k(x_i) = p_{k-1}(x_i) = y_i, \qquad 0 \leqslant i \leqslant k-1.$$

Now we determine the unknown coefficient c from the condition $p_k(x_k) = y_k$. This leads to the equation

$$p_k(x_k) = p_{k-1}(x_k) + c\left(x_k - x_0\right)\left(x_k - x_1\right) \ldots \left(x_k - x_{k-1}\right) = y_k.$$

Since the x_i's are distinct, then the factors multiplying c are not zero, therefore the latter equation gives

$$c = \frac{y_k - p_{k-1}(x_k)}{(x_k - x_0)(x_k - x_1) \dots (x_k - x_{k-1})}. \tag{9}$$

The fraction (9) provides an alternative way for constructing recursively the interpolating polynomial. □

Exercise 1

Construct the Lagrange interpolating polynomial of

(a) $(0,2),\ (1,2),\ (3,1)$. (b) $\ln(x),\ 1,\ 2,\ 3,\ 4$. (c) $(-1,a),\ (0,b),\ (1,c)$.

Exercise 2

Construct the Lagrange interpolating polynomial for the function 2^x based on the points -2, -1, 0, 1 and 2. Estimate $\sqrt{2}$, $\sqrt[3]{2}$ and $\sqrt[5]{2}$.

Exercise 3

With $\omega(x) = (x - x_0)(x - x_1) \dots (x - x_n)$, verify that (you can l'Hôpital's rule)

(a) $\omega'(x_i) = (x_i - x_0)(x_i - x_1) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)$

(b) $P_n(x) = \sum_{i=0}^{n} L_i(x) f(x_i) = \sum_{i=0}^{n} \frac{\omega(x)}{(x - x_i)\omega'(x_i)} f(x_i)$. $\left(i.e.\ \ L_i(x) = \frac{\omega(x)}{(x - x_i)\omega'(x_i)}\right)$

When it is desired to compute the Lagrangian coefficients for some given numerical value of the variable x, it is advantageous to arrange the computation in a compact and systematic form. In the case where the values x_0, x_1,..., x_n are not equally spaced, the following scheme is suggested, and is illustrated for the case $n = 3$. We first set up the square array of differences

$$\begin{array}{cccc}
\boxed{(x - x_0)} & (x_0 - x_1) & (x_0 - x_2) & (x_0 - x_3) \\
(x_1 - x_0) & \boxed{(x - x_1)} & (x_1 - x_2) & (x_1 - x_3) \\
(x_2 - x_0) & (x_2 - x_1) & \boxed{(x - x_2)} & (x_2 - x_3) \\
(x_3 - x_0) & (x_3 - x_1) & (x_3 - x_2) & \boxed{(x - x_3)}
\end{array}$$

Let the product of the numbers in the principal diagonal be denoted by $\pi(x)$, let the product of numbers in the first row be $\phi_0(x)$, second row $\phi_1(x)$, etc. Then it is evident that

$$L_i(x) = \frac{\pi(x)}{\phi_i(x)}, \qquad P_n(x) = \sum_{i=0}^{n} L_i(x) f(x_i) = \sum_{i=0}^{n} \frac{\pi(x)}{\phi_i(x)} f(x_i).$$

Sect.4 Newton's Interpolating Polynomials

The Lagrange formula (7) and (8) has one drawback. If we desire to pass from a space of dimension n to a space of one higher dimension, we must determine an entirely new set of elements $\tilde{L}_0$, $\tilde{L}_1$, ..., $\tilde{L}_{n+1}$ that are not related in a simple fashion to the old set x_i, $i = \overline{1, n}$. A representation of Newton gets around this difficulty by taking linear combinations of both the basis elements x_i and L_i . We shall describe this idea here.

Therefore, by remarking that the set

$$\{1,\ (x - x_0),\ (x - x_0)(x - x_1),\ \dots,\ (x - x_0)(x - x_1) \dots (x - x_{n-1})\}$$

are linearly independent (constitutes a basis for $\mathscr{P}_n$), the divided differences of f with respect to $x_0, x_1, \dots, x_n$ are used to express $P_n(x)$ in the form

$$P_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_n(x - x_0)(x - x_1) \dots (x - x_{n-1}) \tag{10}$$

We shall focus next on the determination of the coefficients a_i's of (10) as the point of view of (9) in more concise and compact way.

First, from the graph (in case of degree one), we have firstly

$$\frac{y_1 - y_0}{x_1 - x_0} = \frac{P(x) - y_0}{x - x_0} \implies P(x) = y_0 + \left(\frac{y_1 - y_0}{x_1 - x_0}\right)(x - x_0) = a_0 + a_1\,(x - x_0)$$

which it is an equation of a straight line that passes through the points.
For three given points, (x_0, y_0), (x_1, y_1), and (x_2, y_2), the second-order interpolating polynomial has the form:

$$P(x) = a_0 + a_1\,(x - x_0) + a_2\,(x - x_0)\,(x - x_1)\,. \tag{11}$$

It is an equation of a parabola that passes through the three points. The coefficients a_1, a_2 and a_3 can be determined by substituting the three points in the equation (11). Indeed, substituting $x = x_0$ and $f(x_0) = y_0$ gives $a_0 = y_0$. Substituting the second point $x = x_1$ and $f(x_1) = y_1$ (and $a_0 = y_0$)

in (11) gives:

$$a_1 = \frac{y_1 - y_0}{x_1 - x_0}.$$

Substituting the third point, $x = x_2$ and $f(x_2) = y_2$, (as well as the expressions of a_0 and a_1) in (11) gives:

$$a_2 = \frac{\frac{y_2-y_1}{x_2-x_1} - \frac{y_1-y_0}{x_1-x_0}}{x_2 - x_0}.$$

The coefficients a_0 and a_1 are the same in the first-order and second-order polynomials. This means that if two points are given and a first-order interpolating polynomial is fit to pass through those points, and then a third point is added, the polynomial can be changed to be of second-order and pass through the three points by only determining the value of one additional coefficient.
It is worthy to remark that the above a_i's are solutions of the following triangular system obtained from (10)

$$\begin{cases} y_0 = P_n(x_0) & = a_0 \\ y_1 = P_n(x_1) & = a_0 + a_1(x_1 - x_0) \\ y_2 = P_n(x_2) & = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) \\ \quad\vdots & \quad\quad\quad\quad\quad\vdots \\ y_n = P_n(x_n) & = a_0 + a_1(x_n - x_0) + \cdots + a_n(x_n - x_0)\ldots(x_n - x_{n-1}) \end{cases} \tag{12}$$

Sect.5 Divided Differences

Newton described a method of interpolating equally spaced data. That method is basis of the modern divided difference method, which allows non-equally spaced data. The starting point for this method is the definition of divided differences.

The first-order divided differences

$$f[x_0, x_1] = \frac{f(x_0) - f(x_1)}{x_0 - x_1}, \qquad x_0 \neq x_1. \tag{13}$$

The second-order divided differences (provided that x_i's are distinct)

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

and continues to higher orders by the formula

$$f[x_0, x_1, \dots, x_n] = \frac{f[x_0, x_1, \dots, x_{n-1}] - f[x_1, x_2, \dots, x_n]}{x_0 - x_n}. \tag{14}$$

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.

The first-order divided differences do not depend on the order of the inputs:

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

This works more generally for higher order divided differences: the order of the x_i's in the divided difference does not matter. By induction we have the following Theorem

Theorem 3

If $y_0, y_1, \dots, y_n$ is any permutation of $x_0, x_1, \dots, x_n$ and all x_i's are distinct, then

$$f[x_0, x_1, \dots, x_n] = f[y_0, y_1, \dots, y_n]. \tag{15}$$

To compute the divided differences $f[x_0, x_1, \dots, x_n]$ for $k = 0, 1, \dots, n$, we consider a divided difference table:

x	Zero Diff.	First Diff.	Second Diff.	...	
x_0	$f(x_0)$				
x_1	$f(x_1)$	$f[x_0, x_1]$			
x_2	$f(x_2)$	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$		
x_n	$f(x_n)$	$f[x_{n-1}, x_n]$	$f[x_{n-2}, x_{n-1}, x_n]$	...	$f[x_0, x_1, \dots, x_n]$

We wish to compute the second-order divided difference $f[x_0, x_1, x_2] = \frac{f[x_0,x_1]-f[x_1,x_2]}{x_0-x_2}$, so we need both $f[x_0, x_1]$ and $f[x_1, x_2]$. This uses the values in the table to the entry in the table on its left, and the entry left and up.

We begin the computations by first computing the first-order divided differences $f[x_{i-1}, x_i]$. We do not wish to overwrite $f(x_0)$, but we can overwrite $f(x_1)$, $f(x_2)$, ..., $f(x_n)$ with $f[x_0, x_1]$, $f[x_1, x_2]$, ..., $f[x_{n-1}, x_n]$. We can first overwrite $f(x_n)$ with $f[x_{n-1}, x_n] = \frac{f(x_{n-1})-f(x_n)}{(x_{n-1}-x_n)}$, as there is no further use for $f(x_n)$ in this divided difference table. Then $f(x_{n-1})$ can be overwritten by

$f[x_{n-2}, x_{n-1}] = \frac{f(x_{n-2})-f(x_{n-1})}{(x_{n-2}-x_{n-1})}$. We can repeat this process going from the bottom to the top of the first-order divided differences. Once the first-order divided differences are computed, the second-order divided differences can be computed, again from the bottom to the top. Continuing in this way we can compute all the divided differences we need we obtain

$$P_n(x) = f[x_0] + \sum_{k=1}^{n} f[x_0, x_1, \dots, x_k](x-x_0)(x-x_1)\dots(x-x_k) \tag{16}$$

There are still other algorithms for polynomial interpolation, and these have various advantages and disadvantages. Since there is one and only one polynomial of degree $\leqslant n$ that takes prescribed values at $n+1$ given (distinct) points, these algorithms produce the same polynomial in different forms. For numerical work, it is probably best to use the Newton form of the interpolation polynomial. This can be combined with the divided difference algorithm for computing the required coefficients. We conclude this section with some of the nice properties of divided differences.

Theorem 4

Let p be the polynomial of degree at most n that interpolates a function f at a set of $n+1$ distinct nodes x_0, x_1, ..., x_n. If t is a point different from the nodes, then

$$f(t) - p(t) = f[x_0, x_1, \dots, x_n, t]\,(t-x_0)\,(t-x_1)\dots(t-x_n)\,. \tag{17}$$

Proof. First, let q be the polynomial of degree at most $n+1$ that interpolates f at the nodes x_0, x_1, ..., x_n, t. We know that q is obtained from p by adding one term. In fact

$$q(x) = p(x) + f[x_0, x_1, \dots, x_n, t]\,(x-x_0)\,(x-x_1)\dots(x-x_n)$$

Since $q(t) = f(t)$, then by letting $x = t$ we obtain at once (17). □

Theorem 5

If f is n times continuously differentiable on $[a, b]$ and if x_0, x_1, x_n are distinct points in $[a, b]$, then there exists a point ξ in $]a, b[$ such that

$$f[x_0, x_1, \dots, x_n] = \frac{1}{n!} f^{(n)}(\xi)\,. \tag{18}$$

Proof. Let us prove (15) first. Indeed, the divided difference on the left side of equation (15) is the coefficient of x^n in the polynomial of degree $\leq n$ interpolating f at the points y_0, y_1, ...y_n.

The divided difference on the right is the coefficient of x^n in the polynomial of degree $\leq n$ that interpolates f at the points x_0, x_1, ...x_n. These two polynomials are, of course, the same.
Now, let p be the polynomial of degree at most $n-1$ that interpolates f at the nodes x_0, x_1, ...x_{n-1}. From one hand, by Theorem Sect.6, there exists $\xi \in]a,b[$ such that

$$f(x_n) - p(x_n) = (x_n - x_0)(x_n - x_1) \dots (x_n - x_{n-1}) \frac{f^{(n)}(\xi_x)}{n!}.$$

From another hand, according to the preceding Theorem Sect.5 we have

$$f(x_n) - p(x_n) = (x_n - x_0)(x_n - x_1) \dots (x_n - x_{n-1}) f[x_0, x_1, \dots, x_n].$$

By comparing the above two equalities, we deduce (15). □

Exercise 4

1 Find the Newton interpolating polynomial for these data:

$$(-1,2), \quad (0,1), \quad (1,3), \quad (2,-5), \quad (7,47).$$

2 Find by use of the determinant form, the Lagrange formula and the method of divided differences, the cubic polynomial determined by each set of points.

$$(-3,-1), \quad (-2,2), \quad (1,-1), \quad (3,10).$$

Exercise 5

Let $f(x) = 1/x$. Prove that

$$f[x_0, x_1, \dots, x_n] = (-1)^n \prod_{i=0}^{n} x_i^{-1}.$$

Let us give some properties of divided differences. For example, if we set $x_1 = x + \epsilon$ and if f is differentiable, it follows

$$f[x,x] \xleftarrow[0 \leftarrow \epsilon]{} f[x_1, x] = f[x+\epsilon, x] = \frac{f(x+\epsilon) - f(x)}{\epsilon} \xrightarrow[\epsilon \to 0]{} f'(x)$$

i.e. $f[x,x] = f'(x)$. A similar argument shows that if x_0, x_1, ... , x_n are constants, then

$$\frac{d}{dx} f[x_0, x_1, \dots, x_k, x] = f[x_0, x_1, \dots, x_k, x, x]. \tag{19}$$

