Jiří Fiala

An algorithm for Hermite-Birkhoff interpolation

Aplikace matematiky, Vol. 18 (1973), No. 3, 167–175

Persistent URL: `http://dml.cz/dmlcz/103467`

AN ALGORITHM FOR HERMITE-BIRKHOFF INTERPOLATION

JIŘÍ FIALA

(Received March 17, 1972)

1. Interpolation problem

Let be given numbers $a_1 < a_2 < \ldots < a_n$ and an incidence matrix $E = (\varepsilon_{ij})$, $i = 1, 2, \ldots, n$; $j = 0, 1, \ldots, v$ (each ε_{ij} is either 0 or 1). Given the values $f_i^{(j)}$ (the j-th derivative at the point a_i) for all i, j such that $\varepsilon_{ij} = 1$, find a polynomial $p(x)$ (if it exists) of degree

$$m = \sum_{i,j} \varepsilon_{ij} - 1$$

that satisfies

$$p^{(j)}(x_i) = f_i^{(j)} \quad \text{if} \quad \varepsilon_{ij} = 1 .$$

This problem is known as the Hermite-Birkhoff interpolation problem (see e.g. [1]), shortly HB-interpolation. Hermite interpolation forms a special case of HB-interpolation: the incidence matrix has then the following property:

$$(1) \qquad \varepsilon_{ij} = 1 \Rightarrow \varepsilon_{ij'} = 1 \quad \text{for} \quad 1 \leqq j' \leqq j .$$

Another special case has been investigated in [2]: $v = 1$ and $\varepsilon_{i0} + \varepsilon_{i1} = 1$ for $i = 1, 2, \ldots, n$.

2. H-transformation

Let $e = \{(i,j): 1 \leqq i \leqq n, 0 \leqq j \leqq v, \varepsilon_{ij} = 1\}$ and let nr be an integer, $1 \leqq nr \leqq m$. Let φ_k, ψ_k $(k = 0, 1)$ be the mappings

$$\varphi_k: \{1, 2, \ldots, nr\} \to \{1, 2, \ldots, n\}$$
$$\psi_k: \{1, 2, \ldots, nr\} \to \{0, 1, \ldots, v\} .$$

Let us denote

$$(2) \qquad e_k = \{(\varphi_k(i), \psi_k(i)) : \quad i = 1, 2, \ldots, nr\} .$$

We suppose that $\varphi_k \times \psi_k$ is a one-to-one mapping and, moreover,

$$(i, j) \in e_k \Rightarrow \varepsilon_{ij} = k \quad (k = 0, 1)\,.$$

We suppose also that the mapping Φ: $e_1 \to e_0$ defined by

$$\Phi(\varphi_1(i), \psi_1(i)) = (\varphi_0(i), \psi_0(i))$$

is one-to-one.

We shall call the quadruple $T = (\varphi_1, \varphi_0, \psi_1, \psi_0)$ an H-transformation if the matrix E^T

$$\varepsilon_{ij}^T = \begin{cases} \varepsilon_{ij} & (i, j) \notin e_1 \cup e_0 \\ 1 - \varepsilon_{ij} & (i, j) \in e_1 \cup e_0 \end{cases}$$

satisfies (1). The number nr is then called the degree of the H-transformation.

For each incidence matrix E there is at least one H-transformation.

Examples of H-transformations.

1.
$$E = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\varphi_1(1) = 2 \quad \psi_1(1) = 0 \quad \varphi_1(2) = 2 \quad \psi_1(2) = 4$$
$$\varphi_0(1) = 1 \quad \psi_0(1) = 1 \quad \varphi_0(2) = 3 \quad \psi_0(2) = 1\,.$$

The transformation is represented graphically by the arrows in E.
The transformed matrix:

$$E^T = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

2. Indicating the transformation only graphically:

$$E = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

Another transformation can be found in the example to the algorithm below.

With the matrix E^T an Hermite interpolation problem is associated – of course with partly unknown values. The multiplicity p_i of the point a_i can be expressed in this way:

(3) $$p_i = \max\{j + 1: \ (i, j) \in (e \cup e_0) - e_1\}\,.$$

(maximum of the empty set is 0). It can occur that $p_i = 0$ for some i – then the corresponding points are not present in the Hermite interpolation.

3. System of equations for unknown values

The Hermite interpolation problem associated with E^T, which is equivalent to the given HB-interpolation problem under the condition that we know the values $f_i^{(j)}$ for $(i, j) \in e_0$, reads as follows:

$$p^{(j)}(a_i) = f_i^{(j)} \quad \text{for} \quad i = 1, 2, \ldots, n$$
$$j = 0, 1, \ldots, p_i - 1 \quad (\text{if } p_i \neq 0)\,.$$

