Jiří Fiala

Interpolation with prescribed values of derivatives instead of function values

Aplikace matematiky, Vol. 16 (1971), No. 6, 421–430

Persistent URL: http://dml.cz/dmlcz/103377

INTERPOLATION WITH PRESCRIBED VALUES OF DERIVATIVES INSTEAD OF FUNCTION VALUES

JIŘÍ FIALA

(Received October 2, 1970)

1. INTRODUCTION

The problem of interpolation which we are going to treat can be stated as follows: There are given points

$$a_0, a_1, \ldots, a_n, \quad a_i \neq a_j \quad \text{for} \quad i \neq j,$$

and values (derivatives) at these points

$$f_{i,0}, f_{i,1}, \ldots, f_{i,n} \quad \text{where} \quad i = 0 \quad \text{or} \quad 1.$$

Subscript $i = 0$ or 1 in $f_{i,j}$ means that at the point a_j the value or the derivative respectively is prescribed. At a given point either the function value or the value of the derivative is given but not both of them simultaneously.

Our aim is to find an interpolation polynomial

$$p(x) = b_0 + b_1 x + b_2 x^2 + \ldots + b_n x^n \tag{1}$$

satisfying the conditions

$$p(a_j) = f_{i,j} \quad \text{for} \quad i = 0,$$
$$p'(a_j) = f_{i,j} \quad \text{for} \quad i = 1.$$

Actually it is sufficient to find the unknown function values and then to use normal interpolation methods.

This problem of interpolation arises in many practical fields. As an example we can mention the process of vehicle braking. The vehicle motion is registered by a set of time measuring instruments. For all registered times the corresponding distances passed by the vehicle (i.e. the function values) are known. We know also the time at which the vehicle stops, but the full distance passed by the vehicle is unknown and is to be calculated. At the last time moment we know the value of the derivative, which is of course zero.

2. EXISTENCE AND UNIQUENESS OF THE SOLUTION

The coefficients of the interpolation polynomial can be obtained from the following system of linear equations

$$
\begin{aligned}
(2) \qquad & b_0 + b_1 a_j + b_2 a_j^2 + \ldots + b_n a_j^n = f_{i,j} \quad \text{for} \quad i = 0\,, \\
& b_1 + 2b_2 a_j + \ldots + n b_n a_j^{n-1} = f_{i,j} \quad \text{for} \quad i = 1\,.
\end{aligned}
$$

Let us denote by $a_{i_1}, a_{i_2}, \ldots, a_{i_m}$ those points at which the values of derivatives are given. In the following we shall frequently use f'_{i_k} for the derivative given at the point a_{i_k}. We suppose that $1 \leqq m < n$ (at least one derivative but not only derivatives), $a_0 < a_1 < \ldots < a_n$ and of course $a_{i_1} < a_{i_2} < \ldots < a_{i_m}$.

The determinant of the system is, as can be easily seen, equal to

$$\frac{\partial^m V(a_0, a_1, \ldots, a_n)}{\partial a_{i_1}\, \partial a_{i_2} \ldots \partial a_{i_m}}\,,$$

where $V(a_0, a_1, \ldots, a_n)$ denotes the Vandermonde determinant:

$$V(a_0, a_1, \ldots, a_n) = \prod_{i>j} (a_i - a_j)\,.$$

The determinant of the system of equations for the coefficients b_j can be equal to zero and the problem of interpolation either has no solution or has infinitely many solutions. This fact is simply illustrated by the case $a_0 = -1$, $a_1 = 0$, $a_2 = 1$ if the derivative is given at the point a_1:

$$\begin{vmatrix} 1 & a_0 & a_0^2 \\ 0 & 1 & 2a_1 \\ 1 & a_2 & a_2^2 \end{vmatrix} = \begin{vmatrix} 1 & -1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{vmatrix} = 0\,.$$

If $f_{0,0} = 0$, $f_{1,1} = 0$, $f_{0,2} = 0$, then the system has infinitely many solutions, if $f_{0,0} = 0$, $f_{1,1} \neq 0$, $f_{0,2} = 0$, then the system has no solution. (All parabolas passing through the points $+1$ and -1 have zero derivative at the point zero.)

It is not easy to recognize the cases when the determinant is non zero. In the following we shall see that the suggested method reduces this problem to the decision whether a system of m equations can be solved.

