Approximation by Lagrange Splines

Boris I. Kvasov* and Anirut Luadsong

School of Mathematics, Suranaree University of Technology
University Avenue 111, 30000, Nakhon Ratchasima, Thailand
boris@math.sut.ac.th and anirut@math.sut.ac.th

Abstract. Polynomials and their smooth piecewise analogs known as splines are used as the basic means of approximation in nearly all areas of numerical analysis. For this reason, the representation and evaluation of polynomials and splines is a fundamental topic in numerical analysis. We will discuss this topic in the context of local spline interpolation, the simplest and certainly the most widely used technique for obtaining spline approximation. One central point of this paper is a generalization of Horner's rule for the simultaneous evaluation of the interpolating polynomial and its derivatives. Such an algorithm is usually not found in standard textbooks on numerical analysis.
We will study the simplest piecewise polynomial approximations known as Lagrange interpolating splines in detail. Using a very simple approach we show how to obtain smooth analogues of Lagrange splines which only approximate the data while still providing the same order of approximation as Lagrange interpolating splines. Such functions are usually called quasi-interpolants. We then study the commonly used Lagrange splines, such as piecewise cubic and piecewise quadratic Lagrange polynomials in detail. Relations between discrete polynomial splines and Lagrange splines are investigated including a generalization of Marsden's identity [16].

1. Polynomial Interpolation Problem

Let a real-valued function f defined on some interval $[a,b]$ be stored in tabular form (x_i, f_i), $i = 0, \dots, N$, where $f_i = f(x_i)$ and where the points x_i form an ordered sequence $a = x_0 < x_1 < \cdots < x_N = b$.

A typical *interpolation problem* consists of the selection of a function P_N from a given class of functions in a way such that the graph of P_N passes through the given set of data points, that is, $P_N(x_i) = f_i$, $i = 0, \dots, N$, where the points x_i are called the *interpolation nodes*.

The traditional and simplest method for solving the interpolation problem is the construction of an interpolating polynomial P_N. The interpolation conditions

$$P_N(x_i) = \sum_{j=0}^{N} a_j x_i^j = f_i, \quad i = 0, \dots, N \tag{1.1}$$

are equivalent to the system of linear algebraic equations

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^N \\ 1 & x_1 & x_1^2 & \cdots & x_1^N \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & x_N^2 & \cdots & x_N^N \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_N \end{bmatrix}. \tag{1.2}$$

The matrix of system (1.2) is called the Vandermonde matrix and its determinant is the Vandermonde determinant. In our case the Vandermonde determinant $D = \prod_{0\le i<j\le N}(x_j - x_i)$ is nonzero, so that system (1.2) has a unique solution. This proves the existence and uniqueness of an interpolating polynomial of degree $\le N$. However a direct solution of system (1.2) can in general not be recommended as its matrix (with "almost" linearly dependent rows) is often ill-conditioned. The evaluation of the interpolating polynomial can be performed very efficiently by using the Lagrange interpolation formula which permits us to write down the solution of system (1.2) explicitly.

2. Lagrange Interpolation Formula

Let us consider the *Lagrange formula for the interpolating polynomial*

$$L_N(x) = \sum_{j=0}^{N} f_j l_j(x), \tag{1.3}$$

where the *Lagrange coefficient polynomial* l_j with the property

$$l_j(x_i) = \begin{cases} 1 & \text{if } i = j, \\ 0, & \text{otherwise} \end{cases}$$

has the explicit form

$$l_j(x) = \frac{(x - x_0)\cdots(x - x_{j-1})(x - x_{j+1})\cdots(x - x_N)}{(x_j - x_0)\cdots(x_j - x_{j-1})(x_j - x_{j+1})\cdots(x_j - x_N)}, \quad j = 0, \dots, N$$

and can be written in short as

$$\begin{aligned} l_j(x) &= \frac{\omega_N(x)}{(x - x_j)\omega_N'(x_j)}, \quad j = 0, \dots, N, \\ \omega_N(x) &= (x - x_0)(x - x_1)\cdots(x - x_N). \end{aligned} \tag{1.4}$$

The graph of the Lagrange coefficient polynomial l_5 with $N = 10$ and nodes $x_i = i$, $i = 0, \dots, 10$ is given in Figure 1.1.

One can easily verify that

$$L_N(x_i) = f_i, \quad i = 0, \dots, N.$$

According to (1.3) and (1.4) the number of arithmetic operations necessary to compute the value of the interpolating polynomial in Lagrange form (or *Lagrange interpolating polynomial* for short) is proportional to N^2.

Lemma 1.1. *The Lagrange interpolating polynomial is exact for polynomials of degree $\le N$, that is, for any polynomial P_k of degree $k \le N$ the following identity is valid,*

$$P_k(x) = \sum_{j=0}^{N} P_k(x_j) l_j(x), \quad 0 \le k \le N.$$

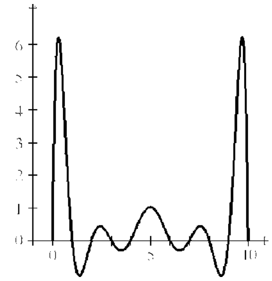

Figure 1.1. The graph of the Lagrange coefficient polynomial l_5 with $N = 10$ and nodes $x_i = i$, $i = 0, \ldots, 10$.

Proof: It is sufficient to verify the validity of the above formula for monomials, that is, to prove the identity

$$x^k \equiv \sum_{j=0}^{N} x_j^k l_j(x), \quad k = 0, \ldots, N.$$

Now the polynomial of the degree N,

$$F_{k,N}(x) = x^k - \sum_{j=0}^{N} x_j^k l_j(x), \quad 0 \le k \le N,$$

has $N+1$ zeros: $F_{k,N}(x_i) = 0$, $i = 0, \ldots, N$. So by the Fundamental Theorem of Algebra, $F_{k,N}$ must be identically equal to zero. This proves the lemma. $\square$

Let us estimate the error in polynomial interpolation. For an integer $k \ge 0$, we denote by $C^k = C^k[a,b]$ the set of functions on $[a,b]$ which have k continuous derivatives.

Theorem 1.1. *Let f be a function in $C^{N+1}[a,b]$, and let L_N be the polynomial of degree $\le N$ which interpolates the function f at $N+1$ distinct points $x_0, x_1, \ldots, x_N$ in the interval $[a,b]$. Then, for each x in $[a,b]$, there exists a number ξ_x in $[a,b]$ such that*

$$f(x) - L_N(x) = \frac{1}{(N+1)!} f^{(N+1)}(\xi_x)\omega_N(x). \tag{1.5}$$

Proof: If x is one of the nodes of interpolation x_i, the assertion is obviously true since both sides of (1.5) reduce to 0. So let x be any point other than a node, and consider the function

$$\Phi(x) = f(x) - L_N(x) - C\omega_N(x),$$

where C is the real number that makes $\Phi(x) = 0$, that is,

$$C = (f(x) - L_N(x))/\omega_N(x).$$

Now $\Phi \in C^{N+1}[a,b]$ and Φ vanishes at $N+2$ points $x, x_0, x_1, \ldots, x_N$. By Rolle's theorem, Φ' has at least $N+1$ distinct ze ros in (a,b). Similarly, Φ'' has at least N distinct zeros in (a,b). Repeating this argument, we conclude eventually that $\Phi^{(N+1)}$ has at least one zero, say ξ_x, in (a,b). Now

$$\begin{aligned}\Phi^{(N+1)}(\xi_x) &= f^{(N+1)}(\xi_x) - C(N+1)! \\ &= f^{(N+1)}(\xi_x) - (N+1)!\frac{f(x) - L_N(x)}{\omega_N(x)} = 0,\end{aligned}$$

and upon solving for f, we have the equality (1.5). This proves the theorem. □

Example 1.1. If the function $f(x) = \sin x$ is approximated by a polynomial of degree 9 which interpolates f at ten points in the interval $[0,1]$, how large is the error on this interval?

Solution. Let us apply theorem 1.1. It is clear that $|f^{(10)}(\xi_x)| \leq 1$ and $|(x - x_0)\cdots(x - x_9)| \leq 1$. Thus, for all x in $[0,1]$, according to (1.5)

$$|\sin x - L_9(x)| \leq \frac{1}{10!} < 2.8 \times 10^{-7}.$$

If one does not need the interpolating polynomial L_N itself but only its value $L_N(x)$ at x, then one can use the *Aitken interpolation scheme.* Let $L_{0,k-1}$ and $L_{1,k-1}$ be the Lagrange interpolating polynomials associated with the data (x_i, f_i) for $i = 0, \ldots, k-1$ and $i = 1, \ldots, k$ correspondingly.

Lemma 1.2. *Let f be defined at $x_0, x_1, \ldots, x_k$ and x_0 and x_k be two distinct numbers in this set. Then the Aitken interpolation formula*

$$L_{0,k}(x) = \frac{x_k - x}{x_k - x_0} L_{0,k-1}(x) + \frac{x - x_0}{x_k - x_0} L_{1,k-1}(x), \qquad k = 1, \ldots, N, \tag{1.6}$$

describes the Lagrange interpolating polynomial of degree $\leq k$ which interpolates f at the $k+1$ points $x_0, x_1, \ldots, x_k$.

