

ฉบับหน้าปก

รหัสโครงการ SUT1-103-42-12-17



รายงานการวิจัย

DB-Spline Approximation Through
Lagrange-Newton Polynomial



ศูนย์บรรณสารและสื่อการศึกษา

มหาวิทยาลัยเทคโนโลยีสุรนารี

คณะผู้วิจัย

หัวหน้าโครงการ

Assoc. Prof. Dr. Boris I. Kvasov

สาขาวิชาคณิตศาสตร์

สำนักวิชาวิทยาศาสตร์

มหาวิทยาลัยเทคโนโลยีสุรนารี

ได้รับเงินอุดหนุนการวิจัยจากมหาวิทยาลัยเทคโนโลยีสุรนารี ปีงบประมาณ พ.ศ. 2542

ผลงานวิจัยเป็นความรับผิดชอบของหัวหน้าโครงการวิจัยแต่เพียงผู้เดียว

มกราคม 2543

A c k n o w l e d g e m e n t

This is to acknowledge with many thanks the Suranaree University of Technology Research Fund Administration for financial support which made this project possible and led to its successful completion.



บทคัดย่อ

ในรายงานนี้เราพิจารณาการประมาณค่าในช่วงโดยพหุนามลากรานจ์ต้นแบบ และขยายคุณสมบัติคลุมบนช่วงย่อยที่เรียกว่า พหุนามลากรานจ์-สไปลน์ เราได้ศึกษา พหุนามลากรานจ์-สไปลน์อย่างละเอียด อาทิ เชิงเส้นบนช่วงย่อย กำลังสองบนช่วงย่อย และกำลังสามบนช่วงย่อย เราแสดงให้เห็นได้โดยง่ายว่า จะได้ลากรานจ์-สไปลน์เรียบ ซึ่งทราบกันทั่วไปว่าเป็นสไปลน์ประมาณค่าเฉพาะที่ ถึงแม้ว่าสไปลน์ประมาณค่าเฉพาะที่ นี้จะไม่มีคุณสมบัติประมาณค่าในช่วง แต่ก็ยังมีความถูกต้องในการประมาณค่าเช่นเดียวกับลากรานจ์-สไปลน์ จุดสำคัญในรายงานยังได้ทำการขยายกฎของฮอร์เนอร์มาประยุกต์ การหาค่าพหุนามลากรานจ์และค่าอนุพันธ์ ซึ่งวิธีขั้นตอนดังกล่าวไม่ค่อยปรากฏในหนังสือ วิธีการวิเคราะห์เชิงตัวเลขทั่วไป การเสนอเนื้อหาดังกล่าวได้มีตัวอย่างอธิบายประกอบ ด้วย

A b s t r a c t

In this report we consider the interpolation by classical Lagrange polynomials and by their piecewise generalizations which are usually called Lagrange splines. We study in details such common in practice Lagrange splines as piecewise linear, piecewise quadratic and piecewise cubic Lagrange polynomials. It is shown as using simple technique one can obtain smooth analogs of Lagrange splines known as local approximating splines. Although the local approximating splines do not have the interpolation property but they provide practically the same accuracy of approximation as the Lagrange splines. One central point in the report is also the generalization of Horner's rule for a simultaneous evaluation of Lagrange interpolation polynomial and its derivatives. Such algorithm is usually lacking in standard textbooks on numerical analysis. The presentation is illustrated by examples.



C o n t e n t s

Acknowledgement	i
Abstract in Thai	ii
Abstract in English	iii
Contents	iv
1. Introduction	1
2. Polynomial Interpolation Problem	1
3. Lagrange Interpolation Formula	2
4. Newton Interpolating Polynomial	4
5. Evaluation of the Newton Interpolating Polynomial and Its Derivatives by a Generalized Horner's Rule	7
6. Convergence of the Interpolating Polynomials	11
7. Piecewise Linear Interpolation	12
8. Interpolation by Cubic Lagrange Splines	15
9. Local Approximation by Cubic Lagrange Splines	17
10. Local Approximation by Cubic B-Splines	18
11. Interpolation by Quadratic Lagrange Splines	20
12. Local Approximation by Quadratic Lagrange Splines	22
References	24
Curriculum Vitae of Principal Investigator	25

1. Introduction

Polynomials and their smooth piecewise analogues 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 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 research is a generalization of Horner's rule for the simultaneous evaluation of the Newton interpolating polynomial and its derivatives. Such an algorithm is usually not found in standard textbooks on numerical analysis.

We 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 splines are usually called quasi-interpolants. We 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.

2. 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 $a = x_0 < x_1 < \dots < x_N = b$.

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

$$P_N(x_i) = \sum_{j=0}^N a_j x_i^j = f_i, \quad i = 0, \dots, N \quad (1)$$

are equivalent to the system of linear algebraic equations

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

The matrix of system (2) is called the Vandermonde matrix and its determinant is the Vandermonde determinant. In our case the Vandermonde determinant $D = \prod_{0 \leq i < j \leq N} (x_j - x_i)$ is nonzero, so that system (2) has a unique solution. This proves the existence and uniqueness of an interpolating polynomial of degree $\leq N$.