Theorem 1. *The problem of interpolation has exactly one solution if and only if*

$$(3) \qquad \frac{\partial^m V(a_0, \ldots, a_m)}{\partial a_{i_1}\, \partial a_{i_2} \ldots \partial a_{i_m}} \neq 0\,.$$

This condition is satisfied if the derivatives are given at the last m mesh points only.

It is sufficient to prove only the last conclusion of the theorem. If the derivative is given only at the point a_n, then the determinant is equal to

$$\frac{\partial V(a_0, \ldots, a_n)}{\partial a_n} = V(a_0, \ldots, a_n) \sum_{i \neq n} \frac{1}{a_n - a_i}$$

and so it is not equal to zero. Similarly for the derivatives given at the last two points: we have

$$\frac{\partial^2 V(a_0, \ldots, a_n)}{\partial a_{n-1}\, \partial a_n} = \sum_{\substack{n-1>i \\ n-1>j}} \frac{(1 + \delta_{ij})}{(a_n - a_i)\,(a_{n-1} - a_j)} V(a_0, \ldots, a_n)\,,$$

where $\delta_{ij} = 0$ for $i \neq j$, $\delta_{ii} = 1$ and hence the determinant is positive.

Naturally it is possible to get a general formula for the derivatives of the Vandermonde determinant, but this way is rather difficult because of the complicated form of the higher derivatives. Fortunately, we can prove that the determinant is positive without this general formula. We shall prove that the determinant

$$D_m = \begin{vmatrix} 1 & a_0 & a_0^2 & \ldots & a_0^n \\ 1 & a_1 & a_1^2 & \ldots & a_1^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & a_m & a_m^2 & \ldots & a_m^n \\ 0 & 1 & 2a_{m+1} & & na_{m+1}^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 1 & 2a_n & \ldots & na_n^{n-1} \end{vmatrix}$$

is positive. First of all let $m = 0$. Then the determinant can be expanded with respect to the first column and the resulting determinant is equal to

$$\begin{vmatrix} 1 & 2a_1 & \ldots & na_1^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & 2a_n & \ldots & na_n^{n-1} \end{vmatrix} = n!\, V(a_1, \ldots, a_n)$$

Hence $D_0 > 0$. Let us suppose that $D_{m-1} > 0$; we shall prove that $D_m > 0$. We have

$$\frac{\partial D_m}{\partial a_m} = D_{m-1} > 0$$

and hence D_m is an increasing function of a_m when we fix the other mesh points. If $a_m = a_{m-1}$ in D_m, it is $D_m = 0$ and then $D_m > 0$ for $a_m > a_{m-1}$.

It is clear that it is no use trying to solve the system for the coefficients directly because this system is in general ill-conditioned. In fact we are not interested in the coefficients, because we need to know only the values of the interpolation polynomial at the prescribed points. It is even sufficient to know the values at those mesh points in which the derivatives are given, and then to use usual methods of interpolation.

Let us suppose that we dispose of an interpolation method which enables us to calculate $p(x)$ and $p'(x)$ for a given value x, where $p(x)$ is the usual interpolation polynomial for the problem with all values at the mesh points known. It is even sufficient that this method work only at the mesh points. In the following we shall use the Neville-Aitken method and its modification for calculation of the derivatives.

Lagrange form of the interpolation polynomial is

$$p(x) = \Sigma' f_{0,j}\, \varphi_j(x) + \Sigma'' y_j\, \varphi_j(x)$$

where Σ' means that the sum is taken over those mesh points at which the value is given, and Σ'' has the analogous meaning for the derivatives. $y_1, y_2, \ldots, y_n$ are the values unknown at the moment at the points $a_{i_1}, a_{i_2}, \ldots, a_{i_m}$. Further

$$\varphi_i(x) = \frac{(x - a_0) \ldots (x - a_{i-1})(x - a_{i+1}) \ldots (x - a_n)}{p_i}$$

where we use the notation

$$p_i = (a_i - a_0) \ldots (a_i - a_{i-1})(a_i - a_{i+1}) \ldots (a_i - a_n)\,.$$

Differentiation yields

$$p'(x) = \Sigma' f_{0,j}\, \varphi_j'(x) + \Sigma'' y_j\, \varphi_j'(x)\,.$$

At the mesh point a_{i_k} we have

$$f'_{i_k} = p'(a_{i_k}) = \Sigma' f_{0,j}\, \varphi_j'(a_{i_k}) + \Sigma'' y_j\, \varphi_j'(a_{i_k})\,.$$