Proof: The polynomial on the right side of (1.6) has degree $\leq k$ and interpolates the data (x_i, f_i), $i = 0, \ldots, k$. As the difference of two interpolating polynomials of degree k would have $k+1$ zeros and therefore would be equal identically to zero, such an interpolating polynomial is unique and thus coincides with the Lagrange interpolating polynomial $L_k \equiv L_{0,k}$. This proves the lemma. □

3. Newton Interpolating Polynomial

Let us consider the recurrence relation for Lagrange interpolating polynomials of a different kind

$$L_k(x) = L_{k-1}(x) + c_k(x - x_0)\cdots(x - x_{k-1}), \quad k = 1, \dots, N.$$

By the interpolation condition $L_k(x_k) = f_k$ we have here

$$c_k = \frac{f_k - L_{k-1}(x_k)}{(x_k - x_0)\cdots(x_k - x_{k-1})} = f[x_0, \dots, x_k].$$

This notation is usually called a *divided difference of order* k. In particular, if $k = 0$ then one sets $c_0 = f[x_0] \equiv f_0$. Therefore,

$$L_k(x) = L_{k-1}(x) + f[x_0, \dots, x_k](x - x_0)\cdots(x - x_{k-1}). \tag{1.7}$$

As according to (1.3) and (1.4),

$$L_k(x) = \sum_{j=0}^{k} f_j \frac{\omega_k(x)}{(x - x_j)\omega_k'(x_j)}, \tag{1.8}$$

then by looking at each kth degree term in (1.8) and comparing with the coefficient of x^k in (1.7) we obtain

$$f[x_0, \dots, x_k] = \sum_{j=0}^{k} \frac{f_j}{\omega_k'(x_j)}. \tag{1.9}$$

From this formula, we obtain an important property of the divided difference. Let $(i_0, \dots, i_k)$ be some permutation of $(0, \dots, k)$. Then it is easily seen that

$$f[x_{i_0}, \dots, x_{i_k}] = \sum_{j=0}^{k} \frac{f_{i_j}}{\omega_k'(x_{i_j})} = \sum_{j=0}^{k} \frac{f_j}{\omega_k'(x_j)} = f[x_0, \dots, x_k].$$

Thus, the divided difference is invariant under any permutation of its arguments.

The formula (1.6) can be rewritten in the form

$$L_k(x) = L_{k-1}(x) + \frac{x - x_0}{x_k - x_0}[L_{1,k-1}(x) - L_{0,k-1}(x)]. \tag{1.10}$$

As

$$\begin{aligned} L_{0,k-1}(x) &= f_1 + f[x_1, x_2](x - x_1) + \cdots \\ &\qquad + f[x_1, \dots, x_{k-1}, x_0](x - x_1)\cdots(x - x_{k-1}), \\ L_{1,k-1}(x) &= f_1 + f[x_1, x_2](x - x_1) + \cdots \\ &\qquad + f[x_1, \dots, x_{k-1}, x_k](x - x_1)\cdots(x - x_{k-1}), \end{aligned}$$

using the property of invariance of the divided difference under permutation of its arguments, one obtains

$$L_{1,k-1}(x) - L_{0,k-1}(x) = (f[x_1, \dots, x_k] - f[x_0, \dots, x_{k-1}])(x - x_1)\cdots(x - x_{k-1}).$$

Substituting this expression into (1.10) and comparing with (1.7) one arrives at the recurrence formula

$$f[x_0, \dots, x_k] = \frac{f[x_1, \dots, x_k] - f[x_0, \dots, x_{k-1}]}{x_k - x_0}.$$

Summing up the equalities (1.7) for k ranging from 1 to N, we obtain *Newton's divided difference formula* for the interpolating polynomial or a formula for the *Newton interpolating polynomial*

$$L_N(x) = L_0(x) + \sum_{k=1}^{N} f[x_0, \dots, x_k]\omega_{k-1}(x)$$

which can be rewritten as

$$L_N(x) = c_0 + c_1(x - x_0) + \cdots + c_N(x - x_0)\cdots(x - x_{N-1}), \tag{1.11}$$

where

$$\begin{aligned} c_0 &= f[x_0] \equiv f_0, \\ c_k &= \frac{f[x_1, \dots, x_k] - f[x_0, \dots, x_{k-1}]}{x_k - x_0}, \quad k = 1, \dots, N. \end{aligned} \tag{1.12}$$

Let us consider the Newton polynomial interpolating a function f at points $x_0, \dots, x_N, t$, where $t \neq x_i$, $i = 0, \dots, N$. Then according to (1.11),

$$L_{N+1}(x) = L_N(x) + f[x_0, \dots, x_N, t]\omega_N(x). \tag{1.13}$$

As $L_{N+1}(t) = f(t)$, then by setting $x = t$ in (1.13) we obtain

$$f(t) - L_N(t) = f[x_0, \dots, x_N, t]\omega_N(t). \tag{1.14}$$

Comparing this formula with (1.5) we conclude that

$$f[x_0, \dots, x_N, x] = \frac{1}{(N+1)!} f^{(N+1)}(\xi_x).$$

If we set $x = x_{N+1}$ and $N = n - 1$ then this formula can be rewritten in symmetric form,

$$f[x_0, \dots, x_n] = \frac{f^{(n)}(\xi)}{n!} \tag{1.15}$$

for some $\xi \in [x_0, x_n]$. Let us note that if f is a polynomial of the degree N and of the form (1.1) then

$$f[x_0, \dots, x_n, x] = \begin{cases} \text{polynomial of degree } N - n - 1, & \text{if } n < N - 1, \\ a_N, & \text{if } n = N - 1, \\ 0, & \text{if } n > N - 1. \end{cases}$$

The proof of this equality can easily be obtained by induction. The description of further valuable properties of divided differences can be found in [2,4,5,15].

4. Evaluation of the Newton Interpolating Polynomial and Its Derivatives by a Generalized Horner's Rule

Formally, to find the value of the l-th derivative of the Newton interpolating polynomial $L_N^{(l)}$, $0 \leq l \leq N$ at $x = z$ for some given real number z, one can consider a substitution of the variable x by the formula $x = y + z$ in (1.11), where y is a new variable. After this substitution and after collecting similar terms one obtains

$$L_N(y+z) = A_0 + A_1 y + \cdots + A_N y^N,$$

where $A_l = L_N^{(l)}(z)/l!$, $l = 0, \ldots, N$.

After the reverse substitution $y = x - z$ one finds that

$$L_N(x) = A_0 + A_1(x-z) + \cdots + A_N(x-z)^N. \tag{1.16}$$

We are interested, however, in a more efficient method for evaluating the Newton interpolating polynomial and its derivatives which generalizes the well-known algorithm of *nested multiplication* or *Horner's rule.*

Let us rename $a_{k,0} = c_k$, $k = 0, \ldots, N$ and rewrite the polynomial L_N in the form

$$L_N(x) \equiv P_{N,0}(x) = a_{0,0} + \sum_{k=1}^{N} a_{k,0}\,\omega_{k-1}(x), \quad \omega_{k-1}(x) = \prod_{i=0}^{k-1}(x - x_i). \tag{1.17}$$

By using parentheses we transform the representation (1.17) into the form

$$\begin{aligned} P_{N,0}(x) =& a_{0,0} + (x - x_0)(a_{1,0} + \\ & \cdots + (x - x_{N-2})(a_{N-1,0} + (x - x_{N-1})a_{N,0})\cdots). \end{aligned} \tag{1.18}$$

To evaluate the polynomial $P_{N,0}$ at $x = z$ we form the sequence of numbers

$$\begin{aligned} a_{N,1} &= a_{N,0}, \\ a_{k,1} &= a_{k,0} + (z - x_k)a_{k+1,1}, \quad k = N-1, \ldots, 0, \end{aligned} \tag{1.19}$$

starting with the innermost parentheses in (1.18). It follows from (1.18) and (1.19) that $L_N(z) \equiv P_{N,0}(z) = a_{0,1}$. To find the value of the polynomial $P_{N,0}$ one needs to perform N multiplications and N additions only.

To evaluate the lth $(0 \leq l \leq N)$ derivative of the Newton interpolating polynomial, let us consider the polynomial

$$\begin{aligned} P_{N,l}(x) =& a_{l,l} + \sum_{k=l+1}^{N} a_{k,l}\omega_{k-l-1}(x) \\ =& a_{l,l} + (x - x_0)(a_{l+1,l} + \\ & \cdots + (x - x_{N-l-2})(a_{N-1,l} + (x - x_{N-l-1})a_{N,l})\cdots). \end{aligned} \tag{1.20}$$

Let us set

$$
\begin{aligned}
a_{N,l+1} &= a_{N,l}, \\
a_{k,l+1} &= a_{k,l} + (z - x_{k-l})a_{k+1,l+1}, \quad k = N-1, \dots, l.
\end{aligned} \tag{1.21}
$$

It follows from (1.20) and (1.21) that $P_{N,l}(z) = a_{l,l+1}$ $(0 \le l \le N)$ and to evaluate the polynomial $P_{N,l}$ at $x = z$, one needs to perform $N-l$ multiplications and $N-l$ additions only.