However a direct solution of system (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 (2) explicitly.

3. Lagrange Interpolation Formula

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

$$L_N(x) = \sum_{j=0}^N f_j l_j(x), \quad (3)$$

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

$$l_j(x) = \frac{\omega_N(x)}{(x - x_j)\omega'_N(x_j)}, \quad j = 0, \dots, N, \quad (4)$$
$$\omega_N(x) = (x - x_0)(x - x_1) \cdots (x - x_N).$$

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.

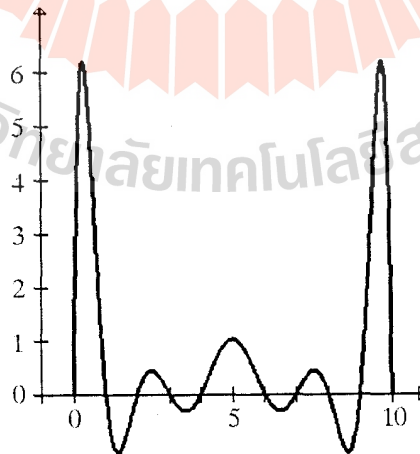


Figure 1. The graph of the Lagrange coefficient polynomial l_5 with $N = 10$ and nodes $x_i = i$, $i = 0, \dots, 10$.

One can easily verify that

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

According to (3) and (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. *The Lagrange interpolating polynomial is exact for polynomials of degree $\leq N$, that is, for any polynomial P_k of degree $k \leq N$ the following identity is valid,*

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

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

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

Now the polynomial of the degree N ,

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

has $N + 1$ zeros: $F_{k,N}(x_i) = 0, i = 0, \dots, N$. 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 $k \geq 0$, we denote by $C^k = C^k[a, b]$ the set of functions on $[a, b]$ which have k continuous derivatives.

Theorem 1. *Let f be a function in $C^{N+1}[a, b]$, and let L_N be the polynomial of degree $\leq N$ which interpolates the function f at $N + 1$ distinct points x_0, x_1, \dots, 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*

$$f(x) - L_N(x) = \frac{1}{(N + 1)!} f^{(N+1)}(\xi_x) \omega_N(x). \quad (5)$$

Proof: If x is one of the nodes of interpolation x_i , the assertion is obviously true since both sides of (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, \dots, x_N . By Rolle's theorem, Φ' has at least $N + 1$ distinct zeros 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 (5). This proves the theorem. \square

Example 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. 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 (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 $i = 0, \dots, k-1$ and $i = 1, \dots, k$ correspondingly.

Lemma 2. Let f be defined at x_0, x_1, \dots, x_k and x_0 and x_k be two distinct numbers in this set. Then the Aitken interpolation formula

$$L_{0,k}(x) = \frac{x_k - x}{x_k - x_0} L_{0,k-1}(x) + \frac{x - x_0}{x_k - x_0} L_{1,k-1}(x), \quad (6)$$

$k = 1, \dots, N,$

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

Proof: The polynomial on the right side of (6) has degree $\leq k$ and interpolates the data (x_i, f_i) , $i = 0, \dots, k$. 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. \square

4. 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,

$$L_k(x) = L_{k-1}(x) + f[x_0, \dots, x_k](x - x_0) \cdots (x - x_{k-1}). \quad (7)$$

As according to (3) and (4),

$$L_k(x) = \sum_{j=0}^k f_j \frac{\omega_k(x)}{(x - x_j)\omega'_k(x_j)}, \quad (8)$$

then by looking at each k th degree term in (8) and comparing with the coefficient of x^k in (7) we obtain

$$f[x_0, \dots, x_k] = \sum_{j=0}^k \frac{f_j}{\omega'_k(x_j)}. \quad (9)$$

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 (6) can be rewritten in the form

$$L_k(x) = L_{k-1}(x) + \frac{x - x_0}{x_k - x_0} [L_{1,k-1}(x) - L_{0,k-1}(x)]. \quad (10)$$

As

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

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 (10) and comparing with (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 (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

$$L_N(x) = c_0 + c_1(x - x_0) + \dots + c_N(x - x_0) \cdots (x - x_{N-1}), \quad (11)$$

where

$$\begin{aligned} c_0 &= f[x_0] \equiv f_0, \\ c_k &= \frac{f[x_1, \dots, x_k] - f[x_0, \dots, x_{k-1}]}{x_k - x_0}, \quad k = 1, \dots, N. \end{aligned} \quad (12)$$

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 (11),

$$L_{N+1}(x) = L_N(x) + f[x_0, \dots, x_N, t] \omega_N(x). \quad (13)$$

As $L_{N+1}(t) = f(t)$, then by setting $x = t$ in (13) we obtain

$$f(t) - L_N(t) = f[x_0, \dots, x_N, t] \omega_N(t). \quad (14)$$

Comparing this formula with (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,

$$f[x_0, \dots, x_n] = \frac{f^{(n)}(\xi)}{n!} \quad (15)$$

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) 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 [2,4,5,12].