Therefore, if u_0, u_1, … , u_n are differentiable functions of x, there follows also

$$\frac{d}{dx}f[x_0,x_1,\dots,x_k,u_1,\dots,u_n]=\sum_{i=1}^{n}f[x_0,x_1,\dots,x_k,u_1,\dots,u_n,u_i]\frac{du_i}{dx}$$

and hence, by taking $u_1=\cdots=u_n=x$, we deduce that

$$\frac{d}{dx}f[x_0,x_1,\dots,x_k,\overbrace{x,\dots,x}^{n\text{ times}}]=nf[x_0,x_1,\dots,x_k,\overbrace{x,\dots,x}^{n+1\text{ times}}]. \tag{20}$$

Finally, by successive differentiating (19) combined with the use of (20) at each step, we may establish in additional, the following formula

$$\frac{d^r}{dx^r}f[x_0,x_1,\dots,x_k,x]=r!f[x_0,x_1,\dots,x_k,\overbrace{x,\dots,x}^{r+1\text{ times}}]. \tag{21}$$

Exercise 6

1. Show that the divided differences are linear maps on functions, i.e. prove that

$$\left(\alpha f+\beta g\right)[x_0,x_1,\dots,x_n]=\alpha f[x_0,x_1,\dots,x_n]+\beta g[x_0,x_1,\dots,x_n].$$

2. If $f(x)=u(x)v(x)$, then

$$f[x_0,x_1]=u[x_0]v[x_0,x_1]+u[x_0,x_1]v[x_1].$$

3. Prove the Leibniz formula

$$\left(fg\right)[x_0,x_1,\dots,x_n]=\sum_{k=0}^{n}f[x_0,x_1,\dots,x_k]\,g[x_k,x_{k+1},\dots,x_n]$$

4. When $f(x)=(ax+b)/(cx+d)$, obtain expressions for $f[x,y]$, $f[x,x,y]$, $f[x,x,y,y]$.

5. When $f(x)=1/(a-x)$, by taking $\pi(x)=(x-x_0)(x-x_1)\dots(x-x_n)$, show that

$$f[x_0,x_1,\dots,x_n]=\frac{1}{(a-x_0)(a-x_1)\dots(a-x_n)},$$
$$\frac{1}{a-x}=\frac{1}{a-x_0}+\frac{x-x_0}{(a-x_0)(a-x_1)}+\cdots+\frac{(x-x_0)(x-x_1)\dots(x-x_{n-1})}{(a-x_0)(a-x_1)\dots(x-x_n)}+\frac{\pi(x)}{\pi(a)(x-a)}.$$

Exercise 7

1. Show that if u is any function that interpolates f at $x_0,x_1,\dots,x_{n-1}$, and if v is a function that interpolates f at $x_1,x_2,\dots,x_n$, then the function h given below interpolates f at $x_0,x_1,\dots,x_n$ with

$$h(x) = \frac{(x_n - x)\,u(x) + (x - x_0)\,v(x)}{x_n - x_0}$$

2 Consider the array in side, in which, for some fixed x, the a_i, b_i and c_i are computed by the formulae

x_0	y_0	a_0	b_0	c_0
x_1	y_1	a_1	b_1	
x_2	y_2	a_2		
x_3	y_3			

$$a_i = \frac{(x_{i+1} - x)\,y_i(x) + (x - x_i)\,y_{i+1}(x)}{x_{i+1} - x_i}$$

$$b_i = \frac{(x_{i+2} - x)\,a_i(x) + (x - x_i)\,a_{i+1}(x)}{x_{i+2} - x_i}, \qquad c_i = \frac{(x_{i+3} - x)\,b_i(x) + (x - x_i)\,b_{i+1}(x)}{x_{i+3} - x_i}$$

Show that c_0 is the value of the cubic interpolating polynomial at x.

Exercise 8

Establish an iteration method for solving $f(x) = 0$ as follows. Let Q_n be the interpolating quartic polynomial for the following data and let x_{n+1} be the zero of Q_n closest to x_n .

$f(x_n)$	$f(x_{n-1})$	$f(x_{n-2})$
x_n	x_{n-1}	x_{n-2}

Sect.6 The error in Polynomial Interpolation

The use of the interpolating polynomial to obtain a value of the function for an x different from the given x's for which the polynomial was constructed is based on the assumption that in the interval under consideration the polynomial is a good approximation to the function. It is essential that we have a criterion for checking this assumption.

Theorem 6

Let $f(x)$ has continuous derivatives up to order $n+1$ in an interval $[a,b]$ and let $P_n(x)$ be the polynomial of degree $\leqslant n$ interpolates the function f at the $n+1$ distinct points $x = x_0,\ x_1,\ \ldots, x_n$ in $[a,b]$. To each $x \in [a,b]$ there corresponds a point ξ_x in $]a,b[$ such that

$$f(x) - P_n(x) = (x - x_0)(x - x_1)\ldots(x - x_n)\,\frac{f^{(n+1)}(\xi_x)}{(n+1)!}. \tag{22}$$

Proof. If x is one of the nodes of interpolation x_i, the assertion is obviously true since both sides of (22) reduce to zero. So, let x be any point other than a node. Let the auxiliary function F be defined by the equation

$$F(x) = f(x) - P_n(x) - \lambda\psi(x), \qquad \text{with} \qquad \psi(x) = (x - x_0)(x - x_1)\dots(x - x_n) \tag{23}$$

in which λ is the real number that makes $F(x) = 0$. Thus (since x is fixed)

$$\lambda = \frac{f(x) - P_n(x)}{\psi(x)} \tag{24}$$

Clearly the function $F(x)$ vanishes at $n+2$ distinct points, $x, x_0,\ x_1,\ \dots, x_n$. By Rolle's Theorem $F'(x)$ vanishes at least at $n+1$ distinct points in $]a,b[$, $F''(x)$ at n points, and so on, and in particular $F^{(n+1)}$ vanishes at least at one point ξ_x in $]a,b[$. Now

$$F^{(n+1)}(x) = f^{(n+1)}(x) - P_n^{(n+1)}(x) - \lambda\psi(x)^{(n+1)} = f^{(n+1)}(x) - (n+1)!\lambda, \tag{25}$$

because the $(n+1)^{\text{th}}$ derivative of $P_n(x)$ is zero, since the degree of $P(x)$ is $\leqslant n$, while the product $(x - x_0)(x - x_1)\dots(x - x_n)$, when multiplied out, starts with the term x^{n+1} followed by terms of lower degree. Hence
$F^{(n+1)}(x)$ vanishes, and solve for R, obtaining

$$0 = F^{(n+1)}(\xi_x) = f^{(n+1)}(\xi_x) - (n+1)!\lambda = f^{(n+1)}(\xi_x) - (n+1)!\left(\frac{f(x) - P_n(x)}{\psi(x)}\right)$$

which is the desired formula for the error (22) in disguise. □

Example 1

Let us find the error committed in approximating to $\sin(x)$ by a polynomial of fifth degree which coincides with $\sin(x)$ at the points $0°$, $5°$, $10°$, $15°$, $20°$, $25°$.
Here $f(x) = \sin(x)$, and $f^{(6)} = \sin(x + 6\frac{\pi}{2}) = \sin(x + 3\pi) = -\sin(x)$, so $f^{(6)}$ does not exceed 1 in numerical value. Since $5° = \frac{\pi}{36}$, then

$$\left|f(x) - P(x)\right| \leqslant \frac{1}{6!}x\left(x - \frac{\pi}{36}\right)\left(x - \frac{\pi}{18}\right)\left(x - \frac{\pi}{12}\right)\left(x - \frac{\pi}{9}\right)\left(x - \frac{5\pi}{36}\right)$$

Setting $x = \frac{\pi}{36} = 12°30'$ and $x = 2°$ we find, respectively

$$\left|f(x) - P(x)\right| \leqslant 0.00000000216, \qquad \left|f(x) - P(x)\right| \leqslant 0.00000001.$$

Example 2

Find the error in approximating to $f(x) = \log_{10}(x)$ by a polynomial of the 5^{th} degree which coincides with $\log_{10}(x)$ at $x = 1, 2, 3, 4, 5, 6$.

Here we find upon differentiating six times that $f^{(6)} = -\frac{5!}{x^6}\log_{10}(2)$, then

$$\left|\log_{10}(x) - P(x)\right| \leqslant \frac{\log_{10}(2)}{6s^6}(x-1)(x-2)(x-3)(x-4)(x-5)(x-6)$$

If x is in the interval $1 \leqslant x \leqslant 6$, then s also is in this interval, and in the worst case possible, namely, when $s = 1$, we have

$$\frac{\log_{10}(2)}{6s^6} = \frac{\log_{10}(2)}{6} < 0.0724$$

In particular if $x = 1.5$ we have $\left|\log_{10}(x) - P(x)\right| < 1.07$

Example 3

To show the effect of a shorter interval upon the error, consider again the function $f(x) = \log_{10}(x)$ and let us approximate by a polynomial of fifth degree coinciding with $\log_{10}(x)$ at $x = 1$, 1.01, 1.02, 1.03, 1.04, 1.05. Here we have

$$\left|\log_{10}(x) - P(x)\right| \leqslant (0.0724)(x-1)(x-1.01)(x-1.02)(x-1.03)(x-1.04)(x-1.05)$$

At $x = 1.015$ for example, we find $\left|\log_{10}(x) - P(x)\right| \leqslant 3.6 \times 10^{-13}$.

Another way to express the error function using divided difference is as follows. From the basic definition (13) and (14) we have

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

$$f[x_0, x_1, \dots, x_{n-1}, x] = f[x_0, x_1, \dots, x_n] + (x - x_n)f[x_0, x_1, \dots, x_n, x]$$

Now by successive substitution from subsequent equations, it follows

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

where $E(x)$ is the error function given by

$$E(x) = (x - x_0)(x - x_1)\dots(x - x_n)f[x_0, x_1, \dots, x_n, x]. \tag{26}$$

The approximate relation obtained by suppressing the error term $E(x)$ is exactly the expression (16) known as the Newton's interpolation formula.

Now by comparing the error expressions (22) and (26), we remark that the latter construction provides another proof of the equality (18).

Example 4

Compare the error (22) and (26) corresponding to $\sinh(x)$ interpolated at 0.0, 0.02 and 0.03.

Sect.7 Hermite Interpolation

So far, we have studied interpolation that involved only knowledge of function values. What happens if we include information about the derivatives as well?

To be specific, we require now a polynomial of least degree that interpolates a function f and its derivative f' at two distinct points, say x_0 and x_1. The polynomial sought will satisfy these

$$p(x_i) = f(x_i), \qquad p'(x_i) = f'(x_i), \qquad i = 0,1.$$

Since there are four conditions, it seems reasonable to look for a solution in $\mathbb{P}_3$, i.e. the linear space of all polynomials of degree $\leqslant 3$. An element of $\mathbb{P}_3$ has four coefficients at our disposal. Rather than writing $p(x)$ in terms of 1, x, x^2, x^3, however, let us write

$$p(x) = a + b\,(x - x_0) + c\,(x - x_0)^2 + d\,(x - x_0)^2\,(x - x_1)$$

since this will simplify the work. This leads to

$$p'(x) = b + 2c(x - x_0) + 2d(x - x_0)(x - x_1) + d(x - x_0)^2.$$

The four conditions on p can now be written in the form

$$\begin{cases} f(x_0) &= a \\ f'(x_0) &= b \\ f(x_1) &= a + bh + ch^2, \qquad h = x_1 - x_0 \\ f'(x_1) &= b + 2ch + dh^2. \end{cases} \tag{27}$$

We remark here that the problem can be solved no matter what the values of $f(x_i)$ and $f'(x_i)$ may be. In general, if values of a function f and some of its derivatives are to be interpolated by a polynomial, we shall encounter some difficulties because the linear systems of equations may be singular. A simple example will illustrate this.

Example 5

Find a polynomial p that assumes these values: $p(0) = 0$, $p(1) = 1$, $p'(1/2) = 1/2$.
Since there are three conditions, we try a quadratic $p(x) = a + bx + cx^2$. The condition $p(0) = 0$ leads to $a = 0$. The other two conditions lead to

$$\begin{cases} 1 = p(1) & = b + c \\ 1/2 = p'(1/2) & = b + c \end{cases}$$

Thus, no quadratic solves our problem. Notice that the coefficient matrix is singular.
If we now try a cubic polynomial for the same problem: $p(x) = a + bx + cx^2 + dx^3$ we discover that there exists a solution but it is not unique. We notice that $a = 0$ as before. The remaining conditions are

$$\begin{cases} 1 = & p(1) = b + c + d \\ 1/2 = & p'(1/2) = b + c + \frac{3}{4}d \end{cases} \qquad \begin{cases} d & = 2 \\ b + c & = -1 \end{cases}$$

The general problem of this type obviously has some intriguing difficulties associated with it.