Lemma 1.3. *Let $P_{N,N+1} \equiv 0$. The following equalities are valid*

$$
P_{N,l}(x) = P_{N,l}(z) + (x-z)P_{N,l+1}(x), \quad l = 0, \dots, N. \tag{1.22}
$$

Proof: If $k = N - l = 0$ then equality (1.22) is evident as in this case we have $P_{N,N}(x) = P_{N,N}(z) = a_{N,N}$ by (1.20). Now suppose the equality (1.22) is satisfied for all $k = 0, \dots, N - l'$ $(l < l' \le N)$. We show that it is also fulfilled for $k = N - l$. Using formulae (1.21) one obtains

$$
\begin{aligned}
&P_{N,l}(z) + (x-z)P_{N,l+1}(x) \\
=&a_{l,l+1} + (x - x_0 + x_0 - z)\Big(a_{l+1,l+1} + \sum_{k=l+2}^{N} a_{k,l+1} \prod_{i=0}^{k-l-2} (x - x_i)\Big) \\
=&a_{l,l} + (x - x_0)\Big(a_{l+1,l+1} + (x - x_1 + x_1 - z) \sum_{k=l+2}^{N} a_{k,l+1} \prod_{i=1}^{k-l-2} (x - x_i)\Big) \\
=&a_{l,l} + (x - x_0)(a_{l+1,l} + \cdots + (x - x_{N-l-2})(a_{N-1,l+1} \\
&\qquad + (x - x_{N-l-1} + x_{N-l-1} - z)a_{N,l+1}) \cdots) = P_{N,l}(x).
\end{aligned}
$$

This proves the lemma. □

By repeated differentiation of equality (1.22) and setting $l = 0$ and $x = z$ we obtain

$$
P_{N,0}^{(\alpha)}(z) = P_{N,\alpha}(z), \quad \alpha = 0, \dots, N.
$$

As $L_N \equiv P_{N,0}$, this permits us to rewrite the representation (1.16) in the form

$$
L_N(x) = P_{N,0}(z) + (x-z)P_{N,1}(z) + \cdots + (x-z)^N P_{N,N}(z),
$$

where $P_{N,l}(z) = a_{l,l+1}$, $l = 0, \dots, N$ and $a_{N,N+1} = a_{N,0}$.

Thus, the following result is valid.

Theorem 1.2. *Let L_N be a polynomial of the form (1.11), where one needs to evaluate the derivatives $L_N^{(l)}$, $0 \le l \le N$ at $x = z$.*

Set $a_{k,0} = c_k$, $k = 0, \dots, N$, and $a_{N,l+1} = a_{N,0}$, $l = 0, \dots, N$, and evaluate

$$
a_{k,l+1} = a_{k,l} + (z - x_{k-l})a_{k+1,l+1}; \quad k = N-1, \dots, l, \quad l = 0, \dots, N.
$$

If

$$P_{N,l}(x) = a_{l,l} + \sum_{k=l+1}^{N} a_{k,l}\,\omega_{k-l-1}(x), \quad l = 0, \ldots, N,$$

then

$$L_N^{(l)}(z)/l! = P_{N,l}(z) = a_{l,l+1}, \quad 0 \le l \le N.$$

Let us note that if $x_i = 0$ for all i then the polynomial L_N in (1.11) takes the form $L_N(x) = c_0 + c_1 x + \cdots + c_N x^N$ and the algorithm described above is reduced to the well-known algorithm of nested multiplication, also called Horner's rule, which can be found in many textbooks on numerical analysis, see e.g. [3,5,10,11,13].

The algorithm above can be easily coded. Let us assume that we have two arrays of the data $t[1:n+1]$ and $f[1:n+1]$. First, by using formula (1.12) one computes the divided differences.

```
for i := 1 to n + 1 do a[i] := f[i];
  for i := 1 to n do
    for j := n + 1 downto i + 1 do
      a[j] := (a[j] − a[j − 1])/(t[j] − t[j − i]);
```

The computation of the divided differences can also be performed by a different algorithm [17]:

```
a[n + 1] := f[n + 1];
for i := n downto 1 do
  begin
    a[i] := f[i];
    for j := i + 1 to n + 1 do
      a[j] := (a[j] − a[j − 1])/(t[j] − t[i]);
  end
```

Now in order to evaluate $L_N^{(l)}$, $0 \le l \le N$, at $x = z$ one can use the following loops.

```
for i := 1 to n + 1 do d[i] := a[i];
  k := 1;
    for i := 1 to l + 1 do
      begin
        if i > 1 then k := k ∗ (i − 1);
        if i < n + 1 then
          for j := n downto i do
            d[j] := d[j] + (z − t[j − i + 1]) ∗ d[j + 1]
      end;
    vnewn := k ∗ d[l + 1];
```

Example 1.2. Four values of the function $f(x) = 1/(1+x^2)$ are given in table 1.1. Form a cubic Newton interpolating polynomial L_3 from the data of this table.

Then evaluate $L_3(1.5)$ and $L_3'(1.5)$ using the generalization of Horner's rule. Finally, estimate the error of the approximation thus obtained.

Table 1.1. The initial data

i	x_i	f_i
0	-1	0.5
1	0	1.0
2	1	0.5
3	2	0.2

Solution. By the data of the table 1.1, using formulae (1.12), we first form the table of divided differences.

Table 1.2. Divided differences values.

x_i	f_i	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$f[x_i, \ldots, x_{i+3}]$
-1	0.5			
0	1.0	0.5		
1	0.5	-0.5	-0.5	
2	0.2	-0.3	0.1	0.2

According to (1.11), the cubic Newton interpolating polynomial L_3 takes the form

$$\begin{aligned} L_3(x) =& c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) \\ &+ c_3(x - x_0)(x - x_1)(x - x_2) \\ =& 0.5 + 0.5(x + 1) - 0.5(x + 1)x + 0.2(x + 1)x(x - 1). \end{aligned}$$

By setting $a_{i,0} = c_i$, $i = 0, \ldots, 3$, let us rewrite the polynomial L_3 in the form

$$\begin{aligned} L_3(x) \equiv P_{3,0}(x) &= a_{0,0} + (x - x_0)(a_{1,0} + (x - x_1)(a_{2,0} + (x - x_2)a_{3,0})) \\ &= 0.5 + (x + 1)(0.5 + x(-0.5 + (x - 1)0.2)). \end{aligned}$$

The value of the function $P_{3,0}$ at $z = 1.5$ can be found by Horner's rule

$$\begin{aligned} a_{3,1} &= a_{3,0} = 0.2, \\ a_{2,1} &= a_{2,0} + (z - x_2)a_{3,1} = -0.5 + 0.5 \cdot 0.2 = -0.4, \\ a_{1,1} &= a_{1,0} + (z - x_1)a_{2,1} = 0.5 + 1.5(-0.4) = -0.1, \\ a_{0,1} &= a_{0,0} + (z - x_0)a_{1,1} = 0.5 + 2.5(-0.1) = 0.25. \end{aligned}$$

Thus, $L_3(1.5) \equiv P_{3,0}(1.5) = 0.25$.

In order to evaluate the derivative $L_3'(1.5)$, let us write down the polynomial

$$\begin{aligned} P_{3,1}(x) &= a_{1,1} + (x - x_0)(a_{2,1} + (x - x_1)a_{3,1}) \\ &= -0.1 + (x + 1)(-0.4 + x \cdot 0.2). \end{aligned}$$

Computations are again performed by Horner's rule,

$$\begin{aligned}
a_{3,2} &= a_{3,1} = 0.2, \\
a_{2,2} &= a_{2,1} + (z - x_1)a_{3,2} = -0.4 + 1.5 \cdot 0.2 = -0.1, \\
a_{1,2} &= a_{1,1} + (z - x_0)a_{2,2} = -0.1 + 2.5(-0.1) = -0.35.
\end{aligned}$$

Therefore, $L_3'(1.5) = P_{3,1}(1.5) = -0.35$.

Using the explicit formula for the function f we find a bound for the error of approximation,

$$\begin{aligned}
f(1.5) - L_3(1.5) &= 0.30769 - 0.25 = 0.05769, \\
f'(1.5) - L_3'(1.5) &= -0.28402 + 0.35 = 0.06598.
\end{aligned}$$

5. Convergence of the Interpolating Polynomials

The choice of polynomials as a tool for approximation of functions is usually motivated by the following well-known theorem by Weierstrass.

Theorem 1.3. *Let f be a function which is continuous on the interval $[a, b]$ and let $\varepsilon > 0$. Then there exists a polynomial P_N of degree $N = N(\varepsilon)$ for which*

$$\max_{a \le x \le b} |f(x) - P_N(x)| < \varepsilon.$$

However, we are interested in the interpolation problem and in particular in the convergence of the interpolation process. That is, if f is a continuous function on $[a, b]$ and $P_N(x_i) = f(x_i)$, $i = 0, \ldots, N$, will the quantity $\max_{a \le x \le b} |f(x) - P_N(x)|$ tend to zero as $N \to \infty$?

One can give examples where one does not have convergence. The most famous one was given by Runge in 1901. Let the function $f(x) = 1/(1 + 25x^2)$ be interpolated on the interval $[-1, 1]$ by using equally spaced nodes $x_i = -1 + 2i/N$, $i = 0, \ldots, N$. Then one can show (see [8]) that

$$\lim_{N \to \infty} \max_{0.726\ldots \le |x| < 1} |f(x) - L_N(x)| = \infty.$$

Figure 1.2 illustrates the divergence of the interpolation process for Runge's example. In Figure 1.2, the interpolating polynomial L_{20} deviates substantially from the interpolated function near the ends of the interval $[-1, 1]$. The oscillations tend to infinity with growing of N.

Even more, the following result by Faber (see [10]) is valid.

Theorem 1.4. *For any prescribed system of nodes*

$$a \le x_0^{(N)} < x_1^{(N)} < \cdots < x_N^{(N)} \le b \quad (N \ge 0) \tag{1.23}$$

there exists a continuous function f on $[a, b]$ such that the interpolating polynomials for f using these nodes fail to converge uniformly to f.