5. 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 may replace the variable x by the expression $x = y + z$ in (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 $A_l = L_N^{(l)}(z)/l!$, $l = 0, \dots, N$.

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

$$L_N(x) = A_0 + A_1(x - z) + \cdots + A_N(x - z)^N. \quad (16)$$

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 $a_{k,0} = c_k$, $k = 0, \dots, N$ and rewrite the polynomial L_N in the form

$$L_N(x) \equiv P_{N,0}(x) = a_{0,0} + \sum_{k=1}^N a_{k,0} \omega_{k-1}(x), \quad \omega_{k-1}(x) = \prod_{i=0}^{k-1} (x - x_i). \quad (17)$$

By using parentheses we transform the representation (17) into the form

$$P_{N,0}(x) = a_{0,0} + (x - x_0)(a_{1,0} + \cdots + (x - x_{N-2})(a_{N-1,0} + (x - x_{N-1})a_{N,0}) \cdots). \quad (18)$$

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

$$\begin{aligned} a_{N,1} &= a_{N,0}, \\ a_{k,1} &= a_{k,0} + (z - x_k)a_{k+1,1}, \quad k = N - 1, \dots, 0, \end{aligned} \quad (19)$$

starting with the innermost parentheses in (18). It follows from (18) and (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 l th ($0 \leq l \leq N$) derivative of the Newton interpolating polynomial, let us consider the polynomial

$$\begin{aligned} P_{N,l}(x) &= a_{l,l} + \sum_{k=l+1}^N a_{k,l} \omega_{k-l-1}(x) \\ &= a_{l,l} + (x - x_0)(a_{l+1,l} + \cdots + (x - x_{N-l-2})(a_{N-1,l} + (x - x_{N-l-1})a_{N,l}) \cdots) \end{aligned} \quad (20)$$

and set

$$\begin{aligned} a_{N,l+1} &= a_{N,l}, \\ a_{k,l+1} &= a_{k,l} + (z - x_{k-l})a_{k+1,l+1}, \quad k = N - 1, \dots, l. \end{aligned} \quad (21)$$

It follows from (20) and (21) that $P_{N,l}(z) = a_{l,l+1}$ ($0 \leq l \leq N$), and to evaluate the polynomial $P_{N,l}$ at $x = z$, one needs to perform $N - l$ multiplications and $N - l$ additions only.

Lemma 3. Let $P_{N,N+1} \equiv 0$. The following equalities are valid

$$P_{N,l}(x) = P_{N,l}(z) + (x - z)P_{N,l+1}(x), \quad l = 0, \dots, N. \quad (22)$$

Proof: If $k = N - l = 0$ then equality (22) is evident as in this case we have $P_{N,N}(x) = P_{N,N}(z) = a_{N,N}$ by (20). Now suppose the equality (22) is satisfied for all $k = 0, \dots, N - l'$ ($l < l' \leq N$). We show that it is also fulfilled for $k = N - l$. Using formulae (21) one obtains

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

This proves the lemma. □

By repeated differentiation of equality (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 $L_N \equiv P_{N,0}$, we now can rewrite the representation (16) in the form

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

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 2. Let L_N be a polynomial of the form (11), where one needs to evaluate the derivatives $L_N^{(l)}$, $0 \leq l \leq N$ 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

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

then

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

Let us note that if $x_i = 0$ for all i , then the polynomial L_N in (11) takes the form $L_N(x) = c_0 + c_1x + \dots + c_Nx^N$ and the algorithm described above is reduced to the well-known synthetic division algorithm, also called the algorithm of nested

multiplication or Horner's rule, which can be found in many textbooks on numerical analysis, see e.g. [3,5,9,10,11].

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 (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 [13]:

```

 $a[n + 1] := f[n + 1]$ ;
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 $L_N^{(l)}$, $0 \leq l \leq N$, 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;
     $v_{newn} := k * d[l + 1]$ ;
  end

```

Example 2. Four values of the function $f(x) = 1/(1 + x^2)$ are given in table 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. 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, using formulae (12), we first form the table of divided differences.

Table 2. Divided differences values.

x_i	f_i	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$f[x_i, \dots, x_{i+3}]$
-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 (11), the cubic Newton interpolating polynomial L_3 takes the form

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

By setting $a_{i,0} = c_i$, $i = 0, \dots, 3$, 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,

$$\begin{aligned} f(1.5) - L_3(1.5) &= 0.30769 - 0.25 = 0.05769, \\ f'(1.5) - L'_3(1.5) &= -0.28402 + 0.35 = 0.06598. \end{aligned}$$

6. 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 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*

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

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 $P_N(x_i) = f(x_i)$, $i = 0, \dots, N$, will the quantity $\max_{a \leq x \leq b} |f(x) - P_N(x)|$ tend to zero as $N \rightarrow \infty$?

