

A SIMPLE ALGORITHM FOR THE CONSTRUCTION OF LAGRANGE AND HERMITE INTERPOLATING POLYNOMIAL IN TWO VARIABLES

Milan A. Kovačević

Abstract. A simple algorithm for the construction of the unique Hermite interpolating polynomial (in the special case, the Lagrange interpolating polynomial) is given. The interpolation matches preassigned data of function and consecutive derivatives on a set of points laying on several radial rays. This algorithm is realized in the software package *Mathematica*.

1. Introduction

Given a function f , the interpolating problem consists of finding another function p , belonging to a prescribed finite dimensional space of functions (usually algebraic polynomials), whose values at prescribed points (interpolating nodes) coincide with those of f . This problem is referred to as the Lagrange interpolating problem. If the values of p and some of its derivatives are equal to the corresponding values of f and its derivatives at the interpolating nodes, we have the Hermite interpolating problem.

Let Π_n^2 denote the space of polynomials P of two variables of total degree n ,

$$(1.1) \quad P(x, y) = \sum_{k=0}^n \sum_{j=0}^k C_{jk} x^j y^{k-j}.$$

It is known that $\dim \Pi_n^2 = (n+1)(n+2)/2$. We consider the case when the number of interpolating conditions matches the dimension of Π_n^2 . If there is a unique solution to an interpolating problem, we say that the problem is *poised*. Unlike the polynomial interpolation in one variable, the Hermite or Lagrange interpolation in several variables is not always poised. For refinements of this result see [6], [11], [13], [2], [12], [16]. However, the problem of the choice of a particular set of points so that the interpolating problem is poised, especially if it leads to the construction

Received June 20, 2007.

2000 *Mathematics Subject Classification*. Primary 41A05; Secondary 65D05

of the interpolating polynomial, remains difficult. Some results connected with this can be found in [1], [17], [15], [5], [7], [10], [14], [2], [8], [4], [3].

In this paper we choose one of the possible situations which leads us to the poised the Lagrange and the Hermite interpolation in two variables (see, for example, [7], [5]) and allows us the simple construction of such interpolating polynomials.

In that sense, in Section 2 we consider the construction of the unique Lagrange interpolating polynomial on a set of interpolating nodes on several radial rays. That allows us to notice the strategy of constructing the unique Hermite interpolating polynomial, that is going to be used in Section 3. In Section 4 we give the realization of constructing the Hermite interpolating polynomial (in the special case, the Lagrange interpolating polynomial).

2. Lagrange Interpolation on Radial Rays

Putting $y = \ell x$ in (1.1), we have

$$(2.1) \quad p_\ell(x) = P(x, \ell x) = \sum_{k=0}^n \sum_{j=0}^k C_{jk} \ell^{k-j} x^k = \sum_{k=0}^n a_k(\ell) x^k,$$

where

$$(2.2) \quad a_k(\ell) = \sum_{j=0}^k C_{jk} \ell^{k-j} \quad (k = 0, 1, \dots, n),$$

i.e.,

$$(2.3) \quad \begin{aligned} a_0(\ell) &= C_{00}, \\ a_1(\ell) &= C_{01}\ell + C_{11}, \\ a_2(\ell) &= C_{02}\ell^2 + C_{12}\ell + C_{22}, \\ &\vdots \\ a_n(\ell) &= C_{0n}\ell^n + C_{1n}\ell^{n-1} + C_{2n}\ell^{n-2} + \dots + C_{nn}. \end{aligned}$$

On the other hand, for the given directions (radial rays)

$$\ell := \ell_0, \ell_1, \dots, \ell_n \quad (y = \ell_i x),$$

we have

$$f(x_j, \ell_i x_j) \equiv f_{j,i} \quad (j = 0, 1, \dots, n-i).$$

Remark 2.1. Notice that the x -coordinates $(x_j, j = 0, 1, \dots, n-i)$ of the interpolating nodes do not necessarily need to be the same for each direction ℓ_i ($i = 0, 1, \dots, n$) but we have them the same because we want to simplify situation.

Let us pose **the Lagrange interpolating problem**

$$(2.4) \quad P(x_j, \ell_i x_j) = f_{j,i}$$

based on distinct points $\{(x_j, \ell_i x_j) : j = 0, 1, \dots, n-i\}$ ($x_m \neq 0, m = 0, 1, \dots, n$) on the given distinct directions $\ell_i \in \mathbb{R}$ ($i = 0, 1, \dots, n$).