However, convergence of the interpolation process can be ensured by a special choice of the interpolation nodes (see [10]).

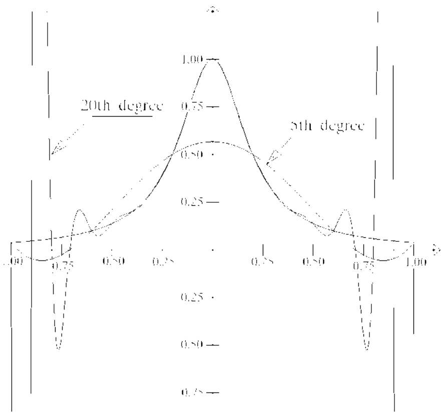

Figure 1.2. Runge's function interpolated by 5th-degree and 20th-degree interpolating polynomials using equidistant data.

Theorem 1.5. *If f is a continuous function on $[a, b]$, then there exists a system of nodes as in equation (1.23) such that the polynomials P_N which interpolate f at these nodes converge to f, that is,*

$$\lim_{N\to\infty} \max_{a\le x\le b} |P_N(x) - f(x)| = 0.$$

In practice one often chooses the roots of the Chebyshev polynomials as a system of interpolation nodes which guarantee the convergence (see [2]). As a rule, the problem of convergence disappears if one turns to interpolation by piecewise Lagrange polynomials which are also called *Lagrange splines* [1].

6. Piecewise Linear Interpolation

The simplest example of Lagrange splines which guarantee the convergence of the interpolation process to the interpolated function is piecewise linear interpolation. In this case, one has a Lagrange interpolating polynomial of first degree on each interval $[x_i, x_{i+1}]$, $i = 0, \ldots, N-1$

$$L_{i,1}(x) = f_i \frac{x_{i+1} - x}{h_i} + f_{i+1} \frac{x - x_i}{h_i}, \quad h_i = x_{i+1} - x_i. \tag{1.24}$$

Thus, on the whole interval $[a, b]$ one has a set of N Lagrange interpolating polynomials of first degree forming a *linear Lagrange spline* or, what is the same, a *Lagrange spline of the first degree.*

Setting $x_{-1} < a$ and $b < x_{N+1}$, let us define linear basis splines (B-splines for short)

$$B_{j,1}(x) = (x_{j+2} - x_j)\varphi[x, x_j, x_{j+1}, x_{j+2}], \quad j = -1, \ldots, N-1,$$

where $\varphi(x, y) = (x - y)_+ = \max(0, x - y)$, or according to formula (1.9)

$$\begin{aligned} B_{j,1}(x) =& (x_{j+2} - x_j) \sum_{k=j}^{j+2} \frac{(x - x_k)_+}{\omega'_{j,2}(x_k)}, \quad j = -1, \ldots, N-1, \\ \omega_{j,2}(x) =& (x - x_j)(x - x_{j+1})(x - x_{j+2}). \end{aligned}$$

The functions $B_{j,1}$ can be written in the form

$$B_{j,1}(x) = \begin{cases} \dfrac{x - x_j}{h_j}, & x_j \le x < x_{j+1}, \\ \dfrac{x_{j+2} - x}{h_{j+1}}, & x_{j+1} \le x < x_{j+2}, \\ 0, & \text{otherwise}. \end{cases} \tag{1.25}$$

It is easy to show (see chapter 3) that the functions $B_{j,1}$, $j = -1, \ldots, N-1$ are linearly independent on the interval $[a, b]$. Every Lagrange spline of first degree S_1^L can be uniquely represented in the form

$$S_1^L(x) = \sum_{j=-1}^{N-1} f_{j+1} B_{j,1}(x), \quad x \in [a, b]. \tag{1.26}$$

On the interval $[x_i, x_{i+1}]$ only the basis splines $B_{j,1}$, $j = i-1, i$ are different from zero in this sum and by formula (1.25), the representation (1.26) takes the form (1.24).

Let us also note that linear Lagrange splines are exact for polynomials of first degree, that is, every polynomial of first degree P_1 can be written in the form

$$P_1(x) = \sum_{j=-1}^{N-1} P_1(x_{j+1}) B_{j,1}(x), \quad x \in [a, b].$$

It is sufficient to verify this equality for the monomials 1 and x, that is, to show the validity of the identities

$$\sum_{j=-1}^{N-1} x_{j+1}^{\alpha} B_{j,1}(x) \equiv x^{\alpha}, \quad \alpha = 0, 1, \quad x \in [a, b].$$

This can be easily done by using formula (1.25).

Example 1.3. Let the function f be interpolated on the interval $[a, b]$ by a linear Lagrange spline on the set of equally spaced nodes $x_i = a + i(b - a)/N$, $i = 0, \dots, N$, and suppose that in the evaluation of f, round-off errors do not exceed $\varepsilon > 0$. How many interpolation nodes have to be chosen to provide an exactness of approximation E ($\varepsilon < E$)?

Solution. Let $f(x_i) = \bar{f}_i + \varepsilon_i$, $i = 0, \dots, N$, where ε_i is a round-off error. Let us set $h = (b - a)/N$. Using equality (1.5), on the interval $[x_i, x_{i+1}]$, $i = 0, \dots, N - 1$, we have

$$
\begin{aligned}
|f(x) - L_{i,1}(x)| &= \left| f(x) - \bar{f}_i \frac{x_{i+1} - x}{h} + \bar{f}_{i+1} \frac{x - x_i}{h} \right| \\
&= \left| f(x) - f(x_i) \frac{x_{i+1} - x}{h} - f(x_{i+1}) \frac{x - x_i}{h} \right. \\
&\quad \left. + \varepsilon_i \frac{x_{i+1} - x}{h} + \varepsilon_{i+1} \frac{x - x_i}{h} \right|
\end{aligned}
$$

$$
\begin{aligned}
&\le \frac{1}{2}(x_{i+1} - x)(x - x_i)|f''(\xi_x)| + \max(|\varepsilon_i|, |\varepsilon_{i+1}|) \\
&\le \frac{h^2}{8} \max_{x_i \le x \le x_{i+1}} |f''(x)| + \varepsilon \le E.
\end{aligned}
$$

Let $\max_{a \le x \le b} |f''(x)| = M$. As in our case $h = (b - a)/N$, we have the estimate

$$
\frac{1}{8}\left(\frac{b - a}{N}\right)^2 M \le E - \varepsilon \quad \text{or} \quad N \ge \frac{(b - a)M^{1/2}}{[8(E - \varepsilon)]^{1/2}}.
$$

Example 1.4. Under the conditions of example 1.3, what number of interpolation nodes provides the minimal error of approximation for f' on $[a, b]$?
Solution. On the interval $[x_i, x_{i+1}]$, $i = 0, \dots, N - 1$, one has

$$
\begin{aligned}
f'(x) - L'_{i,1}(x) &= f'(x) - \frac{\bar{f}_{i+1} - \bar{f}_i}{h} \\
&= f'(x) - \frac{f(x_{i+1}) - f(x_i)}{h} + \frac{\varepsilon_{i+1} - \varepsilon_i}{h}.
\end{aligned}
$$

Using the Taylor expansion we obtain

$$
\begin{aligned}
f(x_i) &= f(x) + f'(x)(x_i - x) + f''(\xi_1)\frac{(x_i - x)^2}{2}, & \xi_1 \in (x_i, x), \\
f(x_{i+1}) &= f(x) + f'(x)(x_{i+1} - x) + f''(\xi_2)\frac{(x_{i+1} - x)^2}{2}, & \xi_2 \in (x, x_{i+1}).
\end{aligned}
$$

Hence

$$
\frac{f(x_{i+1}) - f(x_i)}{h} = f'(x) + f''(\xi_2)\frac{(x_{i+1} - x)^2}{2h} - f''(\xi_1)\frac{(x_i - x)^2}{2h}.
$$

This gives us the estimate

$$\left|f'(x) - \frac{f(x_{i+1}) - f(x_i)}{h}\right| \le \frac{(x-x_i)^2 + (x_{i+1}-x)^2}{2h} \max_{x_i \le x \le x_{i+1}]} |f''(x)|.$$

Then

$$\begin{aligned} |f'(x) - L'_{i,1}(x)| &\le \frac{h}{2} \max_{x_i \le x \le x_{i+1}} |f''(x)| + \frac{|\varepsilon_i| + |\varepsilon_{i+1}|}{h} \\ &\le \frac{h}{2} M + \frac{2\varepsilon}{h} = \varphi(h, \varepsilon). \end{aligned}$$

The function φ takes a minimal value with respect to h if $\varphi'(h,\varepsilon) = M/2 - 2\varepsilon/h^2 = 0$ or $h = 2(\varepsilon/M)^{1/2}$. As $h = (b-a)/N$ then we have to choose $N \ge (M/\varepsilon)^{1/2}(b-a)/2$.

7. Interpolation by Cubic Lagrange Splines

The approximation can be improved by replacing piecewise linear interpolation by piecewise cubic Lagrange polynomials. Suppose we have data (x_i, f_i), $i = -1, \ldots, N+1$. To obtain a cubic Lagrange spline, one takes the cubic Lagrange polynomial on every interval $[x_i, x_{i+1}]$, $i = 0, \ldots, N-1$,

$$L_{i,3}(x) = \sum_{j=i-1}^{i+2} f_j \frac{\omega_{i-1,3}(x)}{(x-x_j)\omega'_{i-1,3}(x_j)}, \quad \omega_{i-1,3}(x) = \prod_{j=i-1}^{i+2} (x - x_j). \tag{1.27}$$

On the whole interval $[a, b]$, we have a set of N cubic Lagrange polynomials forming a continuous function which is called a *cubic Lagrange spline*. If we do not have the endpoint data (x_j, f_j), $j = -1, N+1$, then we can extend the polynomial $L_{1,3}$ to the interval $[x_0, x_2]$ and the polynomial $L_{N-2,3}$ to the interval $[x_{N-2}, x_N]$. However, in this case the goodness of approximation on the intervals $[x_0, x_1]$ and $[x_{N-1}, x_N]$ will be lower (see [19]).