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 $x_i = -1 + 2i/N$, $i = 0, \dots, N$. Then one can show (see [7]) that

$$\lim_{N \rightarrow \infty} \max_{0.726 \dots \leq |x| < 1} |f(x) - L_N(x)| = \infty.$$

Figure 2 illustrates the divergence of the interpolation process for Runge's example. In Figure 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 [9]) is valid.

Theorem 4. *For any prescribed system of nodes*

$$a \leq x_0^{(N)} < x_1^{(N)} < \dots < x_N^{(N)} \leq b \quad (N \geq 0) \quad (23)$$

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 [9]).

Theorem 5. *If f is a continuous function on $[a, b]$, then there exists a system of nodes as in equation (23) such that the polynomials P_N which interpolate f at these nodes converge to f , that is,*

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

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].

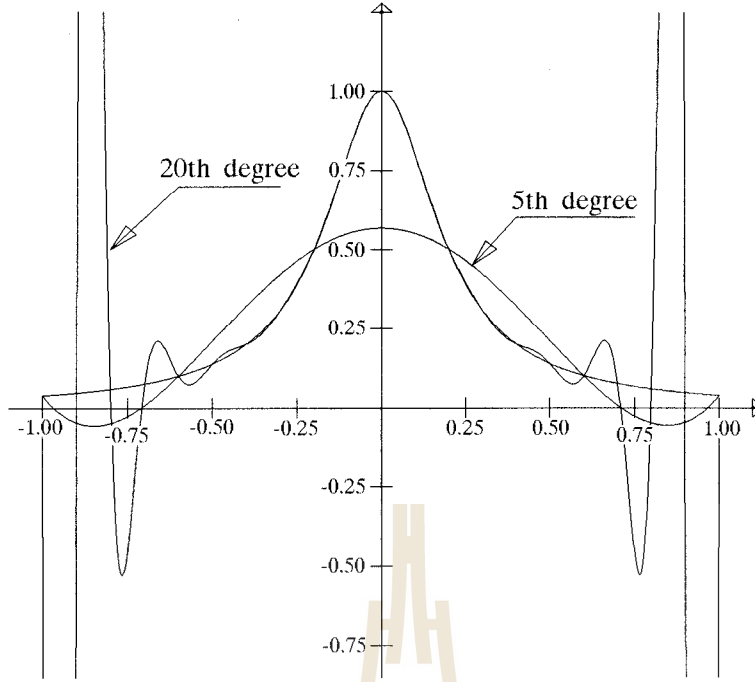


Figure 2. Runge's function interpolated by 5th-degree and 20th-degree interpolating polynomials using equidistant data.

7. 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 $[x_i, x_{i+1}]$, $i = 0, \dots, N - 1$

$$L_{i,1}(x) = f_i \frac{x_{i+1} - x}{h_i} + f_{i+1} \frac{x - x_i}{h_i}, \quad h_i = x_{i+1} - x_i. \quad (24)$$

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)

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

where $\varphi(x, y) = (x - y)_+ = \max(0, x - y)$, or according to formula (9)

$$B_{j,1}(x) = (x_{j+2} - x_j) \sum_{k=j}^{j+2} \frac{(x - x_k)_+}{\omega'_{j,2}(x_k)}, \quad j = -1, \dots, N - 1,$$

$$\omega_{j,2}(x) = (x - x_j)(x - x_{j+1})(x - x_{j+2}).$$

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

$$B_{j,1}(x) = \begin{cases} \frac{x - x_j}{h_j}, & x_j \leq x < x_{j+1}, \\ \frac{x_{j+2} - x}{h_{j+1}}, & x_{j+1} \leq x < x_{j+2}, \\ 0, & \text{otherwise.} \end{cases} \quad (25)$$

It is easy to show (e.g., see [14]) that the functions $B_{j,1}$, $j = -1, \dots, N-1$ are linearly independent on the interval $[a, b]$. Every Lagrange spline of first degree S_1^L can be uniquely represented in the form

$$S_1^L(x) = \sum_{j=-1}^{N-1} f_{j+1} B_{j,1}(x), \quad x \in [a, b]. \quad (26)$$

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 (25), the representation (26) takes the form (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 (25).

Example 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 (5), on the interval $[x_i, x_{i+1}]$, $i = 0, \dots, N-1$, we have

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

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

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

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

Example 4. Under the conditions of example 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

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

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

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

Then

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

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 $N \geq (M/\varepsilon)^{1/2}(b-a)/2$.

8. 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) , $i = -1, \dots, N + 1$. To obtain a cubic Lagrange spline, one takes the cubic Lagrange polynomial on every interval $[x_i, x_{i+1}]$, $i = 0, \dots, N - 1$,

$$L_{i,3}(x) = \sum_{j=i-1}^{i+2} f_j \frac{\omega_{i-1,3}(x)}{(x-x_j)\omega'_{i-1,3}(x_j)}, \quad \omega_{i-1,3}(x) = \prod_{j=i-1}^{i+2} (x-x_j). \quad (27)$$

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 the endpoint data (x_j, f_j) , $j = -1, N + 1$, is not given, 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 [14]).