Let us denote

$$d = \begin{pmatrix} f'_{i_1} \\ \vdots \\ f'_{i_m} \end{pmatrix}, \quad y = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}, \quad r = \begin{pmatrix} r_1 \\ \vdots \\ r_m \end{pmatrix},$$

where

$$r_k = \Sigma' f_{i,j}\, \varphi_j'(a_{i_k})$$

$$R = \begin{pmatrix} \varphi'_{i_1}(a_{i_1}) & \varphi'_{i_2}(a_{i_1}) & \cdots & \varphi'_{i_m}(a_{i_1}) \\ \varphi'_{i_1}(a_{i_2}) & \varphi'_{i_2}(a_{i_2}) & \cdots & \varphi'_{i_m}(a_{i_2}) \\ \vdots & \vdots & & \vdots \\ \varphi'_{i_1}(a_{i_m}) & \varphi'_{i_2}(a_{i_m}) & \cdots & \varphi'_{i_m}(a_{i_m}) \end{pmatrix}$$

The vector d is given. If we solve the system

$$d = Ry + r$$

we obtain the missing values of y. The possibility to solve the system depends on det (R). According to the previous discussion we can state that det $(R) \neq 0$ is a necessary and sufficient condition for the existence of the solution of the interpolation problem.

We can get the explicit form of the matrix. It can be easily verified that

$$\varphi_i'(a_j) = \frac{p_j}{p_i} \frac{1}{a_j - a_i} \quad \text{for} \quad i \neq j \,,$$

$$\varphi_i'(a_i) = \sum_{i \neq j} \frac{1}{a_i - a_j} \,. \tag{4}$$

The determinant of the matrix can be simplified and summarizing the preceding reasoning we have

Theorem 2. *The interpolation problem has exactly one solution if and only if*

$$\begin{vmatrix} \sum_{i \neq i_1} \frac{1}{a_{i_1} - a_i} & \frac{1}{a_{i_1} - a_{i_2}} & \cdots & \frac{1}{a_{i_1} - a_{i_m}} \\ \vdots & \vdots & & \vdots \\ \frac{1}{a_{i_m} - a_{i_1}} & \frac{1}{a_{i_m} - a_{i_2}} & \cdots & \sum_{i \neq i_m} \frac{1}{a_{i_m} - a_i} \end{vmatrix} \neq 0 \,. \tag{5}$$

4. THE ALGORITHM

The explicit form of R (as well as the explicit form of r which can be obtained similarly) cannot be used for direct computations because of the difficulty in calculating the elements of R with the sufficient accuracy. But we can use an other way. Taking $y = 0$ (so we have all the values) we calculate the value of the vector d_0 of the derivatives at the points $a_{i_1}, a_{i_2}, \ldots, a_{i_m}$ (using a suitable interpolation method). Then of course we have $r = d_0$. If we denote $q = d - d_0 = d - r$ we can rewrite our problem in this way: find y so that $q = Ry$. If we now choose arbitrarily linearly independent vectors $y_1, y_2, \ldots, y_m$ we can calculate the vectors d_i of the derivatives at the points $a_{i_1}, \ldots, a_{i_m}$ provided that the values at those points form exactly the vector y_i. We have

$$d_i = Ry_i + r \,.$$

If we denote $q_i = d_i - r$ we have $q_i = Ry_i$. If y_i are unit vectors, then q_i will be naturally equal to the columns of R, but – and this is essential – calculated by another method. If $\det(R) \neq 0$ and y_i are linearly independent, then q_i are also independent. Hence there are numbers $\alpha_1, \alpha_2, \ldots, \alpha_m$ so that

$$q = \Sigma \alpha_i q_i .$$

The solution of the system $d = Ry + r$ is then equal to

$$y = \Sigma \alpha_i y_i$$

Summarizing we have the following algorithm:

Theorem 3. *The unknown values can be obtained by this algorithm:*

1. *Calculate r: Take $y = 0$ and calculate by interpolation the derivatives at the points $a_{i_1}, a_{i_2}, \ldots, a_{i_m}$. These derivatives then form the vector r.*
2. *Choose arbitrarily linearly independent vectors $y_1, y_2, \ldots, y_m$ and for each of them calculate by interpolation vector d_i, the components of which are the derivatives at the mesh points provided that the values at those points are the components of y_i.*
3. *Calculate $q_i = d_i - r$ for $i = 1, 2, \ldots, m$ and $q = d - r$.*
4. *Solve the system of equations for $\alpha_1, \alpha_2, \ldots, \alpha_m$:*

$$q = \Sigma \alpha_i q_i$$

This system has exactly one (more than one, no) solution if and only if the interpolation problem hase one (more than one, no) solution.