So, on direction ℓ_0 we have $n+1$ nodes, on ℓ_1 we have n nodes, ..., on direction ℓ_n we have only one node. All together, we have $(n+1)(n+2)/2$ interpolating conditions, which is the same as the number of the coefficients in (1.1).

According to (2.4) and (2.1), for some fixed i ($0 \leq i \leq n$), i.e., direction ℓ_i , we have

$$\sum_{k=0}^n a_k(\ell_i) x_j^k = f_{j,i} \quad (j = 0, 1, \dots, n-i),$$

i.e.,

$$\sum_{k=i}^n a_k(\ell_i) x_j^k = f_{j,i} - \sum_{k=0}^{i-1} a_k(\ell_i) x_j^k \quad (j = 0, \dots, n-i),$$

or, using matrix notation

$$(2.5) \quad \begin{bmatrix} x_0^i & x_0^{i+1} & \dots & x_0^n \\ \vdots & \vdots & & \vdots \\ x_{n-i}^i & x_{n-i}^{i+1} & \dots & x_{n-i}^n \end{bmatrix} \begin{bmatrix} a_n(\ell_i) \\ \vdots \\ a_n(\ell_i) \end{bmatrix} = \begin{bmatrix} f'_{0,i} \\ \vdots \\ f'_{n-i,i} \end{bmatrix},$$

where

$$(2.6) \quad f'_{j,i} = f_{j,i} - \sum_{k=0}^{i-1} a_k(\ell_i) x_j^k \quad (j = 0, \dots, n-i)$$

and $\sum_{k=0}^{-1} = 0$.

The system (2.5) has a unique solution for the coefficients $a_\nu(\ell_i)$ ($\nu = i, i+1, \dots, n$).

Namely, determinant of the system matrix in (2.5) is the well known modified Vandermonde determinant ($x_r \in \mathbb{R}$ ($\nu = 0, 1, \dots, i$), $i, j \in \mathbb{N}_0$)

$$(2.7) \quad \Delta_i^j(x_0, \dots, x_i) = \begin{vmatrix} x_0^j & x_0^{j+1} & \dots & x_0^{j+i} \\ x_1^j & x_1^{j+1} & \dots & x_1^{j+i} \\ \vdots & \vdots & & \vdots \\ x_i^j & x_i^{j+1} & \dots & x_i^{j+i} \end{vmatrix}$$

and

$$(2.8) \quad \Delta_i^j(x_0, \dots, x_i) = \begin{cases} x_0^j x_1^j \dots x_i^j \prod_{0 \leq k < m < i} (x_m - x_k), & i > 0, \\ x_0^j, & i = 0. \end{cases}$$

Since, in (2.5) we have $\Delta_{n-i}^i(x_0, \dots, x_{n-i}) \neq 0$, because $x_m \neq x_k$ ($m \neq k$) and $x_m \neq 0$ ($m = 0, 1, \dots, n$).

So, solving the system (2.5), we find $a_i(\ell_i), a_{i+1}(\ell_i), \dots, a_n(\ell_i)$. The obtained values $a_{i+1}(\ell_i), \dots, a_n(\ell_i)$ will be used later. The value $a_i(\ell_i)$, the previously computed values $a_i(\ell_0), \dots, a_i(\ell_{i-1})$ in the case $i > 0$, allow us to form the system of linear equations and so, according to (2.2), we have

$$\begin{aligned} C_{ii} + \dots + C_{0i}\ell_0^i &= a_i(\ell_0), \\ &\vdots \\ C_{ii} + \dots + C_{0i}\ell_i^i &= a_i(\ell_i), \end{aligned}$$

i.e.,

$$(2.9) \quad \begin{bmatrix} 1 & \dots & \ell_0^i \\ \vdots & & \vdots \\ 1 & \dots & \ell_i^i \end{bmatrix} \begin{bmatrix} C_{ii} \\ \vdots \\ C_{0i} \end{bmatrix} = \begin{bmatrix} a_i(\ell_0) \\ \vdots \\ a_i(\ell_i) \end{bmatrix}.$$

Since $\Delta_i^0(\ell_0, \dots, \ell_i) \neq 0$ for $\ell_m \neq \ell_k$ ($m \neq k$), the system (2.9) has the unique solution for the coefficients $C_{\nu,i}$ ($\nu = 0, 1, \dots, i$).