Using formula (5) on the interval $[x_i, x_{i+1}]$ one has the estimate

$$|f(x) - L_{i,3}(x)| \leq \frac{1}{4!} |\omega_{i-1,3}(x)| \max_{x_{i-1} < x < x_{i+2}} |f^{(4)}(x)| \leq \frac{9}{384} \bar{h}^4 \|f^{(4)}\|_{C[a,b]}, \quad (28)$$

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

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

$$B_{j,3}^L(x) = \begin{cases} \frac{\omega_{k-1,3}(x)}{(x-x_{j+2})\omega'_{k-1,3}(x_{j+2})}, & \text{if } x \in [x_k, x_{k+1}], \\ 0, & \text{otherwise,} \end{cases} \quad k = j, \dots, j+3, \quad (29)$$

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

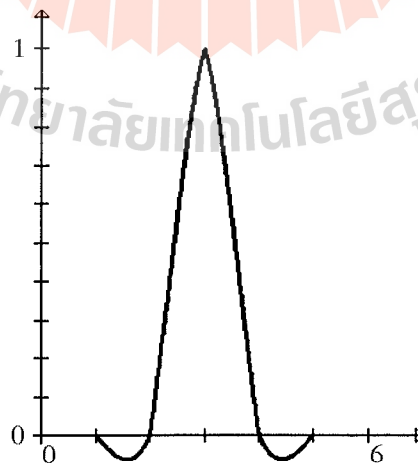


Figure 3. Cubic Lagrange B-spline with equally spaced nodes $x_i = i$, $i = 1, \dots, 5$.

It is easy to show (e.g., see [14]) that the functions $B_{j,3}^L$, $j = -3, \dots, N-1$ are linearly independent. Every cubic Lagrange spline S_3^L can be uniquely written in the form

$$S_3^L(x) = \sum_{j=-3}^{N-1} f_{j+2} B_{j,3}^L(x), \quad x \in [a, b]. \quad (30)$$

In this sum only the B-splines $B_{j,3}^L$, $j = i-3, \dots, i$ will be different from zero on the interval $[x_i, x_{i+1}]$. Using formula (29) we verify that the representation (30) coincides with formula (27) on the interval $[x_i, x_{i+1}]$

$$S_3^L(x) = \sum_{j=-3}^{N-1} f_{j+2} B_{j,3}^L(x) = \sum_{j=i-3}^i f_{j+2} B_{j,3}^L(x) = \sum_{j=i-1}^{i+2} \frac{f_j \omega_{i-1,3}(x)}{(x-x_j) \omega'_{i-1,3}(x_j)}. \quad (31)$$

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

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

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

$$x^\alpha = \sum_{j=-3}^{N-1} x_{j+2}^\alpha B_{j,3}^L(x), \quad \alpha = 0, 1, 2, 3, \quad x \in [a, b], \quad (32)$$

or in equivalent form

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

As in (31) we have on the interval $[x_i, x_{i+1}]$, $i = 0, \dots, N-1$, that

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

This proves the equalities (32).

Unfortunately, on a coarse mesh the graph of a cubic Lagrange spline can have corners as the derivatives of consecutive polynomials are not adjusted 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.

9. Local Approximation by Cubic Lagrange Splines

We now discuss a “corrected” cubic Lagrange polynomial on the interval $[x_i, x_{i+1}]$, $i = 0, \dots, N - 1$,

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

We will assume that

$$S_{i-1,3}^{(r)}(x_i - 0) = S_{i,3}^{(r)}(x_i + 0), \quad r = 0, 1, 2, \quad i = 1, \dots, N - 1. \quad (33)$$

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

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

Subtracting these polynomials we have

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

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

Hence, using the conditions (33) we obtain the system of equations

$$\begin{aligned} h_{i-1}^3 C_{i-1,1} - h_i^3 C_{i,2} &= 0, \\ 3h_{i-1}^2 C_{i-1,1} + 3h_i^2 C_{i,2} &= -h_{i-1}h_i\theta_{i,4}, \\ 3h_{i-1}C_{i-1,1} - 3h_iC_{i,2} &= (h_{i-1} - h_i)\theta_{i,4}. \end{aligned} \quad (34)$$

The equations in the overdetermined system (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

$$S_{i,3}(x) = L_{i,3}(x) - \frac{h_{i+1}^2\theta_{i+1,4}}{3h_i(h_i + h_{i+1})}(x - x_i)^3 - \frac{h_{i-1}^2\theta_{i,4}}{3h_i(h_{i-1} + h_i)}(x_{i+1} - x)^3 \quad (35)$$

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 (14) we can rewrite formula (35) in the form

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

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

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

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

Using equality (15) we can rewrite this estimate in the form

$$|f(x) - S_{i,3}(x)| \leq \left[t^2(1-t)^2 + \frac{2}{3} \right] \frac{\bar{h}_i^4}{24} M \leq \frac{35}{1152} \bar{h}_i^4 M, \quad (36)$$

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

Comparing now the estimates (28) and (36) we conclude that when replacing a cubic Lagrange interpolating spline by a local approximating one, practically the same accuracy of approximation can be obtained (compared with the estimate (28), the constant in the estimate (36) is increased only slightly).