5. *Calculate the vector of the values: $y = \Sigma \alpha_i y_i$.*

Example.

$$a_0 = 0 \qquad a_1 = 1 \qquad a_2 = 2$$
$$f_0 = 1 \qquad f_1' = 2 \qquad f_2' = 4 .$$

1. The solution of the interpolation problem with $f_1 = f_2 = 0$ is the polynomial $\frac{1}{2}(x-1)(x-2)$ and its derivatives at the points 1 and 2 are $-\frac{1}{2}$ and $+\frac{1}{2}$.
2.

$$y_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad y_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

For $f_1 = 0, f_2 = 1$ the interpolation polynomial is $(x-1)^2$, for $f_1 = 1, f_2 = 0$ this polynomial is $-\frac{1}{2}(x-2)(x+1)$. Hence

$$d_1 = \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \quad d_2 = \begin{pmatrix} -\frac{1}{2} \\ -\frac{3}{2} \end{pmatrix}$$

3.
$$q = \begin{pmatrix} \frac{5}{2} \\ \frac{7}{2} \end{pmatrix}, \quad q_1 = \begin{pmatrix} \frac{1}{2} \\ \frac{3}{2} \end{pmatrix}, \quad q_2 = \begin{pmatrix} 0 \\ -2 \end{pmatrix}$$

4.
$$\begin{pmatrix} \frac{5}{2} \\ \frac{7}{2} \end{pmatrix} = \alpha_1 \begin{pmatrix} \frac{1}{2} \\ \frac{3}{2} \end{pmatrix} + \alpha_2 \begin{pmatrix} 0 \\ -2 \end{pmatrix}, \quad \alpha_1 = 5, \quad \alpha_2 = 2$$

5.
$$y = \alpha_1 y_1 + \alpha_2 y_2 = \begin{pmatrix} 2 \\ 5 \end{pmatrix}$$

(The final interpolation polynomial is $x^2 + 1$.)

5. METHOD OF INTERPOLATION FOR DERIVATIVES

We turn now to the method of interpolation and of the calculation of the derivatives by interpolation. For the interpolation on a computer the Neville-Aitken interpolation seems to be the most suitable. Starting from the points $a_0, a_1, \ldots, a_n$ and the values $f_0, f_1, \ldots, f_n$ at the mesh points, we calculate the values

$$\begin{array}{llllll} f_0^0 & f_1^0 & f_2^0 & \cdots & f_{n-1}^0 & f_n^0 \\ f_0^1 & f_1^1 & f_2^1 & \cdots & f_{n-1}^1 & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \\ f_0^{n-1} & f_1^{n-1} & & & & \\ f_0^n & & & & & \end{array}$$

according to the relations

(6)
$$f_j^0 = f_j$$
$$f_j^{k+1} = f_j^k + \frac{(f_{j+1}^k - f_j^k)(x - a_j)}{a_{j+k+1} - a_j}.$$

f_0^n is then equal to $p(x)$, i.e. to the value of the interpolation polynomial at the point x. (Neville-Aitken interpolation see e.g. in [1]. [2], [3].)

It is not difficult to find similar scheme for the calculation of the derivative. Instead of f_i^j let us write g_i^j for derivatives. The relations are then

(7)
$$g_j^0 = 0$$
$$g_j^{k+1} = g_j^k + \frac{(g_{j+1}^k - g_j^k)(x - a_j) + f_{j+1}^k - f_j^k}{a_{j+k+1} - a_j}$$

and g_0^n is equal to $p'(x)$.