Notice that using this procedure we have computed up to now

$$\begin{aligned} &C_{00} \\ &C_{01}, C_{11}, \\ &\vdots \\ &C_{0i}, C_{1i}, \dots, C_{ii}, \end{aligned}$$

and, consequently (2.2), we have $a_k(\ell)$ ($k = 0, 1, \dots, i$). Now, it is clear why we have used the notation as that in (2.5).

We continue the algorithm and so we increase i for one, we solve the system (2.5) and the system (2.9). All of that we repeat including the case when i is equal to n . After that we have computed all coefficients of the polynomial (1.1) and we have proved its uniqueness.

3. Hermite Interpolation on Radial Rays

There is no general agreement in the multivariate case on the definition of ‘‘Hermite interpolation’’. However, it is very common to associate this name to the problems whose data are function evaluations and derivatives at the same points.

Let us consider the polynomial (1.1) for $y = \ell x$,

$$(3.1) \quad p_\ell(x) = P(x, \ell x) = \sum_{k=0}^n \sum_{j=0}^k C_{jk} \ell^{k-j} x^k = \sum_{k=0}^n a_k(\ell) x^k,$$

where $a_k(\ell)$ ($k = 0, 1, \dots, n$) are defined by (2.2) or (2.3).

In the same manner, let us observe function $(x, y) \mapsto f(x, y)$ (for $y = \ell x$),

$$(3.2) \quad u_\ell(x) = f(x, \ell x).$$

Denote by

$$(3.3) \quad \{(x_j, \ell_i x_j) : \ell_i \in \mathbb{R}, x_j \neq 0, i = 0, \dots, n, j = 0, 1, \dots, m_i\}$$

the set of interpolating nodes (distinct points), where $m_i + 1$ is the number of nodes on the direction ℓ_i ($i = 0, \dots, n$). Let

$$(3.4) \quad \{f_{j,i,k} : 0 \leq i \leq n, 0 \leq j \leq m_i, 0 \leq k \leq k_j^{(i)} - 1\}$$

be the given values, where $f_{j,i,k} \equiv u_{\ell_i}^{(k)}(x_j)$ (see (3.2)) and the integers m_i and $k_j^{(i)}$, $i = 0, 1, \dots, n, j = 0, \dots, m_i$, satisfy

$$(3.5) \quad k_0^{(i)} + \dots + k_{m_i}^{(i)} = n + 1 - i \quad (i = 0, 1, \dots, n).$$

Now, let us pose **the Hermite interpolating problem**

$$(3.6) \quad p_{\ell_i}^{(k)}(x_j) = f_{j,i,k}, \quad 0 \leq i \leq n, \quad 0 \leq j \leq m_i, \quad 0 \leq k \leq k_j^{(i)} - 1,$$

where $x \mapsto p_{\ell}(x)$ is defined by (3.1) and $\{f_{j,i,k}\}$ is defined by (3.4) and (3.5). In other words, the problem requires interpolation up to $(k_j^{(i)} - 1)$ th order derivative at node $(x_j, \ell_i x_j)$ or, we can say that we have the multiplicity $k_j^{(i)}$ of x_j on the direction ℓ_i . Obviously, when $k_j^{(i)} = 1$ for each i and j ($i = 0, 1, \dots, n, j = 0, 1, \dots, n - i$) the interpolating problem reduces to the Lagrange interpolating problem.

We note that the number of interpolating conditions (3.6) equals $\dim \Pi_n^2$, as

$$\sum_{i=0}^n (k_0^{(i)} + \dots + k_{m_i}^{(i)}) = \sum_{i=0}^n (n + 1 - i) = \frac{(n + 1)(n + 2)}{2}.$$

The algorithm for the computation of the interpolating polynomial coefficients in (3.1) for the Hermite interpolating problem (3.6) we give along.

As first we put $i = 0$, i.e., $\ell := \ell_0$, in (3.6). Then we have $k_0^{(0)} + \dots + k_{m_0}^{(0)} = n + 1$ conditions of Hermite type for the polynomial $x \mapsto p_{\ell_0}(x)$. It is known that the Hermite or Lagrange interpolating problem in one variable (the polynomial case) has the unique solution. So, we can compute $a_k(\ell_0)$, $k = 0, 1, \dots, n$. According to (2.3) we find C_{00} ($a_0(\ell_0) = C_{00}$) and memorize $a_1(\ell_0), \dots, a_n(\ell_0)$.