Applications of local approximation methods to the problems of computer aided geometric design (CAGD for short) are described in [6,8].

10. Local Approximation by Cubic B-Splines

Let us discuss one more approach [15] for obtaining a formula of local approximation (35). Consider cubic B-splines

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

$j = -3, \dots, N - 1$. The graph of the cubic B-spline $B_{j,3}$ with equally spaced nodes $x_i = i, i = 1, \dots, 5$ is shown on Figure 4.

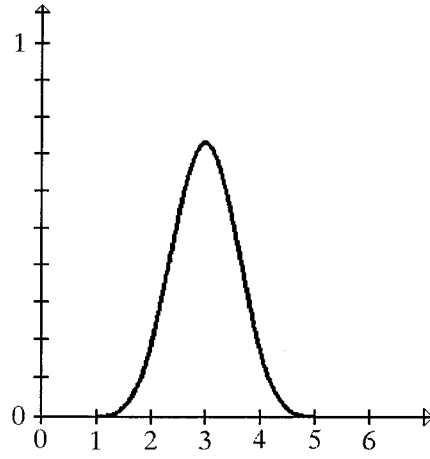


Figure 4. Cubic B-spline $B_{j,3}$ with equally spaced nodes $x_i = i, i = 1, \dots, 5$.

Using formula (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 [14]) that the functions $B_{j,3}, j = -3, \dots, N-1$ are linearly independent on $[a, b]$ and have the properties

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

$$(y - x)^3 = \sum_{j=-3}^{N-1} (y - x_{j+1})(y - x_{j+2})(y - x_{j+3})B_{j,3}(x), \quad x \in [a, b]. \quad (37)$$

Equality (37) can also be rewritten in the equivalent form

$$x^\alpha = \frac{1}{C_3^\alpha} \sum_{j=-3}^{N-1} \text{symm}_\alpha(x_{j+1}, x_{j+2}, x_{j+3})B_{j,3}(x), \quad x \in [a, b], \quad (38)$$

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

$$\begin{aligned} \text{symm}_0(x, y, z) &= 1, \\ \text{symm}_1(x, y, z) &= x + y + z, \\ \text{symm}_2(x, y, z) &= xy + xz + yz, \\ \text{symm}_3(x, y, z) &= xyz. \end{aligned}$$

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

$$S_f(x) = \sum_{j=-3}^{N-1} b_{j+2}B_{3,j}(x), \quad (39)$$

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 (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 (39) we obtain the equalities (38).

According to formula (14) we have

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

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

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

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

where ψ_j are some cubic polynomials. As $R_{i,3}(x_j) = 0$ for $j = i - 1, \dots, i + 2$ 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 (35).

11. 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 $i = -1, \dots, N$ or $i = 0, \dots, N + 1$ and considers quadratic Lagrange polynomials on the intervals $[x_i, x_{i+1}]$, $i = 0, \dots, N - 1$,

$$L_{j,2}(x) = \sum_{k=j}^{j+2} f_k \frac{\omega_{j,2}(x)}{(x - x_k)\omega'_{j,2}(x_k)} \quad (40)$$

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) , $i = 0, \dots, N$, 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) , $i = 0, \dots, N + 1$.

Using formula (5) one has on the interval $[x_i, x_{i+1}]$

$$|f(x) - L_{i,2}(x)| \leq \frac{1}{3!} |\omega_{i,2}(x)| \max_{x_i \leq x \leq x_{i+2}} |f^{(3)}(x)| \leq \frac{\sqrt{3}}{9} \bar{h}_i^3 \|f^{(3)}\|_{C[a,b]}. \quad (41)$$

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

$$B_{j,2}^L(x) = \begin{cases} \frac{\omega_{k,2}(x)}{(x - x_{j+2})\omega'_{k,2}(x_{j+2})}, & \text{if } x \in [x_k, x_{k+1}], \\ 0, & \text{otherwise,} \end{cases} \quad k = j, j+1, j+2, \quad (42)$$

$j = -2, \dots, N-1$. The graph of a quadratic Lagrange B-spline $B_{j,2}$ with equally spaced nodes $x_i = i, i = 1, \dots, 4$ is given on Figure 5.