Let us now consider $n+1$ distinct points. The interpolation problem is as follows:
Given the set of data points $\left(x_i, y_i, y_i'\right)$, $i = 1, \ldots, n$, it is required to determine a polynomial of least degree, say H_{2n+1} such that

$$H_{2n+1}(x_i) = y_i, \qquad H'_{2n+1}(x_i) = y'_i, \qquad i = 0, 1, \ldots, n. \tag{28}$$

This is known as the Hermite interpolation problem. We have here $2n+2$ conditions and therefore the number of coefficients to be determined is $2n+2$ and the degree of the polynomial is $2n+1$. In analogy with the previous interpolation formula, we seek a representation of the form

$$H_{2n+1}(x) = \sum_{i=0}^{n} u_i(x) y_i + \sum_{i=0}^{n} v_i(x) y'_i, \tag{29}$$

where $u_i(x)$ and $v_i(x)$ are polynomials in x of degree $2n+1$. Using condition (28), we obtain

$$\left.\begin{array}{ll} u_i(x_j) = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases} & v_i(x) = 0, \text{ for all } i, \\ u'_i(x) = 0, \text{ for all } i, & v'_i(x_j) = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases} \end{array}\right\} \tag{30}$$

Since $u_i(x)$ and $v_i(x)$ are polynomials in x of degree $2n+1$, we write

$$u_i(x) = A(x)\,[L_i(x)]^2 \qquad \text{and} \qquad v_i(x) = B(x)\,[L_i(x)]^2$$

where $L_i(x)$ are the Lagrange interpolating polynomials given by (7). It is easy to see that $A_i(x)$ and $B_i(x)$ are linear in x, i.e. of degree one. We write therefore

$$u_i(x) = (a_i x + b_i)\,[L_i(x)]^2 \qquad \text{and} \qquad v_i(x) = (c_i x + d_i)\,[L_i(x)]^2 \tag{31}$$

Using conditions (30) and (31), we obtain

$$\left.\begin{array}{r} a_i x_i + b_i = 1 \\ c_i x_i + d_i = 0 \end{array}\right\} \quad \text{and} \quad \left.\begin{array}{r} a_i + 2L_i'(x_i) = 0 \\ c_i = 1 \end{array}\right\} \implies \left\{\begin{array}{ll} a_i = -2L_i'(x_i), & b_i = 1 + 2x_i L_i'(x_i) \\ c_i = 1, & d_i = -x_i \end{array}\right.$$

Hence, expressions (31) become

$$u_i(x) = \Big[1 - 2\,(x - x_i)\,L_i'(x_i)\Big]\,[L_i(x)]^2 \qquad \text{and} \qquad v_i(x) = (x - x_i)\,[L_i(x)]^2\,. \tag{32}$$

Using the latter expressions of $u_i(x)$ and $v_i(x)$ in (29), we obtain

$$H_{2n+1}(x) = \sum_{i=0}^{n} \Big[1 - 2\,(x - x_i)\,L_i'(x_i)\Big]\,[L_i(x)]^2\,y_i + \sum_{i=0}^{n} (x - x_i)\,[L_i(x)]^2\,y_i', \tag{33}$$

which is the required Hermite interpolation formula.

Particular case: n=2

The third-order Hermite interpolating polynomial passing through the points $\big(x_i, y_i, \alpha_i\big)$, $i = 0,1$, can be obtained by putting $n = 1$ in (33). In this case, the latter expression gives

$$\begin{aligned} H_3(x) = {} & \Big[1 - 2\,(x - x_0)\,L_0'(x_0)\Big]\,[L_0(x)]^2\,y_0 + \Big[1 - 2\,(x - x_1)\,L_1'(x_1)\Big]\,[L_1(x)]^2\,y_1 \\ & + (x - x_0)\,[L_0(x)]^2\,y_0' + (x - x_1)\,[L_1(x)]^2\,y_1'. \end{aligned} \tag{H}$$

Since

$$\left\{\begin{array}{l} L_0(x) = \dfrac{x - x_1}{x_0 - x_1} = \dfrac{x_1 - x}{h_1} \\[2ex] L_1(x) = \dfrac{x - x_0}{x_1 - x_0} = \dfrac{x - x_0}{h_1} \end{array}\right. \implies \left\{\begin{array}{l} L_0'(x) = -\dfrac{1}{h_1} \\[2ex] L_1'(x) = \dfrac{1}{h_1} \end{array}\right. \qquad \text{with} \quad h_1 = x_1 - x_0.$$

Hence, (H) becomes

$$H_3(x) = \left[1 + \frac{2\,(x-x_0)}{h_1}\right]\frac{(x-x_1)^2}{h_1^2}y_0 + \left[1 + \frac{2\,(x_1-x)}{h_1}\right]\frac{(x-x_0)^2}{h_1^2}y_1$$
$$+ (x-x_0)\,\frac{(x-x_1)^2}{h_1^2}y_0' + (x-x_1)\,\frac{(x-x_0)^2}{h_1^2}y_1' \tag{G}$$

which is the required Hermite formula.

Exercise 9

Let $f(x) = 2^x$ and $x_0 = 1$, $x_1 = 2$ and $x_2 = 3$. Use divided differences to compute the interpolation polynomial $p(x)$ satisfying:

1. $p(x_i) = f(x_i)$, $i = 0,1,2$ and $p'(x_1) = f'(x_1)$.
2. $p(x_i) = f(x_i)$, $i = 0,1,2$ and $p'(x_j) = f'(x_j)$, $j = 0,1$.

Exercise 10

Let $\epsilon \in]0,1[$ and $f \in \mathscr{C}([0,1])$. Set $f(0) = a$ and $f(1) = b$.

1. Compute the interpolating polynomial P_ϵ of f at $0, \epsilon, 1$.
2. Show that for any $s \in [0,1]$ we have

$$\lim_{\epsilon \to 0} P_\epsilon(x) = \big(b - a - f'(0)\big)\,x^2 + xf'(0) + a := Q(x)$$

3. Show that $Q(x)$ is the unique polynomials of degree $\leqslant 2$ satisfying

$$Q(0) = a, \qquad Q'(0) = f'(0), \qquad Q(1) = f(1).$$

Another Technique of Construction

Let us consider the interpolating polynomial for a function f on the $2n+2$ abscissas x_0, x_1, ..., x_n and $x_0 + h$, $x_1 + h$, ...$x_n + h$ and let f' exist on an interval containing all those abscissas.
This interpolating polynomial is of the form

$$P_{2n+1}(x) = \sum_{i=0}^{n} \big(f(x_i)a_i(x;h) + f(x_i+h)b_i(x;h)\big) \tag{34}$$

where a_i and b_i are polynomials on x that depend on h. The polynomial $P_{2n+1}(x)$ can be rearranged in the following way

$$\sum_{i=0}^{n} f(x_i)\Big(a_i(x;h) + b_i(x;h)\Big) + h\sum_{i=0}^{n}\left(\frac{f(x_i+h) - f(x_i)}{h}\right) b_i(x;h)$$

Let $h \to 0$, to give

$$P_{2n+1}(x) = \sum_{i=0}^{n}\Big(f(x_i)u_i(x) + f'(x_i)v_i(x)\Big).$$

We can show that

$$u_i(x) = \Big(1 - 2L_i'(x_i)(x - x_i)\Big)\left(L_i(x)\right)^2 = \lim_{h\to 0}\Big(a_i(x;h) + b_i(x;h)\Big) \tag{35}$$

$$v_i(x) = (x - x_i)\left(L_i(x)\right)^2 = \lim_{h\to 0} hb_i(x;h) \tag{36}$$

Since $L_i(x)$ is of degree n, it is clear from (35) and (36) that $u_i(x)$ and $v_i(x)$ are of degree $2n+1$, and it then follows that the polynomial $P_{2n+1}(x)$ in (34) is of degree $2n+1$.
It is, however, straightforward to verify from (35) and (36) that we have

$$u_i(x_k) = \delta_{ik}, \quad u_i'(x_k) = 0, \qquad \text{and} \qquad v_i(x_k) = 0, \quad v_i'(x_k) = \delta_{ik}.$$

In Terms of Divided Differences

Like the Lagrange form in ordinary interpolation, this form of the Hermite polynomial is not conducive to efficient computation. A variation of the Newton approach, including the use of divided difference tables, can be constructed. According to (21), we shall use the notation

$$f[a,a] = f'(a), \qquad f[a,a,b] = \frac{f'(a) - f[a,b]}{a-b}, \qquad f[a,a,a] = f''(a), \dots$$

To use this result to generate the Hermite polynomial, first suppose the distinct numbers x_0, x_1, ..., x_n are given together with the values of f and f' at these numbers. Define a new sequence z_0, z_1, ..., z_{2n+1} by

$$z_{2k} = z_{2k+1} = x_k, \quad \text{for each} \quad k = 0, 1, \dots n.$$

Now construct the divided-difference table using the variables z_0, z_1, ..., z_{2k+1}. Since $z_{2k} = z_{2k+1} = x_k$ for each k, we have $f[z_{2k}, z_{2k+1}] = f[x_k, x_k] = f'(x_k)$. This provides us with an alternative, and more easily evaluated, method for deter mining coefficients of the Hermite polynomial:
If $f \in \mathscr{C}([a,b])$ and x_0, x_1, ..., x_n are distinct in $[a,b]$, then

$$H_{2n+1}(x) = f[z_0] + \sum_{k=0}^{2n+1} f[z_0, z_1, \dots, z_k](z - z_0)(z - z_1)\dots(z - z_{k-1}) \tag{37}$$

where $z_{2k} = z_{2k+1} = x_k$ and $f[z_{2k}, z_{2k+1}] = f'(x_k)$, for each $k = 0, 1, \dots, n$.

z_k	$f_0(z_k)$	1st Diff	2nd Diff	3rd Diff		
z_0	$f(z_0)$					
		$f[z_0,z_1] = f'(x_0)$				
z_1	$f(z_0)$		$f[z_0,z_1,z_2]$			
		$f[z_1,z_2]$		$f[z_0,z_1,z_2,z_3]$		
z_2	$f(z_2)$		$f[z_1,z_2,z_3]$		$f[z_0,\dots,z_4]$	
		$f[z_2,z_3] = f'(x_1)$		$f[z_1,z_2,z_3,z_4]$		$f[z_0,\dots,z_5]$
z_3	$f(z_3)$		$f[z_2,z_3,z_4]$		$f[z_1,\dots,z_5]$	
		$f[z_3,z_4]$		$f[z_2,z_3,z_4,z_5]$		
z_4	$f(z_4)$		$f[z_3,z_4,z_5]$			
		$f[z_4,z_5] = f'(x_2)$				
z_5	$f(z_5)$					

Exercise 11

1. Use the Newton divided difference method to obtain a quartic polynomial that takes the values give in the table.
2. Find a quintic polynomial that takes the values in the table and, in addition, satisfies $p(4) = -21$.

x	1	2	3
$p(x)$	3	-5	7
$p'(x)$	-2	6	

Exercise 12

Compute in two different ways, the Hermite interpolating polynomial $H_3(x)$ of degree 3, interpolating the function $f(x) = x|x|$ and satisfying $H(\pm 1) = f(\pm 1)$, $H(0) = f(0)$ and $H'(0) = f'(0)$.

For the error bound of the Hermite interpolation, we have the following result.

Theorem 7

Let x_0, x_1, ..., x_n distinct nodes in the interval $[a, b]$ and $f \in \mathscr{C}^{2n+2}([a, b])$. If $P(x)$ is a polynomials of degree $\leqslant (2n+1)$ such that $P(x_i) = f(x_i)$ and $P'(x_i) = f'(x_i)$, $\forall\, 0 \leqslant i \leqslant n$, then, for each $x \in [a, b]$, there exists a number ξ in $]a, b[$ (generally unknown), such that

$$f(x) = P(x) + \frac{f^{(2n+2)}(\xi)}{(2n+2)!}(x - x_0)^2 (x - x_1)^2 \dots (x - x_n)^2. \tag{38}$$

Proof. The proof follows the same techniques we used in proving Theorem Sect.6. If x is one of the interpolation points, the result trivially holds. We thus fix x as a non-interpolation point and define

$$\phi(x) = f(x) - P(x) - \lambda\pi(x), \qquad \pi(x) = (x - x_0)^2(x - x_1)^2 \ldots (x - x_n)^2$$

in which λ is the real number that makes $\phi(x) = 0$. Thus (since x is fixed)

$$\lambda = \frac{f(x) - P(x)}{\pi(x)} \tag{39}$$

Clearly the function $\phi(x)$ vanishes at $n + 2$ distinct points, $x, x_0,\ x_1,\ \ldots, x_n$. By Rolle's theorem we know that $\phi'(x)$ vanishes at least at $n + 1$ distinct points in $]a, b[$ that are different than $x, x_0,\ x_1,\ \ldots, x_n$. Also $\phi'(x)$ vanishes at $x, x_0,\ x_1,\ \ldots, x_n$ which means that $\phi'(x)$ vanishes at least at $2n+2$ zeros in $]a, b[$. Similarly, Rolle's theorem implies that $\phi''(x)$ has at least $2n+1$ zeros in $]a, b[$, and by induction $\phi^{(2n+2)}$ vanishes at least at one point in $]a, b[$, say ξ. Hence

$$0 = \phi^{(2n+2)}(\xi) = f^{(2n+2)}(\xi) - P^{(2n+2)}(\xi) - \lambda\pi^{(2n+2)}(\xi) = f^{(2n+2)}(\xi) - (2n+2)!\lambda,$$

because the leading term in $\pi(x)$ is x^{2n+2}, then $\pi^{(2n+2)}(\xi) = (2n+2)!$. Also $(2n+2)^{\text{th}}$ derivative of $P(x)$ is zero, since the degree of $P(x)$ is $\leqslant 2n+1$. Recall that x is arbitrary (non-interpolation) point and hence the desired result is obtained. □

Sect.8 Spline Interpolation

In previous sections, we introduced 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.
In this section, we introduce the interpolation using piecewise polynomials. This will effectively prevent the oscillation.
The goal in this section is to develop a new interpolation so that

✓ it uses piecewise polynomials (to avoid oscillation).
✓ It is continuously differentiable over the entire domain (to ensure the smoothness).
✓ It requires no specific derivative information of the original function, except perhaps at the two endpoints of the interval (minimum information from original function).