Now we put $i = 1$, ($\ell := \ell_1$) in (3.6). Then we have $k_0^{(1)} + \dots + k_{m_1}^{(1)} = n$ conditions for the polynomial $x \mapsto p_{\ell_1}(x)$ and another one which is dictated by the previously found value of C_{00} , i.e., $a_0(\ell_1) = C_{00}$. So, we can say, again, that we have $n + 1$ conditions of Hermite type for the polynomial $x \mapsto p_{\ell_1}(x)$ if the previously found coefficient $a_0(\ell_1)$ of the polynomial $x \mapsto p_{\ell_1}(x)$ is simulated by the condition $p_{\ell_1}(0) = C_{00}$. (For this reason we have imputed conditions $x_j \neq 0, j = 0, 1, \dots$, in (3.3).) Solving the constituted Hermite or Lagrange interpolating problem in one variable we can compute $a_k(\ell_1)$, $k = 1, \dots, n$. According to (2.3), we obtain the system

$$\begin{aligned} C_{11} + C_{01}\ell_0 &= a_1(\ell_0), \\ C_{11} + C_{01}\ell_1 &= a_1(\ell_1), \end{aligned}$$

i.e., (2.9) for $i = 1$, which has the unique solution C_{01}, C_{11} (due to $\Delta_1^0(\ell_0, \ell_1) \neq 0$ (see (2.7) and (2.8))). We memorize the values $a_2(\ell_1), \dots, a_n(\ell_1)$ because we will use them later.

Now we put $i = 2$, ($\ell := \ell_2$) in (3.6). Then we have $k_0^{(2)} + \dots + k_{m_2}^{(2)} = n - 1$ conditions for the polynomial $x \mapsto p_{\ell_2}(x)$ and another two conditions which are dictated by the previously found values of C_{00} , and C_{01}, C_{11} , i.e., the previously found values of $a_0(\ell_2) = C_{00}$ and $a_1(\ell_2) = C_{01}\ell_2 + C_{11}$. So, we can say, again, that we have $n + 1$ conditions of Hermite type for the polynomial $x \mapsto p_{\ell_2}(x)$ if the previously found coefficients $a_0(\ell_2)$ and $a_1(\ell_2)$ of the polynomial $x \mapsto p_{\ell_2}(x)$ are simulated by the conditions $p_{\ell_2}(0) = C_{00}$, $p_{\ell_2}(0)' = 1!(C_{01}\ell_2 + C_{11})$. Solving the constituted Hermite interpolating problem in one variable we can compute $a_k(\ell_2)$, $k = 2 \dots, n$. According to (2.32.3), we have the system

$$\begin{aligned} C_{22} + C_{12}\ell_0 + C_{02}\ell_0^2 &= a_2(\ell_0), \\ C_{22} + C_{12}\ell_1 + C_{02}\ell_1^2 &= a_2(\ell_1), \\ C_{22} + C_{12}\ell_2 + C_{02}\ell_2^2 &= a_2(\ell_2), \end{aligned}$$

i.e., (2.9) for $i = 2$, which has a unique solution C_{02}, C_{12}, C_{22} (due to $\Delta_2^0(\ell_0, \ell_1, \ell_2) \neq 0$ (see (2.7) and (2.8))). We memorize the values $a_3(\ell_2), \dots, a_n(\ell_2)$ because we will use them later.

Continuing the procedure for $i = 3, \dots, n$ we find all coefficients of the polynomial (1.1) and, of course, we proved its uniqueness (due to the uniqueness of the Hermite or Lagrange interpolating polynomial in one variable).

4. Implementation and Numerical Examples

Using the procedure explained in Section 3., we developed a program in the software package *Mathematica* for the construction of the Hermite or Lagrange interpolating polynomial.