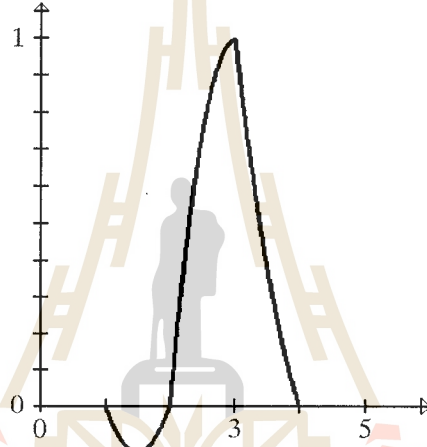


Figure 5. Quadratic Lagrange B-spline $B_{j,2}$ with nodes $x_i = i, i = 1, \dots, 4$.

It is easy to show (see [14]) that the functions $B_{j,2}^L, j = -2, \dots, N-1$ are linearly independent. Any quadratic Lagrange spline S_2^L can be uniquely written in the form

$$S_2^L(x) = \sum_{j=-2}^{N-1} f_{j+2} B_{j,2}^L(x), \quad x \in [a, b]. \quad (43)$$

In this sum only the B-splines $B_{j,2}^L, j = i-2, i-1, i$ will be different from zero on the interval $[x_i, x_{i+1}]$. Using formula (42) we verify that the representation (43) coincides with formula (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

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

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

$$x^\alpha = \sum_{j=-2}^{N-1} x_{j+2}^\alpha B_{j,2}^L(x), \quad \alpha = 0, 1, 2, \quad x \in [a, b] \quad (44)$$

or in the equivalent form

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

The equalities (44) can be verified directly by using formula (42). For $x \in [x_i, x_{i+1}]$, $i = 0, \dots, N - 1$, one has

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

12. 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 9 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

$$S_{i-1,2}^{(r)}(x_i - 0) = S_{i,2}^{(r)}(x_i + 0), \quad r = 0, 1, \quad i = 1, \dots, N - 1. \quad (45)$$

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 (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

$$S_{i,2}(x) = L_{i,2}(x) - \frac{h_{i+1}^2 \theta_{i+1,3}}{2h_i(h_i + h_{i+1})}(x - x_i)^2 - \frac{h_{i-1} \theta_{i,3}}{2(h_{i-1} + h_i)}(x_{i+1} - x)^2. \quad (46)$$

Let us estimate the error of approximation by formula (46). Using (14) one has

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

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

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

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

Using equality (15) one can rewrite this estimate in the form

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

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

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. G. E. Forsythe, M. A. Malcolm, and C. B. Moler, *Computer Methods for Mathematical Computations* (Englewood Cliffs, NJ: Prentice Hall, 1977).
8. J. Hoschek and D. Lasser, *Fundamentals of Computer Aided Geometric Design* (A K Peters, Wellesley, Massachusetts, 1993).
9. D. Kincaid and W. Cheney, *Numerical Analysis*. Second Edition (Brooks / Cole Publishing Company, Pacific Grove, California, 1996).
10. D. E. Knuth, *Seminumerical Algorithms. The Art of Computer Programming*. Vol. 2 (Addison-Wesley, Reading Mass., 1969).
11. J. H. Mathews, *Numerical Methods for Mathematics, Science and Engineering* (Prentice-Hall Int. Editions, New York, 1992).
12. H. Rutishauser, *Lectures on Numerical Mathematics* (Boston, Birkhäuser Verlag, 1990).
13. A. F. Shampine, R. C. Allen, Jr., and S. Pruess, *Fundamentals of Numerical Computing* (John Wiley & Sons, Inc., New York, 1997).
14. Yu. S. Zavyalov, B. I. Kvasov, and V. L. Miroshnichenko, *Methods of spline functions* (Nauka, Moscow, 1980, in Russian).
15. V. A. Zheludev, *Local spline approximation on arbitrary meshes*, Izv. Vysch. Uchebn. Zaved. Mat. **8** (1987) 14–18 (in Russian).

CURRICULUM VITAE

Boris I. KVASOV

ADDRESS:

Suranaree University of Technology
School of Mathematics
111, University Avenue, Muang District
Nakhon Ratchasima 30000, Thailand
Tel.: (66-44) 224283, Fax: (66-44) 224185
Email: boris@math.sut.ac.th

DEGREES:

D.Sc., Habilitation Degree in Computational Mathematics (CM for short),
Russian Academy of Sciences (RAS for short), Moscow, 1997
Senior Researcher Degree in CM, RAS, Moscow, 1986
Associate Professor Degree in CM, Ministry of Higher Education, Moscow, 1986
Ph.D. in CM, Institut of Mathematics RAS, Novosibirsk, 1973
M.Sc. in CM, Novosibirsk State University (NSU for short), Novosibirsk, 1969

POSITIONS:

1995/to date Suranaree University of Technology, Associate Professor of Mathematics
1991–1995 Institute of Computational Technology RAS, Senior Researcher, Leading Researcher
1978–1991 Institute of Theoretical and Applied Mechanics RAS, Senior Researcher
1976–1978 Institute of Mathematics Belarus Acad. of Sci., Gomel, Head of Laboratory
1974–1976 Institute of Mathematics RAS, Novosibirsk, Assistant Researcher
1975 (2 mo.) Stefan Banach Mathematical Center, Warsaw, Poland, Visiting Research Fellow
1972–1974 Krasnoyarsk State University, Senior Lecturer

ACADEMIC EXPERIENCE:

1996 (1 mo.) University degli Studi di Firenze, Italy, Visiting Professor of Mathematics
1985–1995 NSU, Associate Professor of CM
1978–1985 NSU, Assistant Professor of CM
1983 (2 mo.) Grodno State University, Visiting Assistant Professor of CM
1976–1978 Gomel State University, Senior Lecturer
1974–1976 NSU, Lecturer

Graduate/Postgraduate students: I supervised 15 M.Sc. and 2 Ph.D. in CM.