Definition 1

A spline function of degree k having knots to x_0, x_1, ..., x_n is a function S such that:

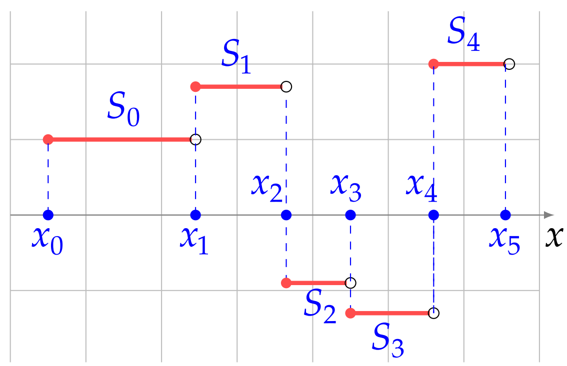
ⓐ On each interval $[x_i, x_{i+1}]$, S is a polynomial of degree $\leqslant k$.

ⓑ S has a continuous $(k-1)^{st}$ derivative on $[x_0, x_n]$, that is to say $S^{(r)}$ is continuous on $[x_0, x_n]$ for $0 \leqslant r \leqslant k-1$.

In other words, S is a continuous piecewise polynomial of degree at most k having continuous derivatives of all orders up to $k-1$.

Splines of degree 0 are piecewise constants. A spline of degree 0 can be given explicitly in the form

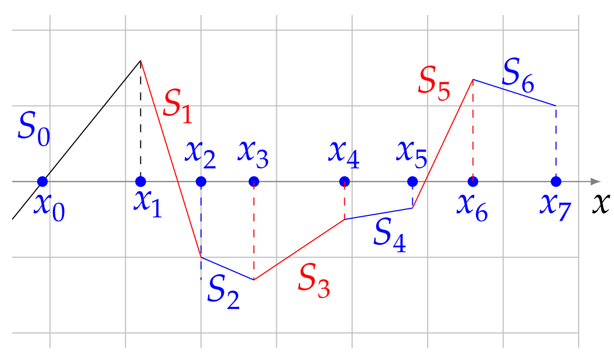
$$S(x) = \begin{cases} S_0(x) = c_0, & x \in [x_0, x_1[\\ S_1(x) = c_1, & x \in [x_1, x_2[\\ \quad \vdots & \vdots \\ S_{n-1}(x) = c_{n-1}, & x \in [x_{n-1}, x_n[\end{cases}$$



Linear Splines

Linear Splines are polynomials of degree $\leqslant 1$. Figure in below shows the graph of a typical spline function of degree 1, with eight knots. Such functions can be defined explicitly by writing

$$S(x) = \begin{cases} S_0(x) = a_0 + b_0(x - x_0), & x \in [x_0, x_1[\\ S_1(x) = a_1 x + b_1(x - x_1), & x \in [x_1, x_2[\\ \quad \vdots & \vdots \\ S_i(x) = a_i + b_i(x - x_i), & x \in [x_i, x_{i+1}[\\ \quad \vdots & \vdots \\ S_{n-1}(x) = a_{n-1} + b_{n-1}(x - x_{n-1}), & x \in [x_{n-1}, x_n] \end{cases}$$



The representation of splines as polynomials in $x - x_i$ rather than x will lead to simpler formulas. The coefficients a_i and b_i may vary with interval index i, but continuity at x_{i+1} requires that $S_{i+1}(x_{i+1}) =$

$S_i(x_{i+1})$. That is, the coefficients must satisfy

$$a_{i+1} + b_{i+1}(x_{i+1} - _{i+1}) = a_{i+1} = a_i + b_i(x_{i+1} - x_i) \implies b_i = \frac{a_{i+1} - a_i}{x_{i+1} - x_i}, \quad 0 \leqslant i \leqslant n-2.$$

Thus each coefficients a_i equals the corresponding function value $y_i = S(x_i) = a_i$ and b_i equals the slope connecting the neighboring data points (x_i, y_i) and (x_{i+1}, y_{i+1}).
In summary: The linear Spline for the data (x_i, y_i), $0 \leqslant i \leqslant n$ must satisfy

$$S_i(x) = y_i + \Big(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\Big)(x - x_i) = y_i + f[y_{i+1}, y_i](x - x_i), \qquad 0 \leqslant i \leqslant n-1. \tag{40}$$

To evaluate $S(x)$ at a point $x^* \in [x_0, x_n]$, find i such that $x_i \leqslant x \leqslant x_{i+1}$, then according to (40) set

$$S(x^*) = y_i + \Big(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\Big)(x^* - x_i).$$

Example 6

Given the set of data points $(1, -8)$, $(2, -1)$ and $(3, 5)$ satisfying the function $y = f(x)$, find the linear splines satisfying the given date. Determine the approximate values of $f(2.5)$ and $f'(2)$.

Solution ✍

Let the given points be A, B and C, respectively. Now, equation of AB and BC are, respectively,

$$s_1(x) = -8 + 7(x-1) = 7x - 15, \qquad s_2(x) = -1 + 6(x-2) = 6x - 13.$$

Since $x = 2.5 \in [2, 3[$, then $y(2.5) \approx s_2(2.5) = 2$.

It is easy to check that the splines $S_i(x)$ of latter example, are continuous on $[1, 3]$ but their slopes are discontinuous. This is clearly a drawback of linear splines and therefore we next discuss quadratic splines which assume the continuity of the slopes in addition to that of the function.
Before moving to the next section, a question that one may ask is about the goodness of fit when we interpolate a function by a linear spline. The answer is found in the following theorem.

Theorem 8: Linear Spline Accuracy

Suppose f is twice differentiable and continuous on the interval $[a, b]$. If $P(x)$ is a linear spline interpolating f at the knots $a = x_0 < x_1 < \cdots < x_n = b$, then

$$\big|f(x) - P(x)\big| \leqslant \frac{1}{8}Mh^2, \quad a \leqslant x \leqslant b \tag{41}$$

where $h = \max_i (x_{i+1} - x_i)$, and M denotes the maximum of $|f''(x)|$ on $]a, b[$.

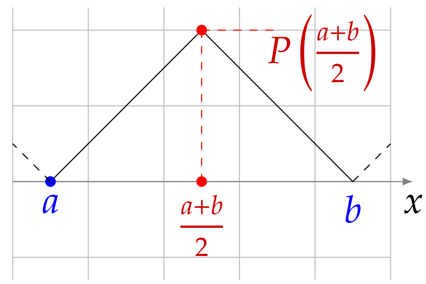
Proof. With $n+1$ nodes, it has been shown in section Sect.6 that

$$f(x) - P(x) = \frac{1}{(n+1)!}(x-x_0)(x-x_1)\dots(x-x_n)f^{(n+1)}(\xi).$$

On the interval $[a,b]$ we have $n=2$, then

$$f(x) - P(x) = \frac{1}{2!}(x-a)(x-b)f''(\xi)$$

for some $\xi \in]a,b[$. Now, since f is continuous and twice differentiable on the closed interval $[a,b]$, then $|f''(x)| \leqslant M$ on $]a,b[$.



It is clear that

$$\max_x |x-a||x-b| = \left|\frac{a+b}{2} - a\right|\left|\frac{a+b}{2} - b\right| = \frac{(a-b)^2}{4}$$

It follows that

$$|f(x) - P(x)| \leqslant \frac{1}{2!}M\frac{(a-b)^2}{4} = \frac{1}{8}Mh^2. \qquad \square$$

Quadratic Splines

For piecewise linear interpolation, we choose two points (x_i, x_{i+1}) and (y_i, y_{i+1}) in the subinterval $[x_i, x_{i+1}]$ and draw a line through those two points to interpolate the data. This approach is easily extended to construct the quadratic splines. Instead of choosing two points, we choose three points in the subinterval $[x_i, x_{i+1}]$ and pass a second-degree polynomial through these points. We shall show that there is only one such polynomial. Therefore, to construct a quadratic spline $Q(x)$, we first define a quadratic function in each subinterval $[x_i, x_{i+1}]$ by

$$Q(x) = a_i + b_i(x-x_i) + c_i(x-x_i)^2 \tag{42}$$

where a_i, b_i and c_i are three constants to be determined. Next, by we use the conditions from the definition of general spline that must satisfy we can determine $Q(x)$ (with one more condition on c_0).

Another Technique of Constructing Quadratic Splines

Let $S_i(x)$ be the quadratic spline approximating the function $y = f(x)$ in the interval $[x_i, x_{i+1}]$ with $x_{i+1} - x_i = h_i$. Let $S_i(x)$ and $S_i'(x)$ be continuous on $[x_0, x_n]$ satisfying $S_i(x_i) = y_i$ and $S_i'(x_i) = m_i$, $i = 0, \dots, n$.

Since $S_i(x)$ is quadratic, then a shorter way of driving a such spline, is to remark that $S_i'(x)$ is linear. Therefore, on the subinterval $[x_i, x_{i+1}]$ we may write

$$S_i'(t) = \frac{1}{h_i}\Big[\,(t_{i+1} - x)\,m_i + (t - x_i)\,m_{i+1}\Big] = m_i + \Big(\frac{m_{i+1} - m_i}{h_i}\Big)\,(t - x_i)\,, \quad \forall\, t \in [x_i, x_{i+1}]$$

Integration of the latter equation from x_i to x: $x_i \leqslant x \leqslant x_{i+1}$, yields

$$S_i(x) = y_i + m_i\,(x - x_i) + \frac{1}{2} f[m_i, m_{i+1}]\,(x - x_i)^2 \tag{43}$$

Imposing next the continuity condition $S_i(x_{i+1}) = S_{i+1}(x_{i+1}) = y_{i+1}$, $0 \leqslant i \leqslant n-1$, provides a new set of n equations

$$y_{i+1} = y_i + m_i\,(x_{i+1} - x_i) + \frac{1}{2}\left(\frac{m_{i+1} - m_i}{x_{i+1} - x_i}\right)(x_{i+1} - x_i)^2 = y_i + \frac{h_i}{2}\,(m_{i+1} + m_i) \tag{44}$$

Obviously, to determine $S_i(x)$, we should first compute m_i from the following recursion obtained by rearranging (44)

$$m_{i+1} = -m_i + 2f[y_i, y_{i+1}], \quad 0 \leqslant i \leqslant n-1, \qquad m_0 \ \text{arbitrary}. \tag{45}$$

To obtain a unique determination of the interpolating quadratic spline, there appears to be a deficit of one further constraint! There is a variety of ways of providing an additional condition. One may impose specific values on m_0. For instance, the condition $S'(x_0) = 0$ is called the natural quadratic spline. It suffices next to determine the equations of the quadratic spline over the interval $[a, b]$, from equation (43).

Example 7

Determine the natural quadratic splines satisfying the data given in the previous example. Find also an approximate values of $f(2.5)$ and $f'(2)$.

Solution

We have $n = 2$ and $h = 1$. With $m_0 = 0$, we have from (45) that

$$\begin{aligned} i = 0: &\quad m_1 = -m_0 + 2f[y_0, y_1] = 0 + 2\big(-1 + 8\big) = 14 \\ i = 1: &\quad m_2 = -m_1 + 2f[y_1, y_2] = -14 + 2\big(5 + 1\big) = -2 \end{aligned}$$

Now, (43) gives the quadratic splines

$$i=0: \quad S_0(x) = y_0 + m_0\,(x-x_0) + \frac{1}{2} f[m_0,m_1]\,(x-x_0)^2 = -8 + 7\,(x-1)^2$$

$$\begin{aligned} i=1: \quad S_1(x) &= y_1 + m_1\,(x-x_1) + \frac{1}{2} f[m_1,m_2]\,(x-x_1)^2 \\ &= -1 + 14\,(x-2) + \frac{1}{2}\big(-2-14\big)\,(x-2)^2 = -1 + 14\,(x-2) - 8\,(x-2)^2 \end{aligned}$$

Since $2.5 \in [2,3]$, then $f(2.5) \approx S_2(2.5) = -1 + 14(0.5) - 8(0.5)^2 = 4$.

Example 8

For what value of k is the following a spline function?

$$\begin{cases} -x^2 + ax + b, & 0 \leqslant x \leqslant 1 \\ (b+3)x^2 - ax, & 1 \leqslant x \leqslant 2 \end{cases} \qquad \begin{cases} -(ax)^2 + ax + b, & 0 \leqslant x \leqslant 1 \\ (b+2)x^2 - ax + a^2, & 1 \leqslant x \leqslant 2 \end{cases} \qquad \begin{cases} ax + e, & -2 \leqslant x \leqslant 1 \\ bx^2 + cx, & 1 \leqslant x \leqslant 2 \\ dx^2, & 2 \leqslant x \leqslant 7 \end{cases}$$

Cubic Splines

Cubic splines allow for smoother data fitting and they are most frequently used in applications (as for quadratic splines, the discontinuity in the second derivative, and therefore the curve may not be visually smooth enough). It can be proved that cubic spline functions are among the best interpolation functions that are available at an acceptable computational cost. In this case, we join cubic polynomials together in such a way that the resulting spline function has its first and second derivatives continuous everywhere in the interval $[a,b]$. Consider a set of data points:

x_0	x_1	x_2	... ...	x_n
$f(x_0)$	$f(x_1)$	$f(x_2)$	... ...	$f(x_n)$

We construct a cubic spline interpolant $S(x)$ for f satisfies

1. On each subinterval $[x_j, x_{j+1}]$, $S(x)$ is a cubic polynomial denoted $S_j(x)$ for $j = \overline{0, n-1}$.
2. For each $j = 0, 1, \dots, n$, we have $S_j(x_j) = f(x_j)$.
3. S, S' and S'' are continuous, i.e. for each $j = 0, 1, \dots, n-2$, we have

   (a) $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$ (b) $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1})$ (c) $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$

The function S is of n different cubic polynomials each with four coefficients, so there are a total of $4n$ unknowns. Interpolation provides $n+1$ equations. continuity of spline and its first two derivatives contribute an additional $3(n-1) = 3n-3$ equations. Added up, the number of conditions is two short of the number of unknowns leaving two degrees of freedom for calculating a valid cubic interpolating spline. In principle, the values of any two of the unknowns can be fixed in order to make the number of unknowns equal to the number of conditions. It is advantageous though, to use the two degrees of freedom for choosing the behavior of the spline at the edges of the interval x_0 and x_n. These boundary conditions can be chosen in a variety of ways.