Using formula (1.5) on the interval $[x_i, x_{i+1}]$ one has the estimate

$$|f(x) - L_{i,3}(x)| \le \frac{1}{4!}|\omega_{i-1,3}(x)| \max_{x_{i-1} \le x \le x_{i+2}} |f^{(4)}(x)| \le \frac{9}{384}\bar{h}^4 ||f^{(4)}||_{C[a,b]}, \tag{1.28}$$

where $\bar{h} = \max_i h_i$ and $||f||_{C[a,b]} = \max_{a \le x \le b} |f(x)|$.

Setting $x_{-3} < x_{-2} < x_{-1} < a$ and $b < x_{N+1} < x_{N+2} < x_{N+3}$, let us consider the cubic Lagrange B-splines

$$B^L_{j,3}(x) = \begin{cases} \dfrac{\omega_{k-1,3}(x)}{(x - x_{j+2})\omega'_{k-1,3}(x_{j+2})}, & \text{if } x \in [x_k, x_{k+1}], \\ & k = j, \ldots, j+3, \\ 0, & \text{otherwise}, \end{cases} \tag{1.29}$$

$j = -3, \ldots, N-1$. The graph of the spline $B^L_{j,3}$ with equally spaced nodes $x_i = i$, $i = 1, \ldots, 5$ is shown on Figure 1.3.

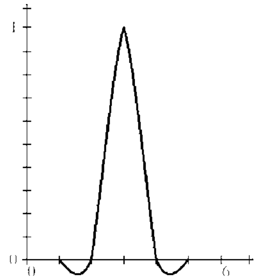

Figure 1.3. Cubic Lagrange B-spline with equally spaced nodes $x_i = i$, $i = 1, \ldots, 5$.

It is easy to show (see chapter 3) that the functions $B^L_{j,3}$, $j = -3, \ldots, N-1$ are linearly independent. Every cubic Lagrange spline S^L_3 can be uniquely written in the form

$$S^L_3(x) = \sum_{j=-3}^{N-1} f_{j+2} B^L_{j,3}(x), \quad x \in [a,b]. \tag{1.30}$$

In this sum only the B-splines $B^L_{j,3}$, $j = i-3, \ldots, i$ will be different from zero on the interval $[x_i, x_{i+1}]$. Using formula (1.29) we verify that the representation (1.30) coincides with formula (1.27) on the interval $[x_i, x_{i+1}]$

$$S^L_3(x) = \sum_{j=-3}^{N-1} f_{j+2} B^L_{j,3}(x) = \sum_{j=i-3}^{i} f_{j+2} B^L_{j,3}(x) = \sum_{j=i-1}^{i+2} \frac{f_j\, \omega_{i-1,3}(x)}{(x - x_j)\omega'_{i-1,3}(x_j)}. \tag{1.31}$$

Cubic Lagrange splines are exact on cubic polynomials, that is, every cubic polynomial P_3 can be uniquely represented in the form

$$P_3(x) = \sum_{j=-3}^{N-1} P_3(x_{j+2}) B^L_{j,3}(x), \quad x \in [a,b].$$

To prove this formula we verify it on the monomials x^α, $\alpha = 0, 1, 2, 3$, that is, we show that the following equalities are valid

$$x^\alpha = \sum_{j=-3}^{N-1} x^\alpha_{j+2} B^L_{j,3}(x), \quad \alpha = 0, 1, 2, 3, \quad x \in [a,b], \tag{1.32}$$

or in equivalent form

$$(y-x)^3 = \sum_{j=-3}^{N-1} (y-x_{j+2})^3 B^L_{j,3}(x), \quad x \in [a,b].$$

As in (1.31) we have on the interval $[x_i, x_{i+1}]$, $i = 0, \ldots, N-1$, that

$$\sum_{j=-3}^{N-1} x^\alpha_{j+2} B^L_{j,3}(x) = \sum_{j=i-1}^{i+2} \frac{x^\alpha_j\, \omega_{i-1,3}(x)}{(x-x_j)\omega'_{i-1,3}(x_j)} = x^\alpha, \quad \alpha = 0,1,2,3.$$

This proves the equalities (1.32).

Unfortunately, on a coarse mesh the graph of a cubic Lagrange spline can have corners as the derivatives of consecutive polynomials are not ajusted smoothly. An exception is the case of equally spaced nodes ($h_i = h$ for all i) where the second derivative of a cubic Lagrange spline turns to be continuous.

Let us use a simple approach to show how we can smoothly adjust consecutive cubic Lagrange polynomials to obtain a smooth function while still providing practically the same accuracy as with Lagrange interpolating spline.

8. Local Approximation by Cubic Lagrange Splines

Let us consider a "corrected" cubic Lagrange polynomial on the interval $[x_i, x_{i+1}]$, $i = 0, \ldots, N-1$,

$$S_{i,3}(x) = L_{i,3}(x) + C_{i,1}(x-x_i)^3 + C_{i,2}(x_{i+1}-x)^3.$$

We will assume that

$$S^{(r)}_{i-1,3}(x_i - 0) = S^{(r)}_{i,3}(x_i + 0), \quad r = 0,1,2, \quad i = 1, \ldots, N-1. \tag{1.33}$$

Let us write the consecutive cubic Lagrange polynomials on the intervals $[x_{i-1}, x_i]$ and $[x_i, x_{i+1}]$ in the form

$$\begin{aligned} S_{i-1,3}(x) = {} & f_{i-1} + f[x_{i-1}, x_i](x - x_{i-1}) \\ & + f[x_{i-1}, x_i, x_{i+1}](x-x_{i-1})(x-x_i) \\ & + f[x_{i-2}, x_{i-1}, x_i, x_{i+1}](x-x_{i-1})(x-x_i)(x-x_{i+1}) \\ & + C_{i-1,1}(x-x_{i-1})^3 + C_{i-1,2}(x_i - x)^3, \\ S_{i,3}(x) = {} & f_{i-1} + f[x_{i-1}, x_i](x - x_{i-1}) \\ & + f[x_{i-1}, x_i, x_{i+1}](x-x_{i-1})(x-x_i) \\ & + f[x_{i-1}, x_i, x_{i+1}, x_{i+2}](x-x_{i-1})(x-x_i)(x-x_{i+1}) \\ & + C_{i,1}(x-x_i)^3 + C_{i,2}(x_{i+1}-x)^3. \end{aligned}$$

Subtracting these polynomials one has

$$\begin{aligned} S_{i,3}(x) - S_{i-1,3}(x) = {} & \theta_{i,4}(x-x_{i-1})(x-x_i)(x-x_{i+1}) \\ & + (C_{i,1} + C_{i-1,2})(x-x_i)^3 + C_{i,2}(x_{i+1}-x)^3 - C_{i-1,1}(x-x_{i-1})^3, \end{aligned}$$

where $\theta_{i,4} = (x_{i+2} - x_{i-2}) f[x_{i-2}, \ldots, x_{i+2}]$.

Hence, using the conditions (1.33) one obtains the system of equations

$$\begin{aligned} h_{i-1}^3 C_{i-1,1} - h_i^3 C_{i,2} &= 0, \\ 3h_{i-1}^2 C_{i-1,1} + 3h_i^2 C_{i,2} &= -h_{i-1} h_i \theta_{i,4}, \\ 3h_{i-1} C_{i-1,1} - 3h_i C_{i,2} &= (h_{i-1} - h_i)\theta_{i,4}. \end{aligned} \tag{1.34}$$

The equations in the overdetermined system (1.34) are linearly dependent. This system has a unique solution

$$C_{i-1,1} = -\frac{h_i^2 \theta_{i,4}}{3h_{i-1}(h_{i-1} + h_i)}, \quad C_{i,2} = \left(\frac{h_{i-1}}{h_i}\right)^3 C_{i-1,1}.$$

Hence, a smooth cubic Lagrange spline takes the form

$$S_{i,3}(x) = L_{i,3}(x) - \frac{h_{i+1}^2 \theta_{i+1,4}}{3h_i(h_i + h_{i+1})}(x - x_i)^3 - \frac{h_{i-1}^2 \theta_{i,4}}{3h_i(h_{i-1} + h_i)}(x_{i+1} - x)^3 \tag{1.35}$$

on the interval $[x_i, x_{i+1}]$. We loose the property of interpolation. Instead, we have the property of local approximation. Let us show that nevertheless the accuracy of approximation will practically be the same as for the cubic Lagrange interpolating spline.