PRINCIPAL RESEARCH INTERESTS:

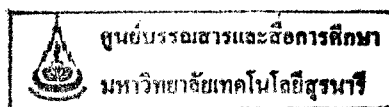
Numerical Analysis, Mathematical Methods in CAGD, Approximation Theory,
Spline Based Curve and Surface Approximation, Scientific Visualization

CURRENT RESEARCH INTERESTS:

Shape Preserving Approximation, Subdivisions, Difference Methods for Constructing Splines,
Tension and Discrete Splines and GB-splines, Curve and Surface Parametrization

RESEARCH GRANTS:

Difference Method for Constructing Shape Preserving Splines / Principal Investigator, The Thailand
Research Fund, Thailand, November 1, 1999 to October 31, 2001 (code BRG/16/2542)



Shape Preserving Parametrization for Spline Interpolation / Principal Investigator, Suranaree University of Technology, Thailand, October 1, 1999 September 30, 2000

Discrete B-Spline Approximation through Lagrange-Newton Polynomials / Principal Investigator, Suranaree University of Technology, Thailand, October 1, 1998 December 31, 1999

Algorithms of Shape Preserving Spline Approximation / Coinvestigator, The Thailand Research Fund, Thailand, July 1, 1997 to June 30, 1999 (code BRG/16/2540)

Difference Method for Construction Tension Splines / Principal investigator, MURST, Università Degli Studi di Firenze, Italy, 20 November to 20 December 1996

Geometric Splines for Curves and Surfaces Design / Principal Investigator, State Committee on Higher Education, Saint-Petersburg Technical University, January 1994 to December 1995 (code PG-13)

Shape Preserving Approximation for Curves and Surfaces / Principal Investigator, The Russian Foundation for Basic Research, RAS, Moscow, January 1993 to December 1995 (code 93-012-495)

PROFESSIONAL ACTIVITIES:

Thailand Research Fund's Royal Golden Jubilee Ph.D. Grantee, 1998

Organizer, Third All-union Conference on Approximation Theory and Problems of CM, Novosibirsk, January 28 - February 1, 1991

Guest Editor, *Approximation Theory and Problems of CM*, Modelirovanie v Mekh., **5(22)** (1991) No. 5

The USSR Government State Prize in Science and Technology Semifinalist, 1986

Referee for Zh. Vychisl. Mat. i Mat. Fiz., Moscow

Reviewer for Mathematical Reviews (in the field of Numerical Analysis, 1975-1985)

CONFERENCE PRESENTATIONS: (recent years)

On generalized discrete tension splines, Computational Techniques and Applications, Canberra, ANU, September 20-24, 1999

A General Approach to Discrete Splines, Third Annual National Symposium on Computational Science and Engineering, Chulalongkorn University, March 24-26, 1999

A General Approach to Discrete Splines, Conference on Surface Approximation and Visualization, University of Canterbury, New Zealand, February 15-18, 1999

On Tension Spline Construction by Difference Method, Int. Conference on Computational Mathematics, Chulalongkorn University, Bangkok, December 8-10, 1997

Splines in CAGD, Int. Workshop on Algebraic Analysis, Nakhon-Ratchasima, Thailand, 19th-25th January, 1997 (invited lecturer)

INVITED LECTURES:

- National University of Singapore, Singapore, 1998
- Suranaree University of Technology, Nakhon Ratchasima, Thailand, 1995, 1997
- Università degli Studi di Firenze, Italy, 1995, 1996
- Università di Milano, Italy, 1996
- Università di Siena, Italy, 1996
- Institutt for Informatikk, Universitetet i Oslo, Norway, 1991

LANGUAGE PROFICIENCY:

Fluent in English and Russian. Read, understand and can translate from German and French into English and Russian. I translated from English and French into Russian four books on splines and the finite element method: (Publ. House "Mir", Moscow)

1. Laurent, P. J., *Approximation et Optimisation*, 1975, 496 pp.
2. Descloux, J., *Methode des Elements Finis*, 1976, 96 pp.
3. Ciarlet, Ph., *The Finite Element Method for Elliptic Problems*, 1980, 512 pp.
4. Zienkiewicz, O. C. and K. Morgan, *Finite Elements and Approximation*, 1986, 320 pp.

SCIENTIFIC PUBLICATIONS: More than 70.