6. ALGOL program

Global variables: n

$a[0:n]$	mesh points
$f[0:n]$	values/derivatives
$dh[0:n]$	**true** ... derivative
	false ... value
m	number of the derivatives

```
begin integer i, j, k; array q[1 : m, 0 : m], alfa[1 : m], ff[0 : n];

    comment now it is necessary to assign following values to the array q: vector
        q[., 0] corresponds to y = 0, vector y = 0, vector q[., k] cor-
        responds to y_k. In this program we choose the unit vectors for y_k :;

    for j := 1 step 1 until m do
    for k := 0 step 1 until m do
    begin q[j, k] := if j = k then 1 else 0 end;

    comment The following part performs the steps 1 and 2 of the algorithm.
        Interpolation is done according to the values of the vector ff. The
        results are assigned to the array q, so that q[., 0] corresponds to r
        and q[., k] to d_k;

    for k := 0 step 1 until m do
    begin j := 0;
        for i := 0 step 1 until n do
        begin if dh[i] then begin j := j + 1;
                                ff[i] := q[j, k]
                                end
                        else ff[i] := f[i]
        end; j := 0;
        for i := 0 step 1 until n do
        begin if dh[i] then begin j := j + 1;
                                    DA (ff, a[i], q[j, k])
                                end
        end
    end;

    comment step 3 of the algorithm, q is q[., 0] and q_k is q[., k];
```

```
    for k := 1 step 1 until m do
    for j := 1 step 1 until m do
        q[j, k] := q[j, k] - q[j, 0];
    j := 0;
    for i := 0 step 1 until n do
    begin if dh[i] then begin j := j + 1;
                          q[j, 0] := f[i] - q[j, 0]
                        end
    end;
    comment Now it is necessary to solve the system of m linear equations with the
            matrix q[j, k], j, k = 1, 2, ..., m and with the right hand side
            q[j, 0], j = 1, 2, ..., m. The solution is supposed in the vector
            alfa[1], ..., alfa[m]. The procedure must contain also measures for
            the case of unsolvability. Under our conditions the vector alfa gives
            directly the searched values and we can replace by them the given
            derivatives:;
    for i := 0 step 1 until n do
    begin if dh[i] then begin j := j + 1;
                          f[i] := alfa[j]
                        end
    end
```

In the above program the procedure *DA* is used to get the derivative d of the interpolation polynomial at the point x. The interpolation polynomial is given by the meshes a_i and by the values f_i. ALGOL program for this procedure is as follows:

```
procedure DA(f, x, d);
    value f; array f; real x, d;
begin integer i, k; real dif; array g[0 : n];
    for i := 0 step 1 until n do g[i] := 0;
    for k := 1 step 1 until n do
    for i := 0 step 1 until n - k do
    begin dif := a[i + k] - a[i];
    g[i] := g[i] + ((g[i + 1] - g[i]) × (x - a[i]) + (f[i + 1] - f[i]))/dif;
    f[i] := f[i] + (f[i + 1] - f[i]) × (x - a[i])/dif
    end;
    d := g[0]
end;
```

References

[1] *G. N. Lance:* Numerical methods for high speed computers. London 1960.
[2] *L. Boothroyd:* Algorithms 10, 11, The Comp. J. 9 (1966) 211—212.
[3] *J. Fiala:* Neville-Aitkenova iterační interpolace. Program a zpráva 7-00-40, VLD, Praha 1964.

Souhrn

INTERPOLACE SE ZADANÝMI HODNOTAMI DERIVACÍ NAMÍSTO FUNKČNÍCH HODNOT

JIŘÍ FIALA

Vyšetřuje se následující interpolační problém: Dána je síť bodů a v části těchto bodů jsou předepsány funkční hodnoty; ve zbývajících bodech jsou naproti tomu dány *pouze* hodnoty derivací. Hledá se interpolační polynom, který vyhoví těmto podmínkám. Tento problém nemusí mít řešení, nebo jich může mít nekonečně mnoho. Jednoznačnost i existence je zaručena na příklad, když jsou derivace zadány pouze v pravých krajních bodech.

Problém se redukuje na nalezení neznámých funkčních hodnot. Pro tyto neznámé hodnoty se sestavuje systém lineárních rovnic. Matice tohoto systému i pravá strana se získává interpolační metodou, která je modifikací Neville-Aitkenovy metody, upravené pro výpočet derivací. V závěru je uveden program v ALGOLu.

Author's address: Dr. *Jiří Fiala,* Výpočetní laboratoř dopravy, Olšanská 7, Praha 3.