Using (1.14) we can rewrite formula (1.35) in the form

$$\begin{aligned} f(x) - S_{i,3}(x) =& f[x_{i-1}, \ldots, x_{i+2}, x]\omega_{i-1,3}(x) + \frac{h_{i+1}^2 \theta_{i+1,4}}{3h_i(h_i + h_{i+1})}(x - x_i)^3 \\ &+ \frac{h_{i-1}^2 \theta_{i,4}}{3h_i(h_{i-1} + h_i)}(x_{i+1} - x)^3 \\ =& \Big[\omega_{i-1,3}(x) + \frac{h_{i+1}^2 (x_{i+3} - x_{i-1})}{3h_i(h_i + h_{i+1})}(x - x_i)^3 + \frac{h_{i-1}^2 (x_{i+2} - x_{i-2})}{3h_i(h_{i-1} + h_i)} \\ &\times (x_{i+1} - x)^3\Big] f[x_{i-1}, \ldots, x_{i+2}, \xi], \quad \xi \in [x_{i-2}, x_{i+3}]. \end{aligned}$$

Hence, one has the estimate for $x \in [x_i, x_{i+1}]$

$$\begin{aligned} |f(x) - S_{i,3}(x)| \le& \Big[t^2(1-t)^2 + \frac{2}{3}\Big]\bar{h}_i^4 \max_{x_{i-2} \le \xi \le x_{i+3}} |f[x_{i-1}, \ldots, x_{i+2}, \xi]| \\ \le& \frac{35}{48}\bar{h}_i^4 \max_{x_{i-2} \le \xi \le x_{i+3}} |f[x_{i-1}, \ldots, x_{i+2}, \xi]|, \end{aligned}$$

where $\bar{h}_i = \max_{|i-j| \le 2} h_j$ and $t = (x - x_i)/h_i$.

Using equality (1.15) we can rewrite this estimate in the form

$$|f(x) - S_{i,3}(x)| \le \Big[t^2(1-t)^2 + \frac{2}{3}\Big]\frac{\bar{h}_i^4}{24} M \le \frac{35}{1152}\bar{h}^4 M, \tag{1.36}$$

where $M = ||f^{(4)}||_{C[a,b]}$.

Comparing now the estimates (1.28) and (1.36) we conclude that when replacing a cubic Lagrange interpolating spline by a local approximating one obtains practically the same accuracy of approximation (compared with the estimate (1.28), the constant in the estimate (1.36) is increased only slightly).

Applications of local approximation methods to the problems of computer aided geometric design (CAGD for short) are described in [6,9].

9. Local Approximation by Cubic B-Splines

Let us use one more approach [20] for obtaining the formula of local approximation (1.35). Let us consider cubic B-splines

$$B_{j,3}(x) = (x_{j+4} - x_j)\varphi_3[x, x_j, \ldots, x_{j+4}], \quad \varphi_3(x, y) = (x - y)^3_+,$$

$j = -3, \ldots, N-1$. The graph of the cubic B-spline $B_{j,3}$ with equally spaced nodes $x_i = i$, $i = 1, \ldots, 5$ is shown on Figure 1.4.

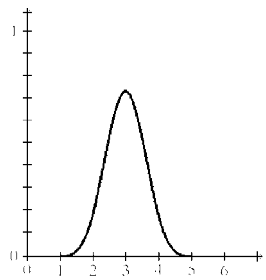

Figure 1.4. Cubic B-spline $B_{j,3}$ with equally spaced nodes $x_i = i$, $i = 1, \ldots, 5$.

Using formula (1.9) one can also rewrite the spline $B_{j,3}$ in the form

$$B_{j,3}(x) = (x_{j+4} - x_j) \sum_{k=j}^{j+4} \frac{(x - x_k)^3_+}{\omega'_{j,4}(x_k)}, \quad \omega_{j,4}(x) = \prod_{k=j}^{j+4} (x - x_k).$$

It is easy to show (see chapter 3) that the functions $B_{j,3}$, $j = -3, \ldots, N-1$ are linearly independent on $[a, b]$ and have the properties

$$B_{j,3}(x) \begin{cases} > 0, & \text{if } x \in (x_j, x_{j+4}), \\ 0, & \text{otherwise,} \end{cases}$$

$$(y - x)^3 = \sum_{j=-3}^{N-1} (y - x_{j+1})(y - x_{j+2})(y - x_{j+3}) B_{j,3}(x), \quad x \in [a, b]. \tag{1.37}$$

Equality (1.37) can also be rewritten in the equivalent form

$$x^\alpha = \frac{1}{C_3^\alpha} \sum_{j=-3}^{N-1} symm_\alpha(x_{j+1}, x_{j+2}, x_{j+3}) B_{j,3}(x), \quad x \in [a, b], \tag{1.38}$$

where $C_3^\alpha = \binom{3}{\alpha}$ is the usual binomial coefficient and

$$\begin{aligned}
symm_0(x,y,z) &= 1,\\
symm_1(x,y,z) &= x+y+z,\\
symm_2(x,y,z) &= xy+xz+yz,\\
symm_3(x,y,z) &= xyz.
\end{aligned}$$

Let us consider the following formula of local approximation by cubic B-splines

$$S_f(x) = \sum_{j=-3}^{N-1} b_{j+2} B_{3,j}(x), \tag{1.39}$$

where $b_j = b_{j,-1} f_{j-1} + b_{j,0} f_j + b_{j,1} f_{j+1}$. If we set $b_{j,0} = 1 - b_{j,-1} - b_{j,1}$,

$$b_{j,-1} = -\frac{h_j^2}{3h_{j-1}(h_{j-1}+h_j)}, \quad b_{j,1} = -\frac{h_{j-1}^2}{3h_j(h_{j-1}+h_j)},$$

then formula (1.39) will be exact for cubic polynomials. To verify this property, one can use the monomials x^α, $\alpha = 0,1,2,3$. Substituting these monomials into (1.39) we obtain the equalities (1.38).

According to formula (1.14) one has

$$f(x) = L_{i,3}(x) + R_{i,3}(x),$$

where $R_{i,3}(x) = f[x_{i-1},\ldots,x_{i+2},x]\omega_{i-1,3}(x)$.

As the spline S_f is exact for cubic polynomials, then

$$S_f(x) = L_{i,3}(x) + S_{R_{i,3}}(x).$$

Using formula (1.39) one has on the interval $[x_i, x_{i+1}]$

$$\begin{aligned}
S_{R_{i,3}}(x) = b_{i-1,-1}R_{i,3}(x_{i-2})B_{i-3,3}(x) + b_{i+2,1}R_{i,3}(x_{i+3})B_{i,3}(x)\\
+ \sum_{j=i-1}^{i+2} \psi_j(x) R_{i,3}(x_j),
\end{aligned}$$

where ψ_j are some cubic polynomials. As $R_{i,3}(x_j) = 0$ for $j = i-1,\ldots,i+2$ then

$$S_f(x) = L_{i,3}(x) + b_{i-1,-1}R_{i,3}(x_{i-2})B_{i-3,3}(x) + b_{i+2,1}R_{i,3}(x_{i+3})B_{i,3}(x).$$

Substituting here the expressions for B-splines and for the remainder we again obtain the formula (1.35).

10. Interpolation by Quadratic Lagrange Splines

One can also perform the interpolation by piecewise quadratic Lagrange polynomials. Suppose one has data (x_i, f_i) with $i = -1, \ldots, N$ or $i = 0, \ldots, N+1$ and considers quadratic Lagrange polynomials on the intervals $[x_i, x_{i+1}]$, $i = 0, \ldots, N-1$,

$$L_{j,2}(x) = \sum_{k=j}^{j+2} f_k \frac{\omega_{j,2}(x)}{(x - x_k)\, \omega'_{j,2}(x_k)} \tag{1.40}$$

for $j = i-1$ or $j = i$ correspondingly. This gives us a set of N quadratic Lagrange polynomials forming a continuous function on $[a, b]$ which is also called a *quadratic Lagrange spline*. If we have only the data (x_i, f_i), $i = 0, \ldots, N$, then one can extend the polynomial $L_{0,2}$ to the interval $[x_0, x_2]$ or the polynomial $L_{N-2,2}$ to the interval $[x_{N-2}, x_N]$. Let us assume that we have the data (x_i, f_i), $i = 0, \ldots, N+1$.

Using formula (1.5) one has on the interval $[x_i, x_{i+1}]$

$$|f(x) - L_{i,2}(x)| \le \frac{1}{3!} |\omega_{i,2}(x)| \max_{x_i \le x \le x_{i+2}} |f^{(3)}(x)| \le \frac{\sqrt{3}}{9} \bar{h}_i^3 ||f^{(3)}||_{C[a,b]}. \tag{1.41}$$

Setting $x_{-2} < x_{-1} < a$ and $b < x_{N+1} < x_{N+2}$ let us define quadratic Lagrange B-splines

$$B^L_{j,2}(x) = \begin{cases} \dfrac{\omega_{k,2}(x)}{(x - x_{j+2})\, \omega'_{k,2}(x_{j+2})}, & \text{if } x \in [x_k, x_{k+1}], \\ & k = j, j+1, j+2, \\ 0, \quad \text{otherwise}, & \end{cases} \tag{1.42}$$

$j = -2, \ldots, N-1$. The graph of a quadratic Lagrange B-spline $B_{j,2}$ with equally spaced nodes $x_i = i$, $i = 1, \ldots, 4$ is given on Figure 1.5.

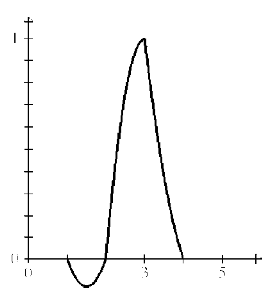

Figure 1.5. Quadratic Lagrange B-spline $B_{j,2}$ with equally spaced nodes $x_i = i$, $i = 1, \ldots, 4$.