The input data in presented program in APPENDIX A are:

- $f[x_-, y_-]$ – the function that have to be interpolated;
- n – the total degree of the interpolating polynomial;
- the interpolating nodes are on the radial rays $y = \ell x$, where ℓ is an element of the table ℓv , consisting of $n + 1$ elements and x is an element of the table xv , consisting of jj elements, i.e., $\ell v = \{\ell_0, \ell_1, \dots, \ell_n\}$ and $xv = \{x_0, x_1, \dots, x_{jj}\}$;
- r – the table of the multiplicities of interpolating nodes, where $r[[i, j]]$ is the multiplicity of the node $(xv[[j]], \ell v[[i]] * xv[[j]])$ and $r[[i, 1]] + r[[i, 2]] + \dots + r[[i, j]] = n + 2 - i$ ($i = 1, \dots, n + 1$).

The output data are:

- $\text{pol}[x_-, y_-]$ – interpolating polynomial.

The program calculates the necessary derivatives up to the third order maximum, i.e., $0 \leq r[[i, j]] \leq 4$. If we take $r[[i, j]] = 1$ ($i = 1, \dots, n + 1$, $j = 1, \dots, n + 2 - i$)

and the other elements of the table r equal zero, we obtain the Lagrange interpolating polynomial.

In order to illustrate the program we took a polynomial of the third total degree as a function that needs to be interpolated, i.e., $f[x, y] = 5 + 3y + 7x + 1/2y^2 + xy + 1/4x^2 + 1/3y^3 + 2xy^2 + 3x^2y + 4x^3$. Since we took that the total degree of the interpolating polynomial in the program is equal to 3, we must expect the same function for the interpolating polynomial because of its uniqueness. Taking

$$\begin{aligned} \ell v &= \{1, 2, 3, 4\}, \\ xv &= \{1/2, 1\}, \\ r &= \begin{bmatrix} 4 & 0 \\ 2 & 1 \\ 2 & 0 \\ 1 & 0 \end{bmatrix} \end{aligned}$$

for the Hermite case, and

$$\begin{aligned} \ell v &= \{1, 2, 3, 4\}, \\ xv &= \{1/2, 1, 2, 5/2\}, \\ r &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

for the Lagrange case, our expectations are realized. Namely, we obtain

$$\text{pol}[x, y] = 5 + 7x + \frac{x^2}{4} + 4x^3 + 3y + xy + 3x^2y + \frac{y^2}{2} + 2xy^2 + \frac{y^3}{3}.$$

APPENDIX A.

```
(* Hermite interpolation of the function (x,y)->fun[x,y] *)
fun[x,y]:=5 + 3*y + 7*x +1/2*y^2 + x*y +
          1/4*x^2+1/3*y^3+2*x*y^2+3*x^2*y+4*x^3
(*      n - degree of the Hermite interpolating polynomial      *)
n=3
(* The interpolating nodes are on the radial rays y=l*x, where *)
(*           l is an element of the table lv,                      *)
(*           consisting of n+1 elements, and                      *)
(*           x is an element of the table xv,                      *)
(*           consisting of jj (<=n+1) elements)                    *)
lv={1,2,3,4}
```

```

xv={1,-1,2,-2}
jj=Dimensions[xv][[1]]
r=Table[0,{k,1,n+1},{kk,1,jj}]
(* r - table of the multiplicity of interpolating nodes where *)
(*           r[[i,j]] is the multiplicity                               *)
(*           of the node (xv[[j]],lv[[i]]*xv[[j]])                       *)
(* and r[[i,1]]+r[[i,2]]+...+r[[i,jj]]=n+2-i (i=1,...,n+1) *)
r[[1,1]]=4
r[[2,1]]=2
r[[2,2]]=1
r[[3,1]]=2
r[[4,1]]=1
(*                               end of input data                               *)
fun1[x_,l_]=D[fun[x,l*x],x]
fun2[x_,l_]=D[fun1[x,l],x]
fun3[x_,l_]=D[fun2[x,l],x]
ul[x_,l_,k_]=Which[k==0,fun[x,l*x],k==1,fun1[x,l],
                    k==2,fun2[x,l], k==3,fun3[x,l]]
a=Table[0,{k,1,n+1},{kk,1,n+1}]
c=a
Do[
  number=0;
  Do[
    If[r[[i+1,j]]!=0,number=number+1],
    {j,1,jj}];
  If[i==0, Goto[downi0]];
  vek=Table[0,{k,1,number+1}];
  vek1=Table[0,{k,1,i}];
  Do[
    vek1[[ip]]=(ip-1)!*Sum[c[[k,ip]]*(lv[[i+1]]^(ip-k)),{k,1,ip}],
    {ip,1,i}];
  vek[[1]]={0,vek1};
  ii=2;
  Clear[vek1];
  Goto[jump];
Label[downi0];
ii=1;