Choices for boundary conditions

- Natural Spline: Set second derivatives at the boundaries $S_0''(x_0) = S_{n-1}''(x_n) = 0$.
  Generalized natural spline $S_0''(x_0) = y_0''$ and $S_{n-1}''(x_n) = y_n''$.
- Complete Spline (Clamped boundary conditions): Set second derivatives at the boundaries $S_0'(x_0) = y_0'$ and $S_{n-1}'(x_n) = y_n'$.
- Not-a-knot boundary conditions: Effectively eliminate x_1 and x_{n-1} from the list of knots: $S_0''' = S_1'''$ and $S_{n-2}''' = S_{n-1}'''$ (S_0 and S_1 are the same functions then, and so are S_{n-2} and S_{n-1})
- Periodic boundary conditions: $S_0'(x_0) = S_{n-1}'(x_n)$ and $S_0''(x_0) = S_{n-1}''(x_n)$.

Sect.8.1 First Technique

Consider a cubic spline interpolating data points (x_k, y_k) for a given $n+1$ knots x_k. For convenience, we use the abbreviation $h_k = x_{k+1} - x_k$. In each interval $[x_k, x_{k+1}]$ we have a third order polynomial

$$S_k(x) = a_k + b_k(x - x_k) + c_k(x - x_k)^2 + d_k(x - x_k)^3, \quad 0 \leqslant k \leqslant n-1. \tag{46}$$

$$S_k'(x) = b_k + 2c_k(x - x_k) + 3d_k(x - x_k)^2, \tag{47}$$

$$S_k''(x) = 2c_k + 6d_k(x - x_k) \tag{48}$$

Setting $x = x_k$ in (46) yields

$$y_k = S(x_k) = a_k, \quad 0 \leqslant k \leqslant n-1. \tag{49}$$

Putting $x = x_k$ into the second derivative expression (48) we obtain

$$y_k'' = S''(x_k) = 2c_k \quad \Longrightarrow \quad c_k = \frac{y_k''}{2}, \quad 0 \leqslant k \leqslant n-1. \tag{50}$$

Now the continuous of th second derivative (48) gives

$$S''_{k+1}(x_{k+1}) = 2c_{k+1}$$
$$S''_k(x_{k+1}) = 2c_k + 6d_k(x_{k+1} - x_k) = 2c_k + 6d_kh_k$$

Note that these relations only hold for $0 \leqslant k \leqslant n-2$ for which we deduce

$$2c_{k+1} = 2c_{k+1} + 6d_kh_k \quad \Longrightarrow \quad d_k = \frac{2c_k - 2c_{k+1}}{6h_k} = \frac{y''_{k+1} - y''_k}{6h_k}, \quad 0 \leqslant k \leqslant n-2.$$

In addition, the continuity of the second derivative involves for $k = n-1$

$$S''_n(x_n) = y''_n = S''_{n-1}(x_n) = 2c_{n-1} + 6d_{n-1}h_{n-1} \quad \Longrightarrow \quad d_{n-1} = \frac{y''_n - y''_{n-1}}{6h_{n-1}}$$

Therefore, the coefficient d_k can be written as

$$d_k = \frac{y''_{k+1} - y''_k}{6h_k}, \quad 0 \leqslant k \leqslant n-1. \tag{51}$$

Next, the continuity condition $S_k(x_{k+1}) = S_{k+1}(x_{k+1})$ will provide an expression for b_k. Indeed, the condition demands the equality of the following two equations

$$\left.\begin{aligned} S_{k+1}(x_{k+1}) &= y_{k+1} = a_{k+1} \\ S_k(x_{k+1}) &= a_k + b_kh_k + c_kh_k^2 + d_kh_k^3 \end{aligned}\right\} \Longrightarrow y_{k+1} = y_k + b_kh_k + c_kh_k^2 + d_kh_k^3, \quad k = 0, \dots, n-2$$

By the condition $S_{n-1}(x_n) = y_n$ covers the edge case, which extends the latter equation to all spline segments $0 \leqslant k \leqslant n-1$. By solving this equation for b_k yields

$$b_k = \frac{y_{k+1} - y_k}{h_k} - c_kh_k - d_kh_k^2, \qquad k = 0, \dots, n-1.$$

Finally, by inserting the expression for c_k (50) and d_k (51) we obtain the solution for b_k in terms the second derivative

$$b_k = \frac{y_{k+1} - y_k}{h_k} - \frac{2y''_k + y''_{k+1}}{6}h_k, \qquad k = 0, \dots, n-1. \tag{52}$$

In summary, the equations describing how to get the spline coefficients are

$$\begin{cases} a_k = y_k, & d_k = \dfrac{y''_{k+1} - y''_k}{6h_k}, \\ c_k = \dfrac{y''_k}{2}, & b_k = \dfrac{y_{k+1} - y_k}{h_k} - \dfrac{2y''_k + y''_{k+1}}{6}h_k, \qquad k = 0, \dots, n-1. \end{cases} \tag{53}$$

Notice that we haven't used the continuity of the first derivative yet. It allows to construct a linear equation in the second derivatives y_k''. Let's start first by shifting this conditions as follows

$$S_{k-1}'(x_k) = S_k'(x_k), \qquad k = 1, \dots, n-1. \tag{54}$$

According to (47), the continuity at the inner knots read

$$\left.\begin{aligned} S_k'(x_k) &= b_k \\ S_{k-1}'(x_k) &= b_{k-1} + 2c_{k-1}h_{k-1} + 3d_{k-1}h_{k-1}^2 \end{aligned}\right\} \overset{(54)}{\Longrightarrow} \quad b_k = b_{k-1} + 2c_{k-1}h_{k-1} + 3d_{k-1}h_{k-1}^2 \tag{55}$$

Using the expressions of a_k, b_k, c_k and d_k via (53), the latter equation for b_k (55) becomes

$$\frac{y_{k+1} - y_k}{h_k} - \frac{2y_k'' + y_{k+1}''}{6}h_k = \frac{y_k - y_{k-1}}{h_{k-1}} - \frac{2y_{k-1}'' + y_k''}{6}h_{k-1} + y_{k-1}''h_{k-1} + 3\frac{y_k'' - y_{k-1}''}{6}h_{k-1}$$

which maybe written after simplification as

$$h_k y_{k+1}'' + 2\big(h_k + h_{k-1}\big)y_k'' + h_{k-1}y_{k-1}'' = 6\left(\frac{y_{k+1} - y_k}{h_k} - \frac{y_k - y_{k-1}}{h_{k-1}}\right), \quad k = 1, \dots, n-1. \tag{56}$$

These are $n-1$ equations for $n+1$ unknown y_0'', y_1'', ..., y_n''. Keeping in mind that the boundary values y_0'' and y_n'' will be used to implement the spline boundary conditions later

$$2(h_0 + h_1)y_1'' + h_1 y_2'' = 6\left(\frac{y_2 - y_1}{h_1} - \frac{y_1 - y_0}{h_0}\right) - h_0 y_0'', \tag{57}$$

$$h_{k-1}y_{k-1}'' + 2\big(h_{k-1} + h_k\big)y_k'' + h_k y_{k+1}'' = 6\left(\frac{y_{k+1} - y_k}{h_k} - \frac{y_k - y_{k-1}}{h_{k-1}}\right), \quad k = 2, \dots, n-2 \tag{58}$$

$$h_{n-2}y_{n-2}'' + 2\big(h_{n-2} + h_{n-1}\big)y_{n-1}'' = 6\left(\frac{y_n - y_{n-1}}{h_{n-1}} - \frac{y_{n-1} - y_{n-2}}{h_{n-2}}\right) - h_{n-1}y_n'' \tag{59}$$

The two boundary values y_0'' and y_n'' appear on the right hand side because they can be set arbitrarily to implement the different boundary conditions. The system (56)- (58) can be written in matrix form $Ay = b$ with a square matrix A of order $(n-1)$ and two vectors y and b as follows

$$A = \begin{bmatrix} 2(h_0 + h_1) & h_1 & 0 & \cdots & 0 \\ h_1 & 2(h_1 + h_2) & h_2 & \ddots & \vdots \\ 0 & h_2 & 2(h_2 + h_3) & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & h_{n-2} \\ 0 & \cdots & 0 & h_{n-2} & 2(h_{n-2} + h_{n-1}) \end{bmatrix}$$

$$y = \begin{bmatrix} y_1'' \\ y_2'' \\ y_3'' \\ \vdots \\ y_{n-2}'' \\ y_{n-1}'' \end{bmatrix}, \qquad b = \begin{bmatrix} 6\left(\frac{y_2-y_1}{h_1} - \frac{y_1-y_0}{h_0}\right) - h_0 y_0'' \\ 6\left(\frac{y_3-y_2}{h_2} - \frac{y_2-y_1}{h_1}\right) \\ \vdots \\ 6\left(\frac{y_{n-1}-y_{n-2}}{h_{n-2}} - \frac{y_{n-2}-y_{n-3}}{h_{n-3}}\right) \\ 6\left(\frac{y_n-y_{n-1}}{h_{n-1}} - \frac{y_{n-1}-y_{n-2}}{h_{n-2}}\right) - h_{n-1} y_n'' \end{bmatrix}$$

It is worthy to notice that the boundary conditions will only change the first and last equations (except for the periodic boundary conditions which result in an $n \times n$ strictly diagonally dominant matrix). In those cases the first and last rows of the matrix A will be modified and in all cases its strict diagonal dominance property will be preserved.

Note also that the matrix $A = \left(a_{ij}\right)$ has the following properties

- A is symmetric, since $a_{ij} = a_{ji}$.
- A is tridiagonal, since $a_{ij} = 0$ for all i, j with $|i - j| > 1$.
- A is strictly diagonally dominant, i.e. $|a_{ii} > |\sum_{j \neq i} |a_{ij}|$, since

$$\left|2\left(h_{k-1} + h_k\right)\right| > |h_{k-1}| + |h_k|$$

Under these conditions, the system $Ay = b$ has a unique solution.

Cubic spline equations for specific boundary conditions

ⓐ Generalized Natural Spline

A natural spline has boundary values $y_0'' = y_n'' = 0$ which can directly omit the terms involving y_0'' and y_n'' from the vector b. In the generalized case, we will allow any values for the second derivatives

$$S_0''(x_0) = y_0'', \qquad S_{n-1}''(x_n) = y_n''$$

with y_0'' and y_n'' not necessarily equal to zero. In this case, when using the matrix inversion, the solution reads $A_{gn}y = b_{gn}$ where

$$
\begin{array}{rcl}
 & y_0'' = y_0'' & \\
y = & \begin{bmatrix} y_1'' \\ y_2'' \\ y_3'' \\ \vdots \\ y_{n-2}'' \\ y_{n-1}'' \end{bmatrix} = A_{gn}^{-1}.b & \\
 & y_n'' = y_n'' &
\end{array}
$$

ⓑ Spline with clamped boundary conditions

The boundary values in this case fix the spline's slope at the interval boundaries

$$S_0'(x_0) = y_0', \qquad S_{n-1}'(x_n) = y_n'$$

In order to use the system of linear equations (57)- (59) we need to find expressions for y_0'' and y_n''. The expression (47) of the first derivative S_n' at x_0 gives $S_0'(x_0) = y_0' = b_0$, and from the b_k's expression (53) we get

$$y_0' = b_0 = \frac{y_1 - y_0}{h_0} - \frac{2y_0'' + y_1''}{6}h_0$$

Solving for y_0'' and simplifying we obtain

$$y_0'' = \frac{3}{h_0}\left(\frac{y_1 - y_0}{h_0} - y_0'\right) - \frac{1}{2}y_1'' \tag{60}$$

We use this term for eliminating y_0'' from the system of linear equations (57)- (59). When plugging (60) into the equation (57) with $k = 1$, it results

$$\left(\frac{3}{2}h_0 + 2h_1\right)y_1'' + h_1 y_2'' = 6\left(\frac{y_2 - y_1}{h_1} - \frac{y_1 - y_0}{h_0}\right) - 3\left(\frac{y_2 - y_1}{h_1} - y_0'\right) \tag{61}$$

Now, the expression (47) of the first derivative S_n' gives at the end of the interval $S_{n-1}'(x_n) = y_n'$

$$S_{n-1}'(x_n) = b_{n-1} + 2c_{n-1}h_{n-1} + 3d_{n-1}h_{n-1}^2 = y_n'$$

After substituting the expression b_{n-1}, c_{n-1} and d_{n-1} using (53), the latter expression yields

$$y_n'' = \frac{3}{h_{n-1}}\left(y_n' - \frac{y_n - y_{n-1}}{h_{n-1}}\right) - \frac{1}{2}y_{n-1}''$$

Now we use the latter expression to eliminate y''_n form (59) and obtain, after simplification

$$h_{n-2}y''_{n-2} + \left(2h_{n-2} + \frac{3}{2}h_{n-1}\right)y''_{n-1} = 6\left(\frac{y_n - y_{n-1}}{h_{n-1}} - \frac{y_{n-1} - y_{n-2}}{h_{n-2}}\right) - 3\left(y'_n - \frac{y_n - y_{n-1}}{h_{n-1}}\right) \tag{62}$$