It is easy to show (see chapter 3) that the functions $B^L_{j,2}$, $j = -2, \dots, N-1$ are linearly independent. Any quadratic Lagrange spline S^L_2 can be uniquely written in the form

$$S^L_2(x) = \sum_{j=-2}^{N-1} f_{j+2} B^L_{j,2}(x), \quad x \in [a,b]. \tag{1.43}$$

In this sum only the B-splines $B^L_{j,2}$, $j = i-2, i-1, i$ will be different from zero on the interval $[x_i, x_{i+1}]$. Using formula (1.42) we verify that the representation (1.43) coincides with formula (1.40) on the interval $[x_i, x_{i+1}]$.

Quadratic Lagrange splines are exact for quadratic polynomials, that is, for any quadratic polynomial P_2 the following representation is valid

$$P_2(x) = \sum_{j=-2}^{N-1} P_2(x_{j+2}) B^L_{j,2}(x), \quad x \in [a,b].$$

Again, we prove this formula by showing that it is valid on the monomials x^α, $\alpha = 0, 1, 2$, that is,

$$x^\alpha = \sum_{j=-2}^{N-1} x^\alpha_{j+2} B^L_{j,2}(x), \quad \alpha = 0,1,2, \quad x \in [a,b] \tag{1.44}$$

or in the equivalent form

$$(y-x)^2 = \sum_{j=-2}^{N-1} (y - x_{j+2})^2 B^L_{j,2}(x), \quad x \in [a,b].$$

The equalities (1.44) can be verified directly by using formula (1.42). For $x \in [x_i, x_{i+1}]$, $i = 0, \dots, N-1$, one has

$$\sum_{j=-2}^{N-1} x^\alpha_{j+2} B^L_{j,2}(x) = \sum_{j=i}^{i+2} \frac{x^\alpha_j \, \omega_{i,2}(x)}{(x - x_j)\, \omega'_{i,2}(x_j)} = x^\alpha, \quad \alpha = 0,1,2.$$

11. Local Approximation by Quadratic Lagrange Splines

The derivative of a quadratic Lagrange spline is a discontinuous function. To obtain a smooth quadratic spline let us apply the same approach as used in section 1.8 for cubic Lagrange splines.

Let us consider a "corrected" quadratic Lagrange polynomial on the interval $[x_i, x_{i+1}]$, $i = 0, \dots, N-1$,

$$S_{i,2}(x) = L_{i,2}(x) + C_{i,1}(x - x_i)^2 + C_{i,2}(x_{i+1} - x)^2.$$

We will assume that

$$S^{(r)}_{i-1,2}(x_i - 0) = S^{(r)}_{i,2}(x_i + 0), \quad r = 0,1, \quad i = 1, \dots, N-1. \tag{1.45}$$

Let us write the polynomials belonging to the consecutive intervals $[x_{i-1}, x_i]$ and $[x_i, x_{i+1}]$ in the form

$$
\begin{aligned}
S_{i-1,2}(x) &= f_i + f[x_i, x_{i+1}](x - x_i) \\
&\quad + f[x_{i-1}, x_i, x_{i+1}](x - x_i)(x - x_{i+1}) \\
&\quad + C_{i-1,1}(x - x_{i-1})^2 + C_{i-1,2}(x_i - x)^2, \\
S_{i,2}(x) &= f_i + f[x_i, x_{i+1}](x - x_i) \\
&\quad + f[x_i, x_{i+1}, x_{i+2}](x - x_i)(x - x_{i+1}) \\
&\quad + C_{i,1}(x - x_i)^2 + C_{i,2}(x_{i+1} - x)^2.
\end{aligned}
$$

Subtracting these polynomials one obtains

$$
\begin{aligned}
S_{i,2}(x) - S_{i-1,2}(x) &= \theta_{i,3}(x - x_i)(x - x_{i+1}) + (C_{i,1} - C_{i-1,2})(x - x_i)^2 \\
&\quad + C_{i,2}(x_{i+1} - x)^2 - C_{i-1,1}(x - x_{i-1})^2,
\end{aligned}
$$

where $\theta_{i,3} = (x_{i+2} - x_{i-1})f[x_{i-1}, \dots, x_{i+2}]$.

Hence, using the condition (1.45) one has the system of equations

$$
\begin{aligned}
h_{i-1}^2 C_{i-1,1} - h_i^2 C_{i,2} &= 0, \\
2h_{i-1}C_{i-1,1} + 2h_i C_{i,2} &= -h_i\theta_{i,3},
\end{aligned}
$$

from which

$$
C_{i,2} = -\frac{h_{i-1}\theta_{i,3}}{2(h_{i-1} + h_i)}, \quad C_{i-1,1} = \left(\frac{h_i}{h_{i-1}}\right)^2 C_{i,2}.
$$

Thus, on the interval $[x_i, x_{i+1}]$ the smooth quadratic Lagrange spline takes the form

$$
S_{i,2}(x) = L_{i,2}(x) - \frac{h_{i+1}^2\theta_{i+1,3}}{2h_i(h_i + h_{i+1})}(x - x_i)^2 - \frac{h_{i-1}\theta_{i,3}}{2(h_{i-1} + h_i)}(x_{i+1} - x)^2. \tag{1.46}
$$

Let us estimate the error of approximation by formula (1.46). Using (1.14) one has

$$
\begin{aligned}
f(x) - S_{i,2}(x) =& f[x_i, x_{i+1}, x_{i+2}, x]\omega_{i,2}(x) + \frac{h_{i+1}^2\theta_{i+1,3}}{2h_i(h_i + h_{i+1})}(x - x_i)^2 \\
&+ \frac{h_{i-1}\theta_{i,3}}{2(h_{i-1} + h_i)}(x_{i+1} - x)^2 \\
=& \Big[\omega_{i,2}(x) + \frac{h_{i+1}^2(x_{i+3} - x_i)}{2h_i(h_i + h_{i+1})}(x - x_i)^2 + \frac{h_{i-1}(x_{i+2} - x_{i-1})}{2(h_{i-1} + h_i)} \\
&\times (x_{i+1} - x)^2\Big] f[x_i, x_{i+1}, x_{i+2}, \xi], \quad \xi \in [x_{i-1}, x_{i+3}].
\end{aligned}
$$

From here, for $x \in [x_i, x_{i+1}]$ we have the estimate

$$
\begin{aligned}
|f(x) - S_{i,2}(x)| \le& \Big[t(1 - t)(2 - t) + \frac{3}{4}\Big]\bar{h}_i^3 \max_{x_{i-1} \le \xi \le x_{i+3}} |f[x_i, x_{i+1}, x_{i+2}, \xi]| \\
\le& \frac{9 + 8\sqrt{3}}{12}\bar{h}_i^3 \max_{x_{i-1} \le \xi \le x_{i+3}} |f[x_i, x_{i+1}, x_{i+2}, \xi]|,
\end{aligned}
$$

where $\bar{h}_i = \max_j h_j$, $i-1 \le j \le i+2$ and $t = (x - x_i)/h_i$.

Using equality (1.15) one can rewrite this estimate in the form

$$|f(x) - S_{i,2}(x)| \le \left[t(1-t)(2-t) + \frac{3}{4}\right]\frac{\bar{h}_i^3}{6}M \le \frac{9+8\sqrt{3}}{72}\bar{h}^3 M,$$

where $M = ||f^{(3)}||_{C[a,b]}$.

12. The Case of Discrete Cubic Spline

Let us consider the case where the functions Φ_j and Ψ_j are chosen by formula (10.4) and where we use the coefficients from (10.29) in the representation (10.24) (or (10.25))

$$b_{j-2} = b_{j,-1}f_{j-1} + b_{j,0}f_j + b_{j,1}f_{j+1} \tag{10.30}$$

with

$$b_{j,-1} = -\frac{\Phi_j(x_j)}{c_{j-1,2}h_{j-1}}, \quad b_{j,1} = -\frac{\Psi_{j-1}(x_j)}{c_{j-1,2}h_j}, \quad b_{j,0} = 1 - b_{j,-1} - b_{j,1}. \tag{10.31}$$

On the interval $[x_i, x_{i+1}]$, this spline S_f depends on only 6 values $f_{i-2}, \ldots, f_{i+3}$ of the function f. Let us construct a cubic Lagrange polynomial $L_{i,3}$ interpolating f in the points $x_{i-1}, \ldots, x_{i+2}$. Then

$$f(x) = L_{i,3}(x) + R_{i,3}(x) \tag{10.32}$$

with

$$R_{i,3}(x) = Q_{i,4}(x)f[x_{i-1}, \ldots, x_{i+2}, x],$$
$$Q_{4,i}(x) = (x - x_{i-1})(x - x_i)(x - x_{i+1})(x - x_{i+2}).$$

By means of detailed calculation it is possible to verify that the discrete cubic spline S_f is exact for cubic polynomials. When $\tau_j^{L_i} = \tau_j^{R_i} = \tau$, $j = i, i+1$ for all i such a property was proved in Lyche [12]. This permits us to write down

$$S_f(x) = L_{i,3}(x) + S_{R_{i,3}}(x),$$

where $S_{R_{i,3}}$ is a discrete cubic spline constructed by the values of the remainder $R_{i,3}$.