```

```

vek=Table[0,{k,1,number}];
Label[jump];
Do[
  ki=r[[i+1,j]];
  vek[[ii]]=Table[ul[xv[[j]],lv[[i+1]],k],{k,0,ki-1};
  vek[[ii]]={xv[[j]],vek[[ii]]};
  ii=ii+1,
{j,1,number}];
p[x]=InterpolatingPolynomial[vek,x];
Do[
  a[[i+1,k]]=Coefficient[p[x],x,k-1],
{k,1,n+1}];
ll=Table[0,{k,1,i+1},{kk,1,i+1}];
f=Table[0,{k,1,i+1}];
Do[
  Do[
    ll[[k,kk]]=lv[[k]]^(i-kk+1),
    {kk,1,i+1}];
  f[[k]]=a[[k,i+1]],
{k,1,i+1}];
v = LinearSolve[ll,f];
Do[
  c[[k,i+1]]=v[[k]],
{k,1,i+1}];
Clear[ll,f,v,vek],
{i,0,n}];
Print["Interpolating polynomial:  pol[x,y]="]
pol[x_,y_]=Sum[c[[j+1,k+1]]*x^j*y^(k-j),{k,0,n},{j,0,k}]

```

REFERENCES

1. O. BIERMANN: *Über näherungsweise Kubaturen*. Montash. Math. Phys. **14**, (1903), 211–225.
2. B. BOJANOV, H. HAKOPIAN and S. SAHAKIAN: *Multivariate Interpolation and Spline Functions*. Kluwer, Dordrecht, The Netherlands, 1993.
3. B. BOJANOV, and Y. XU: *On a Hermite interpolation by polynomials of two variables*. SIAM J. Numer. Anal. **39**(5) (2002), 1780–1793.

4. J. R. BUSCH: *Oscillatory interpolation in \mathbb{R}^n* . SIAM J. Numer. Anal. **22** (1985), 107–113.
5. K. C. CHUNG and T. H. YAO: *On lattices admitting unique Lagrange interpolation*. SIAM J. Numer. Anal. **14** (1977), 735–743.
6. P. J. DAVIS: *Interpolation and Approximation*. Dover, 1975.
7. M. GASCA and J. J. MAEZTU: *On Lagrange and Hermite Interpolation in \mathbb{R}^k* . Numerische Mathematik, **39** (1982), 1–14.
8. M. GASCA and T. SAUER: *Polynomial interpolation in several variables*. Adv. Comp. Math. **12** (2000), 377–410.
9. S. L. LEE and G. M. PHILLIPS: *Polynomial interpolation at points of a geometric mesh on a triangle*. Proc. Roy. Soc. Edinburgh Sect. A **108** (1988), 75–87.
10. S. L. LEE and G. M. PHILLIPS: *Construction of lattices for Lagrange interpolation in projective space*. Constr. Approx. **7** (1991), 283–297.
11. G. G. LORENTZ: *Approximation of Functions*. Chelsea, 1966.
12. R. A. LORENTZ: *Multivariate Birkhoff Interpolation*. Lecture Notes in Math. 1516, Springer, New York, 1992.
13. J. C. MAIRHUBER: *On Haar's Theorem concerning Chebyshev approximation problems having unique solutions*. Proc. Amer. Math. Soc. **7** (1956), 609–615.
14. G. MÜHLBACH and M. GASCA: *Multivariate interpolation under projectivities I. Lagrange and Newton interpolation formulas*. Numer. Algorithms. **1** (1991), 375–400.
15. R. A. NICOLAIDES: *On a class of finite elements generated by Lagrange interpolation*. SIAM J. Numer. Anal. **9** (1972), 435–445.
16. T. SAUER and Y. XU: *On multivariate Hermite interpolation*. Adv. Comp. Math. **4** (1995), 207–259.
17. I. F. STEFFENSEN: *Interpolation*. Chelsea, New York, 1927.

Milan A. Kovačević
 Faculty of Electronic Engineering
 Department of Mathematics
 P.O. Box 73
 18000 Niš, Serbia
 kovac@elfak.ni.ac.yu