We shall use next the (61)-(62) to compose the matrix and vectors for a cubic spline with clamped boundary conditions as $A_{cl}y = b_{cl}$ where

$$A_{cl} = \begin{bmatrix} \left(\frac{3}{2}h_0 + 2h_1\right) & h_1 & 0 & \cdots & 0 \\ h_1 & 2(h_1+h_2) & h_2 & \ddots & \vdots \\ 0 & h_2 & 2(h_2+h_3) & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & h_{n-2} \\ 0 & \cdots & 0 & h_{n-2} & \left(2h_{n-2} + \frac{3}{2}h_{n-1}\right) \end{bmatrix}$$

$$y = \begin{bmatrix} y''_1 \\ y''_2 \\ y''_3 \\ \vdots \\ y''_{n-2} \\ y''_{n-1} \end{bmatrix}, \qquad b_{cl} = \begin{bmatrix} 6\left(\frac{y_2-y_1}{h_1} - \frac{y_1-y_0}{h_0}\right) - 3\left(\frac{y_1-y_0}{h_0} - y'_0\right) \\ 6\left(\frac{y_3-y_2}{h_2} - \frac{y_2-y_1}{h_1}\right) \\ \vdots \\ 6\left(\frac{y_{n-1}-y_{n-2}}{h_{n-2}} - \frac{y_{n-2}-y_{n-3}}{h_{n-3}}\right) \\ 6\left(\frac{y_n-y_{n-1}}{h_{n-1}} - \frac{y_{n-1}-y_{n-2}}{h_{n-2}}\right) - 3\left(y'_n - \frac{y_n-y_{n-1}}{h_{n-1}}\right) \end{bmatrix}$$

When using the matrix inversion the solution reads

$$\begin{aligned} y''_0 &= y''_0 = \frac{3}{h_0}\left(\frac{y_1-y_0}{h_0} - y'_0\right) - \frac{1}{2}y''_1 \\ y = \begin{bmatrix} y''_1 \\ y''_2 \\ y''_3 \\ \vdots \\ y''_{n-2} \\ y''_{n-1} \end{bmatrix} &= A_{cl}^{-1}.b_{cl} \\ y''_n &= y''_n = \frac{3}{h_{n-1}}\left(y'_n - \frac{y_n-y_{n-1}}{h_{n-1}}\right) - \frac{1}{2}y''_{n-1} \end{aligned}$$

Note that y''_0 and y''_n can be calculated only after the linear system has been solved because y''_1 and y''_{n-1} are required.

ⓒ Spline with not-a-knot boundary conditions

Here, the spline's third derivatives need to be continuous at the second ($k = 1$) and penultimate ($k = n - 1$) knots

$$S_0'''(x_1) = S_1'''(x_1) \tag{63}$$

$$S_{n-2}'''(x_{n-1}) = S_{n-1}'''(x_{n-1}) \tag{64}$$

Differentiate $S_k(x)$ given in (46) three times we get

$$S_k'''(x) = 6d_k, \qquad k = 0, \dots, n-1$$

which translate (63)-(64) to the conditions

$$d_0 = d_1, \qquad d_{n-2} = d_{n-1}$$

We shall next use these conditions to substitute for y_0'' and y_n'' in the equation system (56)- (58). For this end, expanding d_k according to its expression from (53) yields

$$\frac{y_1 - y_0}{6h_0} = \frac{y_2 - y_1}{6h_1} \qquad \Longrightarrow \qquad y_0'' = y_1'' - \frac{h_0}{h_1}\left(y_2'' - y_1''\right) \tag{65}$$

and

$$\frac{y_{n-1} - y_{n-2}}{6h_{n-2}} = \frac{y_n - y_{n-1}}{6h_{n-1}} \qquad \Longrightarrow \qquad y_n'' = y_{n-1}'' + \frac{h_{n-1}}{h_{n-2}}\left(y_{n-1}'' - y_{n-2}''\right) \tag{66}$$

Using the expression (65), the equation of the system (53) yields

$$\left(3h_0 + 2h_1 + \frac{h_0^2}{h_1}\right)y_1'' + \left(h_1 - \frac{h_0^2}{h_1}\right)y_2'' = 6\left(\frac{y_2 - y_1}{h_1} - \frac{y_1 - y_0}{h_0}\right) \tag{67}$$

With (66), the last equation of the system (53) become

$$\left(h_{n-2} - \frac{h_{n-1}^2}{h_{n-2}}\right)y_{n-2}'' + \left(2h_{n-2} + 3h_{n-1} + \frac{h_{n-1}^2}{h_{n-2}}\right)y_{n-1}'' = 6\left(\frac{y_n - y_{n-1}}{h_{n-1}} - \frac{y_{n-1} - y_{n-2}}{h_{n-2}}\right) \tag{68}$$

With (67)and (68), the matrix and vectors for a cubic spline with not-a-knot boundary conditions can be constructed as $A_{nak}y = b_{nak}$ where

$$A_{nak} = \begin{bmatrix} \left(3h_0 + 2h_1 + \frac{h_0^2}{h_1}\right) & h_1 - \frac{h_0^2}{h_1} & 0 & \cdots & \cdots & 0 \\ h_1 & 2(h_1 + h_2) & h_2 & & & \vdots \\ 0 & h_2 & 2(h_2 + h_3) & h_3 & & 0 \\ \vdots & & \ddots & \ddots & \ddots & \\ 0 & \cdots & 0 & h_{n-3} & 2(h_{n-3} + h_{n-2}) & h_{n-2} \\ 0 & \cdots & \cdots & 0 & h_{n-2} - \frac{h_{n-1}^2}{h_{n-2}} & \left(2h_{n-2} + 3h_{n-1} + \frac{h_{n-1}^2}{h_{n-2}}\right) \end{bmatrix}$$

$$y = \begin{bmatrix} y_1'' \\ y_2'' \\ y_3'' \\ \vdots \\ y_{n-2}'' \\ y_{n-1}'' \end{bmatrix}, \qquad b_{nak} = \begin{bmatrix} 6\left(\frac{y_2 - y_1}{h_1} - \frac{y_1 - y_0}{h_0}\right) \\ 6\left(\frac{y_3 - y_2}{h_2} - \frac{y_2 - y_1}{h_1}\right) \\ \vdots \\ 6\left(\frac{y_{n-1} - y_{n-2}}{h_{n-2}} - \frac{y_{n-2} - y_{n-3}}{h_{n-3}}\right) \\ 6\left(\frac{y_n - y_{n-1}}{h_{n-1}} - \frac{y_{n-1} - y_{n-2}}{h_{n-2}}\right) \end{bmatrix}$$

When using the matrix inversion the solution reads

$$\begin{aligned} y_0'' &= y_1'' - \frac{h_0}{h_1}\left(y_2'' - y_1''\right) \\ y = \begin{bmatrix} y_1'' \\ y_2'' \\ y_3'' \\ \vdots \\ y_{n-2}'' \\ y_{n-1}'' \end{bmatrix} &= A_{nak}^{-1}.b_{nak} \\ y_n'' &= y_{n-1}'' + \frac{h_{n-1}}{h_{n-2}}\left(y_{n-1}'' - y_{n-2}''\right) \end{aligned}$$

Note that y_0'' and y_n'' can be calculated only after the linear system has been solved because y_1'', y_2'', y_{n-2}'' and y_{n-1}'' are required.

ⓓ Spline with periodic boundary conditions

This case assumes $y_n = y_0$ and demands the periodic boundary conditions

$$
\begin{aligned}
S_0'(x_0) &= S_{n-1}'(x_n) &\implies\quad & b_0 = b_{n-1} + 2c_{n-1}h_{n-1} + 3d_{n-1}h_{n-1}^2 \\
S_0''(x_0) &= S_{n-1}''(x_n) &\implies\quad & y_n'' = y_0''.
\end{aligned} \tag{69}
$$

Using the expressions for c_k, d_k and b_k form (53), the latter equation (69) yields

$$
\frac{y_1 - y_0}{h_0} - \frac{2y_0'' + y_1''}{6}h_0 = \frac{y_n - y_{n-1}}{h_{n-1}} - \frac{2y_{n-1}'' + y_n''}{6}h_{n-1} + y_{n-1}''h_{n-1} + \frac{3\left(y_n'' - y_{n-1}''\right)}{6h_{n-1}}h_{n-1}^2
$$

After simplifying the equation taking into account $y_n'' = y_0''$, we obtain

$$
2\left(h_0 + h_{n-1}\right)y_0'' + h_0 y_1'' + h_{n-1}y_{n-1}'' = 6\left(\frac{y_1 - y_0}{h_0} - \frac{y_n - y_{n-1}}{h_{n-1}}\right)
$$

Now, the system (57)- (59) will transformed to the following

$$
\begin{aligned}
&2\left(h_0 + h_{n-1}\right)y_0'' + h_0 y_1'' + h_{n-1}y_{n-1}'' = 6\left(\frac{y_1 - y_0}{h_0} - \frac{y_n - y_{n-1}}{h_{n-1}}\right) \\
&h_{k-1}y_{k-1}'' + 2\left(h_{k-1} + h_k\right)y_k'' + h_k y_{k+1}'' = 6\left(\frac{y_{k+1} - y_k}{h_k} - \frac{y_k - y_{k-1}}{h_{k-1}}\right), \qquad k = 1, \dots, n-2, \\
&h_{n-1}y_0'' + h_{n-2}y_{n-2}'' + 2\left(h_{n-2} + h_{n-1}\right)y_{n-1}'' = 6\left(\frac{y_n - y_{n-1}}{h_{n-1}} - \frac{y_{n-1} - y_{n-2}}{h_{n-2}}\right),
\end{aligned}
$$

In matrix representation, the system reads $A_{per}y = b_{per}$ with

$$
A_{per} = \begin{bmatrix}
2\left(h_0 + h_{n-1}\right) & h_0 & 0 & \cdots & 0 & h_{n-1} \\
h_0 & 2(h_0 + h_1) & h_1 & 0 & \cdots & 0 \\
0 & h_1 & 2(h_1 + h_2) & h_2 & \ddots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & \cdots & 0 & h_{n-3} & 2\left(h_{n-3} + h_{n-2}\right) & h_{n-2} \\
h_{n-1} & 0 & \cdots & 0 & h_{n-2} & 2\left(h_{n-2} + h_{n-1}\right)
\end{bmatrix}
$$

$$y = \begin{bmatrix} y''_0 \\ y''_1 \\ y''_2 \\ \vdots \\ y''_{n-2} \\ y''_{n-1} \end{bmatrix}, \qquad b_{per} = \begin{bmatrix} 6\left(\frac{y_1-y_0}{h_0} - \frac{y_n-y_{n-1}}{h_{n-1}}\right) \\ 6\left(\frac{y_2-y_1}{h_1} - \frac{y_1-y_0}{h_0}\right) \\ 6\left(\frac{y_3-y_2}{h_2} - \frac{y_2-y_1}{h_1}\right) \\ \vdots \\ 6\left(\frac{y_n-y_{n-1}}{h_{n-1}} - \frac{y_{n-1}-y_{n-2}}{h_{n-2}}\right) \end{bmatrix}$$

Note that A is an $n \times n$ matrix because of the added periodicity condition. It is not a tridiagonal matrix any more but strict diagonal dominance is preserved as

$$|2\left(h_0 + h_{n-1}\right)| > |h_0| + |h_{n-1}|$$
$$|2\left(h_{n-2} + h_{n-1}\right)| > |h_{n-2}| + |h_{n-1}|$$

Using matrix inversion the solution reads

$$\begin{aligned} y = \begin{bmatrix} y''_0 \\ y''_1 \\ y''_2 \\ \vdots \\ y''_{n-2} \\ y''_{n-1} \end{bmatrix} &= A_{per}^{-1}.b_{per} \\ y''_n &= y''_0 \end{aligned}$$

Note that y''_n can be calculated only after the linear system has been solved because the solution value for y''_0 is required.