According to the formula (10.24) one can write

$$S_{R_{i,3}}(x) = b_{i-1,-1}R_{i,3}(x_{i-2})\mathrm{B}_{i-3}(x) + b_{i+2,1}R_{i,3}(x_{i+3})\mathrm{B}_i(x)$$
$$+ \sum_{j=i-1}^{i+2} p_j(x)R_{i,3}(x_j), \quad x \in [x_i, x_{i+1}],$$

where p_j, $j = i-1, \ldots, i+2$, are some cubic polynomials. However $R_{i,3}(x_j) = 0$, $j = i-1, \ldots, i+2$. Therefore on the interval $[x_i, x_{i+1}]$,

$$S_f(x) = L_{i,3}(x) + b_{i-1,-1}R_{i,3}(x_{i-2})\mathrm{B}_{i-3}(x) + b_{i+2,1}R_{i,3}(x_{i+3})\mathrm{B}_i(x).$$

Using formulae (10.15), (10.30), and (10.31), this representation can be rewritten as

$$S_f(x) = L_{i,3}(x) - Q_{i,4}(x_{i-2})f[x_{i-2},\dots,x_{i+2}]\frac{\Phi_{i-1}(x_{i-1})\Phi_i(x)}{h_{i-2}c_{i-2,2}c_{i-1,2}c_{i-2,3}} - Q_{i,4}(x_{i+3})f[x_{i-1},\dots,x_{i+3}]\frac{\Psi_i(x)\Psi_{i+1}(x_{i+2})}{h_{i+2}c_{i,2}c_{i+1,2}c_{i,3}}. \qquad (10.33)$$

If $\tau_j^{L_j} = h_{j-1}$, $\tau_j^{R_j} = h_j = \tau_{j+1}^{L_j}$, $\tau_{j+1}^{R_j} = h_{j+1}$ for all j, then $c_{j-1,k} = (x_{j+1} - x_{j+k-3})/(4-k)$, $k = 2,3$; and by formula (10.6) one obtains $\Phi_j(x_j) = \Psi_j(x_{j+1}) = 0$. Formula (10.29) (or (10.30)) gives $b_{j-2} = f_j$, and from (10.33) $S_f(x) = L_{i,3}(x)$. If $\tau_i^{L_j} = \tau_i^{R_j} = 0$, $i = j, j+1$ for all j, then S_f is a conventional cubic spline. In this case, $c_{j-1,k} = (x_{j+k-1} - x_{j-1})/k$, $k = 2,3$; $\Phi_j(x_j) = \Psi_j(x_{j+1}) = h_j^2$, and formula (10.33) turns into

$$S_f(x) = L_{i,3}(x) - (x_{i+2} - x_{i-2})f[x_{i-2},\dots,x_{i+2}]\frac{h_{i-1}^2h_i^2}{3(h_{i-1}+h_i)}(1-t)^3 - (x_{i+3} - x_{i-1})f[x_{i-1},\dots,x_{i+3}]\frac{h_i^2h_{i+1}^2}{3(h_i+h_{i+1})}t^3.$$

This result was obtained in chapter 1.

Let us now estimate the error of approximation. From (10.32) and (10.33)

$$e_i(x) = f(x) - S_f(x) = Q_{i,4}(x)f[x_{i-1},\dots,x_{i+2},x] + Q_{i,4}(x_{i-2})f[x_{i-2},\dots,x_{i+2}]\frac{\Phi_{i-1}(x_{i-1})\Phi_i(x)}{h_{i-2}c_{i-2,2}c_{i-1,2}c_{i-2,3}} + Q_{i,4}(x_{i+3})f[x_{i-1},\dots,x_{i+3}]\frac{\Psi_i(x)\Psi_{i+1}(x_{i+2})}{h_{i+2}c_{i,2}c_{i+1,2}c_{i,3}}. \qquad (10.34)$$

Let us denote $\bar h_i = \max_j h_j$, $|i-j| \le 2$. Then from (10.34) one obtains the estimate

$$|e_i(x)| \le C\bar h_i^4 \max_{x_{i-2}\le\zeta\le x_{i+3}} |f[x_{i-1},\dots,x_{i+2},\zeta]|$$

or, if $f \in C^4[x_{i-2}, x_{i+3}]$,

$$|e_i(x)| \le C_1\bar h_i^4||f^{(4)}||_{C,i},\ C_1 = C/24;\ ||g||_{C,i} = \max_{x_{i-2}\le x\le x_{i+3}} |g(x)|. \qquad (10.35)$$

Here $C_1 = 9/384$ if S_f coincides on $[x_i, x_{i+1}]$ with an interpolating cubic Lagrange polynomial $L_{i,3}$, and $C_1 = 35/1152$ if S_f is a cubic spline. The last one is usually called a *quasi-interpolant* because it gives the same order of approximation as a conventional cubic interpolational spline of chapter 2.

If $\tau_j^{L_j} = h_{j-1}$, $\tau_j^{R_j} = h_j = \tau_{j+1}^{L_j}$, $\tau_{j+1}^{R_j} = h_{j+1}$ for all j then using (10.4) and (10.17) one can rewrite the representation (10.15) in the form

$$\mathrm{B}_i(x) = \begin{cases} \dfrac{Q_{4,j}(x)}{(x - x_{i+2})Q'_{4,j}(x_{i+2})}, & \text{if } x \in [x_j, x_{j+1}), \\ & j = i, \ldots, i+3, \\ 0, & \text{otherwise.} \end{cases} \tag{10.36}$$

The basis spline (10.36) is also a fundamental discrete cubic spline with the property $\mathrm{B}_i(x_j) = \delta_{i+2,j}$. Any piecewise cubic Lagrange polynomial $S_f(x) = L_{3,i}(x)$ if $x \in [x_i, x_{i+1})$, $i = 0, \ldots, N-1$, where the cubic polynomial $L_{3,i}(x_j) = f_j \equiv f(x_j)$, $j = i-1, \ldots, i+2$, can be uniquely written in the form

$$S_f(x) = \sum_{j=-3}^{N-1} f_{j+2}\mathrm{B}_j(x), \quad x \in [a, b].$$

This formula is exact for cubic polynomials because using (10.36), we obtain in any interval $[x_i, x_{i+1}]$, $i = 0, \ldots, N-1$,

$$\sum_{j=i-3}^{i} x_{j+2}^r \mathrm{B}_j(x) = \sum_{j=i-1}^{i+2} \frac{x_j^r Q_{4,i}(x)}{(x - x_j)Q'_{4,i}(x_j)} = x^r, \quad r = 0, 1, 2, 3.$$

This proves the identity

$$(y - x)^3 = \sum_{j=-3}^{N-1} (y - x_{j+2})^3 \mathrm{B}_j(x), \quad x \in [a, b].$$

References

1. R. R. Akhmerov, J. L. Korobitsyna, and A. G. Sleptsov, *Fundamentals of Numerical Analysis in Problems* (Novosibirsk State University Press, Novosibirsk, 1994).
2. N. S. Bakhvalov, *Numerical Methods.* V. 1 (Nauka, Moscow, 1973, in Russian).
3. L. Collatz, J. Albrecht, *Aufgaben aus der Angewandten Mathematik I* (Akademie-Verlag, Berlin, 1972).
4. C. De Boor, *A Practical Guide to Splines* (Springer Verlag, New York, 1978).
5. B. P. Demidovich and I. A. Maron, *Fundamentals of Computational Mathematics* (Nauka, Moscow, 1970).

6. G. Farin, *Curves and Surfaces for Computer Aided Geometric Design. A Practical Guide* (Academic Press, San Diego, 1993).
7. I. D. Faux and M. J. Pratt, *Computational Geometry for Design and Manufacture* (Ellis Horwood, Chichester, UK, 1979).
8. G. E. Forsythe, M. A. Malcolm, and C. B. Moler, *Computer Methods for Mathematical Computations* (Englewood Cliffs, NJ: Prentice Hall, 1977).
9. J. Hoschek and D. Lasser, *Fundamentals of Computer Aided Geometric Design* (A K Peters, Wellesley, Massachusetts, 1993).
10. D. Kincaid and W. Cheney, *Numerical Analysis.* Second Edition (Brooks / Cole Publishing Company, Pasific Grove, California, 1996).
11. D. E. Knuth, *Seminumerical Algorithms. The Art of Computer Programming.* Vol. 2 (Addison-Wesley, Reading Mass., 1969).
12. T. Lyche, *Discrete polynomial spline approximation methods*, Thesis, University of Texas at Austin, 1975. For a summary, see *Spline Functions, Karlsruhe 1975*, eds. K. Böhmer, G. Meinardus, and W. Schimpp (Lecture Notes in Mathematics No. 501, Springer-Verlag, Berlin/Heidelberg/New York, 1976), pp. 144–176.
13. J. H. Mathews, *Numerical Methods for Mathematics, Science and Engineering* (Prentice-Hall Int. Editions, New York, 1992).
14. P. I. Monastyrnyi, *Problems on Computational Mathematics* (Vysheyshaya Shkola, Minsk, 1972).
15. H. Rutishauser, *Lectures on Numerical Mathematics* (Boston, Birkhäuser Verlag, 1990).
16. L. L. Schumaker, *Spline functions: Basic theory* (John Wiley & Sons, New York, 1981).
17. A. F. Shampine, R. C. Allen, Jr., and S. Pruess, *Fundamentals of Numerical Computing* (John Wiley & Sons, Inc., New York, 1997).
18. S. B. Stechkin and Yu. N. Subbotin, *Splines in Computational mathematics* (Nauka, Moscow, 1976, in Russian).
19. Yu. S. Zavyalov, B. I. Kvasov, and V. L. Miroshnichenko, *Methods of spline functions* (Nauka, Moscow, 1980, in Russian).
20. V. A. Zheludev, *Local spline approximation on arbitrary meshes*, Izv. Vysch. Uchebn. Zaved. Mat. **8** (1987) 14–18 (in Russian).