Sect.8.2 Second Technique

Instead of determining the solution of the problem through a system of $4n$ equations in $4n$ unknowns, we note that $S'(x)$ and $S''(x)$ are quadratic and linear splines, respectively, where the unknowns $S'(x_i) = z_i$ and $S''(x_i) = w_i$, $0 \leqslant i \leqslant n$, represent the sets of slots and moments at the nodes, respectively. We proceed by writing $S''(x_i)$ on $[x_i, x_{i+1}]$ followed by two successive integrations.

$$S''(x_i) = w_i + \left(\frac{w_{i+1} - w_i}{h_i}\right)(t - x_i), \quad \forall\, t \in [x_i, x_{i+1}], \quad \forall\, i = 0, \dots, n-1.$$

Integration of the latter equation from x_i to x: $x_i \leqslant x \leqslant x_{i+1}$, yields

$$S_i'(x) - z_i = w_i\,(x - x_i) + \Big(\frac{w_{i+1} - w_i}{h_i}\Big)\frac{(x - x_i)^2}{2}, \quad \forall\, x \in [x_i, x_{i+1}], \quad \forall\, i = 0, \dots, n-1. \tag{70}$$

Imposing next the condition $S_i'(x_{i+1}) = S_{i+1}'(x_{i+1}) = z_{i+1}$, provide a new set of $n-1$ equations. Specifically,

$$z_{i+1} = z_i + h_i\frac{w_{i+1} + w_i}{2}, \quad i = 0, \dots, n-1. \tag{71}$$

A second integration of equation (70) from x_i to x yields the cubic polynomials $S_i(x)$

$$S_i(x) = y_i + z_i(x - x_i) + w_i\frac{(x - x_i)^2}{2} + \frac{w_{i+1} - w_i}{6h_i}\,(x - x_i)^3\,, \;\forall\, x \in [x_i, x_{i+1}],\; 0 \leqslant i \leqslant n-1 \tag{72}$$

Imposing then the interpolation conditions $S_i(x_{i+1}) = y_{i+1}$ provides a new set of $n-1$ equations given by:

$$y_{i+1} = y_i + z_ih_i + w_i\frac{h_i^2}{2} + \frac{\big(w_{i+1} - w_i\big)h_i^2}{6}, \quad \forall\, 0 \leqslant i \leqslant n-1. \tag{73}$$

This last equation leads to 2 simultaneous equations satisfied at all internal nodes of the spline, i.e. for all $i = 1, \dots n-1$

$$\begin{cases} \dfrac{y_{i+1} - y_i}{h_i} &= z_i + \dfrac{h_i}{6}\big(w_{i+1} + 2w_i\big), \\[2ex] \dfrac{y_i - y_{i-1}}{h_{i-1}} &= z_{i-1} + \dfrac{h_{i-1}}{6}\big(w_i + 2w_{i-1}\big) \end{cases} \tag{74}$$

Subtracting these last two equations and using (70) gives:

$$f[x_i, x_{i+1}] - f[x_{i-1}, x_i] = h_{i-1}\frac{w_{i-1} + w_i}{2} + h_{i+1}\frac{w_i + 2w_{i-1}}{2} - h_{i-1}\frac{w_i + 2w_{i-1}}{6}$$

equivalently,

$$\frac{h_{i-1}}{6}w_{i-1} + \frac{h_{i-1} + h_i}{3}w_i + \frac{h_i}{6}w_{i+1} = f[x_{i-1}, x_i, x_{i+1}]\big(h_{i-1} + h_i\big), \quad 1 \leqslant i \leqslant n-1 \tag{75}$$

Equation (75) provides therefore a system of $n-1$ equations in $n+1$ unknowns given by: $AW = R$ where the coefficient matrix A of size $(n-1)\times(n-1)$ is given by

$$\begin{bmatrix} 2(h_0 + h_1) & h_1 & 0 & \cdots & 0 \\ h_1 & 2(h_1 + h_2) & h_2 & \ddots & \vdots \\ 0 & h_2 & 2(h_2 + h_3) & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & h_{n-2} \\ 0 & \cdots & 0 & h_{n-2} & 2(h_{n-2} + h_{n-1}) \end{bmatrix}$$

and the vectors W and R are respectively:

$$W = (w_1, \dots, w_{n-1})^t, \quad R = (r_1, \dots, r_{n-1})^t, \quad r_i = 6f[x_{i-1}, x_i, x_{i+1}]\big(h_{i-1} + h_i\big),$$

while r_0 and r_n provide two extra free parameters w_0 and w_n (as in the first technique Sect.8.1).

Sect.8.3 Minimal Property of Natural Cubic Splines

Next we provide complete convergence theorems for all four boundary conditions

Theorem 9: Minimal Curvature Property

Let $f \in C^2[a,b]$ and let s be the natural cubic spline interpolating f at nodes $a = x_0 < x_1 < \cdots < x_n = b$, with $s''(a) = s''(b) = 0$. Then for any $g \in C^2[a,b]$ that interpolates f at the same nodes we have the minimal property

$$\int_a^b |s''(x)|^2 dx \leq \int_a^b |g''(x)|^2 dx$$

with equality if and only if $g \equiv s$.

Proof. The proof proceeds in several steps:

Step 1: Variational Approach

Let g be any C^2 interpolant and define $e(x) = g(x) - s(x)$. Note that:

- $e(x_i) = 0$ for all $i = 0, \dots, n$ (since both g and s interpolate f)
- $e''(a) = g''(a) - s''(a) = g''(a)$
- $e''(b) = g''(b) - s''(b) = g''(b)$

Step 2: Key Integral Identity

Consider the integral:

$$\int_a^b s''(x)e''(x)dx = \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} s''(x)e''(x)dx$$

Integrating by parts on each subinterval:

$$\int_{x_i}^{x_{i+1}} s''(x)e''(x)dx = [s''(x)e'(x)]_{x_i}^{x_{i+1}} - \int_{x_i}^{x_{i+1}} s'''(x)e'(x)dx$$

Since s''' is constant on each $[x_i, x_{i+1}]$ (because s is cubic), we can write:

$$-\int_{x_i}^{x_{i+1}} s'''(x)e'(x)dx = -s'''|_{[x_i,x_{i+1}]}\int_{x_i}^{x_{i+1}} e'(x)dx = -s'''|_{[x_i,x_{i+1}]}[e(x_{i+1}) - e(x_i)] = 0$$

Thus the integral simplifies to:

$$\int_a^b s''(x)e''(x)dx = [s''(b)e'(b) - s''(a)e'(a)] = 0 - 0 = 0$$

because $s''(a) = s''(b) = 0$ (natural boundary conditions).

Step 3: Energy Decomposition

Now examine $J(g)$:

$$\begin{aligned}\int_a^b |g''(x)|^2 dx &= \int_a^b |s''(x) + e''(x)|^2 dx \\ &= \int_a^b |s''(x)|^2 dx + 2\int_a^b s''(x)e''(x)dx + \int_a^b |e''(x)|^2 dx \\ &= \int_a^b |s''(x)|^2 dx + \int_a^b |e''(x)|^2 dx \quad \text{(by Step 2)}\end{aligned}$$

Step 4: Conclusion

Since $\int_a^b |e''(x)|^2 dx \geq 0$, we have:

$$\int_a^b |g''(x)|^2 dx \geq \int_a^b |s''(x)|^2 dx$$

with equality if and only if $e''(x) \equiv 0$, which implies $e(x)$ is linear. But since $e(x_i) = 0$ at all nodes, this forces $e(x) \equiv 0$, i.e., $g \equiv s$.

Step 5: Uniqueness

If another function h achieved the same minimal value, then:

$$\int_a^b |h''(x)|^2 dx = \int_a^b |s''(x)|^2 dx$$

and by the above argument, $h \equiv s$. □

Corollary

The natural cubic spline is the unique interpolant that minimizes the strain energy $\int_a^b |g''(x)|^2 dx$ among all C^2 interpolants.

Example 9

Construct a natural cubic spline interpolating the set of data $(-1,2)$, $(1,3)$, $(2,-1)$ and $(2.5,0)$.

Solution

We have $n = 3$. For natural spline we have $w_0 = w_n = 0$, and $h_0 = 2$, $h_1 = 1$, $h_2 = 0.5$. The system $AW = R$ is

$$\begin{bmatrix} (h_0+h_1)/3 & h_1/6 \\ h_1/6 & (h_1+h_2)/3 \end{bmatrix}\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \quad \Longleftrightarrow \quad \begin{bmatrix} 1 & 1/6 \\ 1/6 & 1/2 \end{bmatrix}\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} -7 \\ 6 \end{bmatrix}$$

We get

$$W = \left(w_0 = 0,\ w_1 = -162/17,\ w_2 = 258/17,\ w_3 = 0\right)^t$$

Using next (73) with $i = 0$ leads to

$$z_0 = \frac{y_1 - y_0}{h_0} - \frac{w_1 + 2w_0}{6} h_0 = 125/34$$

Now the value of z_0 allows to compute the vector (71)

$$Z = \left(z_0 = 125/34,\ z_1 = -199/34,\ z_2 = -103/34,\ z_3 = 26/34\right)^t$$

Finally, using the expression (72) for successive $i = 0,1,2$ we obtain the natural cubic spline

$$S(x) = \begin{cases} S_0(x) &= 2 + \frac{125}{34}(x+1) - \frac{27}{17}(x+1)^3, & -1 \leqslant x \leqslant 1 \\ S_1(x) &= 3 - \frac{199}{34}(x-1) - \frac{81}{17}(x-1)^2 + \frac{70}{17}(x-1)^3, & 1 \leqslant x \leqslant 2 \\ S_2(x) &= -1 - \frac{103}{34}(x-2) - \frac{129}{17}(x-2)^2 - \frac{258}{17}(x-2)^3, & 2 \leqslant x \leqslant 2.5 \end{cases}$$

Exercise 13

Show that the clamped cubic spline interpolant is unique for a given dataset and derivative boundary conditions.

Solution

Let $S_1(x)$ and $S_2(x)$ be two clamped cubic splines interpolating the same data with $S_1'(a) = S_2'(a)$ and $S_1'(b) = S_2'(b)$. Define $D(x) = S_1(x) - S_2(x)$. Then:

$$D(x_i) = 0, \quad D'(a) = D'(b) = 0.$$

Since $D(x)$ is a cubic spline with zero derivatives at boundaries, $D''(x)$ is linear. The uniqueness follows from $D''(x) = 0 \Rightarrow D(x) \equiv 0$.

Exercise 14

Explain why clamped boundary conditions $S'(a) = f'(a)$ and $S'(b) = f'(b)$ generally produce more accurate interpolants than natural conditions when the derivative information is known.

Solution

Clamped conditions enforce derivative matching at endpoints, reducing oscillations by constraining the spline's behavior. Natural conditions $S''(a) = S''(b) = 0$ assume zero curvature, which may not reflect the true function's behavior. For example, if $f'(a) \neq 0$, natural splines introduce artificial flatness, increasing approximation error.

Exercise 15

Construct a cubic spline for the data

x	0	2	5
y	1	4	1

with $S'(0) = -1$ (clamped) and $S''(5) = 0$ (natural).

Solution

Define moments $w_i = S''(x_i)$. For intervals $h_1 = 2$, $h_2 = 3$:

$$\begin{cases} 2w_0 + w_1 = \frac{6}{h_1}\left(\frac{y_1 - y_0}{h_1} - S'(0)\right) = \frac{6}{2}(1.5 - (-1)) = 7.5 \\ \frac{h_1}{6}w_0 + \frac{h_1+h_2}{3}w_1 + \frac{h_2}{6}w_2 = \Delta y_2 - \Delta y_1 = -1 - 1.5 = -2.5 \\ w_2 = 0 \quad \text{(natural condition)} \end{cases} \implies \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 3.75 \\ -0.75 \\ 0 \end{bmatrix}$$

The cubic splines polynomials are

$$S_1(x) = 1 + (-1)(x-0) + \frac{3.75}{2}(x-0)^2 - \frac{0.75}{2}(x-0)^3 \quad x \in [0,2]$$

$$S_2(x) = 4 + (0.5)(x-2) + \frac{-0.75}{2}(x-2)^2 + \frac{0}{2}(x-2)^3 \quad x \in [2,5]$$

Exercise 16

Construct a periodic cubic spline for the polar data

θ	0	$\pi/2$	π	$3\pi/2$
r	1	2	1	2

with $S(0) = S(2\pi)$, $S'(0^-) = S'(2\pi^+)$.

Solution

Step 1: Define intervals and moments. Given the periodic nature, we extend the data to include $\theta_4 = 2\pi \equiv 0$, so $r_4 = r_0 = 1$. The intervals have equal width $h = \pi/2$. Let $M_i = S''(\theta_i)$, with $M_0 = M_4$.

Step 2: Set up and solve the system of equations. For periodic cubic splines, the moments satisfy:

$$\frac{h}{6}M_{i-1} + \frac{2h}{3}M_i + \frac{h}{6}M_{i+1} = \frac{r_{i+1} - r_i}{h} - \frac{r_i - r_{i-1}}{h}.$$

Substituting $h = \pi/2$, the right-hand side becomes:

$$d_i = \frac{6}{\pi^2}\left(r_{i+1} - 2r_i + r_{i-1}\right).$$

For $i = 1, 2, 3, 4$:

$$d_1 = \frac{6}{\pi^2}(1 - 2(2) + 1) = -\frac{24}{\pi^2}, \qquad d_2 = \frac{6}{\pi^2}(2 - 2(1) + 2) = \frac{12}{\pi^2},$$

$$d_3 = \frac{6}{\pi^2}(1 - 2(2) + 1) = -\frac{24}{\pi^2}, \qquad d_4 = \frac{6}{\pi^2}(2 - 2(1) + 2) = \frac{12}{\pi^2}.$$

Accounting for periodicity ($M_0 = M_4$), the system becomes:

$$\begin{bmatrix} 4 & 1 & 0 & 1 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 1 & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \end{bmatrix} = \begin{bmatrix} -24/\pi^2 \\ 12/\pi^2 \\ -24/\pi^2 \\ 12/\pi^2 \end{bmatrix} \implies \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \end{bmatrix} = \begin{bmatrix} -12/\pi^2 \\ 6/\pi^2 \\ -12/\pi^2 \\ 6/\pi^2 \end{bmatrix}$$

Step 3: Construct the spline segments.

1. For $\theta \in [0, \pi/2]$: For $\theta \in [0, \pi/2]$:

$$S_1(\theta) = \frac{M_0}{6}\left(\frac{\pi}{2} - \theta\right)^3 + \frac{M_1}{6}\theta^3 + \left(1 - \frac{M_0 h^2}{6}\right)\left(\frac{\pi/2 - \theta}{h}\right) + \left(2 - \frac{M_1 h^2}{6}\right)\left(\frac{\theta}{h}\right).$$

Substituting $M_0 = M_4 = 6/\pi^2$, $M_1 = -12/\pi^2$, and $h = \pi/2$ give

$$S_1(\theta) = \frac{1}{\pi^2}\left(\frac{\pi}{2} - \theta\right)^3 - \frac{2}{\pi^2}\theta^3 + \frac{3}{4}\left(1 - \frac{2\theta}{\pi}\right) + \frac{5}{4}\left(\frac{2\theta}{\pi}\right).$$

2. For $\theta \in [\pi/2, \pi]$:

$$S_2(\theta) = -\frac{2}{\pi^2}(\pi - \theta)^3 + \frac{1}{\pi^2}\left(\theta - \frac{\pi}{2}\right)^3 + \frac{5}{2}\left(1 - \frac{2\theta}{\pi}\right) + \frac{3}{2}\left(\frac{2\theta}{\pi} - 1\right).$$

3. For $\theta \in [\pi, 3\pi/2]$:

$$S_3(\theta) = \frac{1}{\pi^2}\left(\frac{3\pi}{2} - \theta\right)^3 - \frac{2}{\pi^2}(\theta - \pi)^3 + \frac{3}{2}\left(3 - \frac{2\theta}{\pi}\right) + \frac{5}{2}\left(\frac{2\theta}{\pi} - 2\right).$$

4. For $\theta \in [3\pi/2, 2\pi]$:

$$S_4(\theta) = -\frac{2}{\pi^2}(2\pi - \theta)^3 + \frac{1}{\pi^2}\left(\theta - \frac{3\pi}{2}\right)^3 + \frac{5}{2}\left(4 - \frac{2\theta}{\pi}\right) + \frac{3}{2}\left(\frac{2\theta}{\pi} - 3\right).$$

Exercise 17

Given the data points:

x	0	1	2	3
y	1	0	1	0

Construct a cubic spline $S(x)$ with not-a-knot boundary conditions. Write the piecewise polynomial functions and verify continuity of third derivatives at $x = 1$ and $x = 2$

Solution

Step 1: Define Parameters

Given $n = 3$ intervals with knots at $x_0 = 0, x_1 = 1, x_2 = 2, x_3 = 3$.
Interval widths:

$$h_1 = 1 - 0 = 1, \quad h_2 = 2 - 1 = 1, \quad h_3 = 3 - 2 = 1$$

Divided differences:

$$\Delta y_1 = \frac{0-1}{1} = -1, \quad \Delta y_2 = \frac{1-0}{1} = 1, \quad \Delta y_3 = \frac{0-1}{1} = -1$$

Step 2: Set Up System of Equations

Standard Continuity Equations: For interior knots x_1 and x_2:

$$\frac{h_1}{6}M_0 + \frac{h_1+h_2}{3}M_1 + \frac{h_2}{6}M_2 = \Delta y_2 - \Delta y_1 = 2$$
$$\frac{h_2}{6}M_1 + \frac{h_2+h_3}{3}M_2 + \frac{h_3}{6}M_3 = \Delta y_3 - \Delta y_2 = -2$$

Not-a-Knot Conditions: Continuity of third derivative at x_1 and x_2:

$$M_0 - 2M_1 + M_2 = 0 \quad (\text{at } x = 1)$$

$$M_1 - 2M_2 + M_3 = 0 \quad (\text{at } x = 2)$$

Full system:

$$\begin{cases} \frac{1}{6}M_0 + \frac{2}{3}M_1 + \frac{1}{6}M_2 = 2 \\ \frac{1}{6}M_1 + \frac{2}{3}M_2 + \frac{1}{6}M_3 = -2 \\ M_0 - 2M_1 + M_2 = 0 \\ M_1 - 2M_2 + M_3 = 0 \end{cases}$$

Step 3: Solve the System

Matrix form:

$$\begin{bmatrix} \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 \\ 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ M_3 \end{bmatrix} = \begin{bmatrix} 2 \\ -2 \\ 0 \\ 0 \end{bmatrix} \qquad \Longrightarrow \qquad \begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ M_3 \end{bmatrix} = \begin{bmatrix} 6 \\ -6 \\ 6 \\ -6 \end{bmatrix}$$

Step 4: Construct Piecewise Polynomials

For $x \in [0, 1]$:

$$\begin{aligned} S_0(x) &= \frac{M_0}{6}(1-x)^3 + \frac{M_1}{6}x^3 + \left(1 - \frac{M_0}{6}\right)(1-x) + \left(0 - \frac{M_1}{6}\right)x \\ &= (1-x)^3 - x^3 + (1-1)(1-x) + (0+1)x \\ &= (1-x)^3 - x^3 + x \end{aligned}$$

For $x \in [1, 2]$:

$$\begin{aligned} S_1(x) &= \frac{M_1}{6}(2-x)^3 + \frac{M_2}{6}(x-1)^3 + (0+1)(2-x) + (1-1)(x-1) \\ &= -(2-x)^3 + (x-1)^3 + (2-x) \end{aligned}$$

For $x \in [2, 3]$:

$$\begin{aligned} S_2(x) &= \frac{M_2}{6}(3-x)^3 + \frac{M_3}{6}(x-2)^3 + (1-1)(3-x) + (0+1)(x-2) \\ &= (3-x)^3 - (x-2)^3 + (x-2) \end{aligned}$$

Step 5: Verification

Third Derivative Continuity:
For cubic polynomials, the third derivative is constant in each interval:

$$S_0'''(x) = -6, \quad S_1'''(x) = 12, \quad S_2'''(x) = -12$$

However, the not-a-knot condition requires continuity at $x = 1$ and $x = 2$, which is achieved by having the same cubic polynomial on overlapping intervals. Our solution shows $S_0(x)$ and $S_1(x)$ merge into a single cubic on $[0, 2]$, and $S_1(x)$ and $S_2(x)$ merge on $[1, 3]$, satisfying the not-a-knot condition.
Interpolation Check:

$$S_0(0) = 1, \; S_0(1) = 0; \quad S_1(1) = 0, \; S_1(2) = 1; \quad S_2(2) = 1, \; S_2(3) = 0$$

All data points are correctly interpolated.

Exercise 18

Determine a, b, c, and d so that the following function is a natural cubic spline.

$$f(x) = \begin{cases} 3x^3, & 0 \leqslant x \leqslant 2 \\ a(x-2)^3 + b(x-2)^2 + c(x-2) + d, & 2 \leqslant x \leqslant 3 \end{cases}$$

Exercise 19

Find the values of a, b, c, d, e, and f such that the following function defines a cubic spline and find $S(0)$, $S(2)$, and $S(3)$.

$$S(x) = \begin{cases} 2x^3 + 4x^2 - 7x + 5, & 0 \leqslant x \leqslant 1 \\ 3(x-1)^3 + a(x-1)^2 + b(x-1) + c, & 1 \leqslant x \leqslant 2 \\ (x-2)^3 + d(x-2)^2 + e(x-2) + f, & 2 \leqslant x \leqslant 3 \end{cases}$$

Sect.9 Convergence Theorems for Cubic Spline Interpolation

The error theory for spline approximations is somewhat more involved than that for ordinary piecewise polynomial interpolation.
Given a function $f \in C^4[a, b]$ and nodes $a = x_0 < x_1 < \cdots < x_n = b$, a cubic spline interpolant s satisfies:

(i) $s(x_i) = f(x_i)$ for all $i = 0, \dots, n$

(ii) $s \in C^2[a, b]$

(iii) s is a cubic polynomial on each $[x_i, x_{i+1}]$

Define $h = \max_i(x_{i+1} - x_i)$, $h_{\min} = \min_i(x_{i+1} - x_i)$. We assume quasi-uniformity: $h \leq c h_{\min}$ for some $c > 0$.

ⓐ Natural Spline Interpolation

Theorem 1 (Natural Spline Convergence).
For the natural cubic spline with $s''(x_0) = s''(x_n) = 0$*:*

$$\|f - s\|_\infty \leq \frac{5}{384} h^4 \|f^{(4)}\|_\infty$$
$$\|f' - s'\|_\infty \leq \frac{1}{24} h^3 \|f^{(4)}\|_\infty$$
$$\|f'' - s''\|_\infty \leq \frac{3}{8} h^2 \|f^{(4)}\|_\infty$$

Proof. Part 1: Minimal Property The natural spline minimizes $J(g) = \int_a^b |g''(x)|^2 dx$ over all C^2 interpolants. Thus:

$$\int_a^b |s''(x)|^2 dx \leq \int_a^b |f''(x)|^2 dx$$

Part 2: Error Representation For any $x \in [x_i, x_{i+1}]$, using Taylor expansion about x_i:

$$f(x) = f(x_i) + f'(x_i)(x - x_i) + \frac{f''(x_i)}{2}(x - x_i)^2 + \frac{f'''(x_i)}{6}(x - x_i)^3 + \frac{f^{(4)}(\xi_i)}{24}(x - x_i)^4$$

The spline satisfies:

$$s(x) = s(x_i) + s'(x_i)(x - x_i) + \frac{s''(x_i)}{2}(x - x_i)^2 + \frac{s'''(x_i)}{6}(x - x_i)^3$$

Part 3: Error Bounds Subtracting and using $f(x_i) = s(x_i)$:

$$|f(x) - s(x)| \leq \frac{h^4}{24} \|f^{(4)}\|_\infty + \text{derivative error terms}$$

The natural boundary conditions improve this to $\frac{5}{384} h^4 \|f^{(4)}\|_\infty$.
For derivatives, differentiate the error expansion:

$$|f'(x) - s'(x)| \leq \frac{h^3}{6} \|f^{(4)}\|_\infty$$

$$|f''(x) - s''(x)| \leq \frac{h^2}{2} \|f^{(4)}\|_\infty$$

The natural conditions tighten these to the stated bounds. □

ⓑ Clamped Spline Interpolation

Theorem 2 (Clamped Spline Convergence).
For the clamped spline with $s'(x_0) = f'(x_0)$, $s'(x_n) = f'(x_n)$:

$$\|f - s\|_\infty \leq \frac{h^4}{384}\|f^{(4)}\|_\infty$$

$$\|f' - s'\|_\infty \leq \frac{h^3}{24\sqrt{5}}\|f^{(4)}\|_\infty$$

$$\|f'' - s''\|_\infty \leq \frac{3h^2}{8}\|f^{(4)}\|_\infty$$

Proof. Part 1: Optimality The clamped spline minimizes $\int_a^b |g''(x)|^2 dx$ over all C^1 interpolants with correct endpoint derivatives.

Part 2: Error Function Define $e(x) = f(x) - s(x)$. At endpoints:

$$e(x_0) = e(x_n) = 0, \quad e'(x_0) = e'(x_n) = 0$$

Part 3: Integral Representation Using integration by parts four times:

$$e(x) = \int_a^b K(x,t) f^{(4)}(t) dt$$

where $K(x,t)$ is the Peano kernel:

$$K(x,t) = \begin{cases} \frac{(x-t)^3}{6} - \frac{(x-a)^3}{6(b-a)}(b-t) & a \leq t \leq x \\ -\frac{(x-a)^3}{6(b-a)}(b-t) & x \leq t \leq b \end{cases}$$

Part 4: Kernel Estimation Maximizing $|K(x,t)|$ gives $\frac{h^4}{384}$. For derivatives:

$$e'(x) = \int_a^b \frac{\partial K}{\partial x} f^{(4)}(t) dt$$

$$\left|\frac{\partial K}{\partial x}\right| \leq \frac{h^3}{24\sqrt{5}}$$

□

ⓒ Not-a-Knot Spline Interpolation

Theorem 3 (Not-a-Knot Convergence).
For the not-a-knot spline (continuous s''' at x_1 and x_{n-1}):

$$\|f - s\|_\infty \leq Ch^4\|f^{(4)}\|_\infty$$

where C depends only on the mesh ratio $h/h_{\min}$.

Proof. Part 1: Moment Equations The not-a-knot conditions modify the standard tridiagonal system:

$$\mu_i M_{i-1} + 2M_i + \lambda_i M_{i+1} = d_i$$

where $M_i = s''(x_i)$, with special equations at $i = 1$ and $i = n-1$.
Part 2: Stability The system matrix A satisfies:

$$\|A^{-1}\|_\infty \leq \frac{1}{2 - |\mu_i| - |\lambda_i|} \leq C$$

Part 3: Consistency The right-hand side approximates:

$$|d_i - f''(x_i)| \leq Ch^2 \|f^{(4)}\|_\infty$$

Part 4: Error Bound Combining gives:

$$\|M - f''(\mathrm{x})\|_\infty \leq Ch^2 \|f^{(4)}\|_\infty$$

Integrating twice yields the $O(h^4)$ result. □

d Periodic Spline Interpolation

Theorem 4 (Periodic Spline Convergence).
For periodic f with $s(x_0) = s(x_n)$, $s'(x_0) = s'(x_n)$, $s''(x_0) = s''(x_n)$:

$$\|f - s\|_\infty \leq Ch^4 \|f^{(4)}\|_\infty$$
$$\|f' - s'\|_\infty \leq Ch^3 \|f^{(4)}\|_\infty$$
$$\|f'' - s''\|_\infty \leq Ch^2 \|f^{(4)}\|_\infty$$

Proof. Part 1: Fourier Analysis The periodic spline solution can be expressed as:

$$s(x) = \sum_{k=-n/2}^{n/2} c_k e^{ikx}$$

Part 2: Coefficient Bounds The DFT coefficients satisfy:

$$|\hat{f}_k - \hat{s}_k| \leq C \frac{h^4}{k^4} |\hat{f}_k^{(4)}|$$

Part 3: Physical Space Bounds By the Poisson summation formula:

$$\|f - s\|_2 \leq Ch^4 \|f^{(4)}\|_2$$

Sobolev embedding gives the uniform bounds. □