Supposing we know these values (hence they exist) we can write the Hermite interpolation polynomial in the form (see e.g. [7]):

$$p(x) = \sum_{i=1}^{n} \sum_{j=0}^{p_i - 1} H_{ij}(x)\, f_i^{(j)}\,,$$

where

$$H_{ij}(x) = \frac{1}{j!} \frac{\Omega(x)}{(x - a_i)^{p_i - j}} \sum_{k=0}^{p_i - j - 1} \frac{1}{k!} \left[\frac{(x - a_i)^{p_i}}{\Omega(x)}\right]^{(k)}_{x = a_i} (x - a_i)^k \tag{4}$$

and

$$\Omega(x) = (x - a_1)^{p_1} \ldots (x - a_n)^{p_n}\,.$$

Polynomials H_{ij} (of degree $\leqq m$) are fully characterized by the following properties:

$$H_{ij}^{(s)}(a_k) = 0 \quad \text{for} \quad i \neq k \quad \text{and} \quad 0 \leqq s \leqq p_k - 1\,,$$
$$H_{ij}^{(s)}(a_i) = 0 \quad \text{for} \quad 0 \leqq s \leqq p_i - 1 \quad \text{and} \quad s \neq j\,,$$
$$H_{ij}^{(j)}(a_i) = 1\,.$$

The expression for $p(x)$ can be rewritten in the form

$$p(x) = \sum_{(i,j)\in e_0} H_{ij}(x)\, f_i^{(j)} + \sum_{(i,j)\in e - (e_0 \cup e_1)} H_{ij}(x)\, f_i^{(j)}\,.$$

If we take now $f_i^{(j)}$ for $(i, j) \in e_0$ as unknown values, we have for them the following system of equations

$$f_k^{(l)} = p^{(l)}(a_k) = \sum_{(i,j)\in e_0} f_i^{(j)} H_{ij}^{(l)}(a_k) + \sum_{(i,j)\in e - (e_0 \cup e_1)} f_i^{(j)}\, H_{ij}^{(l)}(a_k)$$

for all $(k, l) \in e_1$ — so we have exactly nr equations for nr unknown values $f_i^{(j)}$.

Theorem. *A necessary and sufficient condition for the existence and uniqueness of the solution of the HB-interpolation problem is*

$$\det\left(H_{ij}^{(l)}(a_k)\right) \neq 0\,, \quad (i, j) \in e_0, (k, l) \in e_1\,,$$

where e_0 and e_1 are given by (2) and H_{ij} by (4).

4. Calculation of the system

Let y denote the vector of unknown values:

(5) $$y_i = f_{\varphi_0(i)}^{(\psi_0(i))}, \quad i = 1, 2, \ldots, nr.$$

The system of equations for y is then $d = Ry + q$, where

$$d_i = f_{\varphi_1(i)}^{(\psi_1(i))},$$

$$R_{ij} = H_{\varphi_0(j),\psi_0(j)}^{(\psi_1(i))}(a_{\varphi_1(i)})$$

and

$$q_i = \sum_{(k,l)\in e-(e_0\cup e_1)} f_k^{(l)} H_{kl}^{(\psi_1(i))}(a_{\varphi_1(i)})$$

for all $i = 1, 2, \ldots, nr$.

The direct computation of d, R, q is complicated. Nevertheless, we can use the interpolation once more:

Let y be any nr-dimensional vector. Replacing all unknown values of $f_i^{(j)}$, $(i, j) \in e_0$, by components of y in the way that (5) holds, we get the Hermite interpolation problem. Let p_y be the solution of this problem. Let us denote by $d(y)$ the vector with the components

$$d_i(y) = p_y^{(\psi_1(i))}(a_{\varphi_1(i)}).$$

It is clear that $d(0) = q$. If now $\eta^{(i)}$ $(i = 1, 2, \ldots, nr)$ are the nr-dimensional unit vectors $(\eta_j^{(i)} = \delta_{ij})$ then it holds

$$d_i(\eta^{(j)}) = R_{ij} + q_i.$$

The last formula enables us to calculate a system for the unknown values.

In the preceding method some derivatives of Hermite interpolation polynomials were needed. We can proceed in different ways: Either we can obtain the corresponding polynomial by one of the rapid methods (see e.g. [3]–[6]) and then calculate the necessary derivatives, or we can adapt these methods to obtain the derivatives directly. All rapid methods serve to calculate coefficients of the Hermite interpolation polynomial, i.e., (up to a multiple) the derivatives at zero. It is sufficient to make a shift to get the derivatives at another point. Moreover, all the mentioned methods have the property that for the calculation of the s-th derivative it is necessary to calculate all lower derivatives – but not the higher ones. This property is common to all algorithms of this type and we shall make use of it in the suggested algorithm.

In the Appendix we give an algorithm based on other principles. This algorithm is, however, not so rapid when all the derivatives are needed.

5. Algorithm

We are given: E, values $f_i^{(j)}$ for $(i, j) \in e_1$, numbers a_i, an H-transformation $T = (\varphi_1, \varphi_0, \psi_1, \psi_0)$ of degree nr.

Step 1. Calculate the numbers

$$p_i = \max\{j + 1 : (i, j) \in (e \cup e_0) - e_1\}.$$

Step 2. Calculate q: take $y = 0$ (so we have a Hermite problem). Using a suitable algorithm (see e.g. the Appendix) for calculating derivatives of the Hermite interpolation polynomial, calculate

$$q_i = p_y^{(\psi_1(i))}(a_{\varphi_1(i)}).$$

It is convenient to use the property of the algorithm: if $\varphi_1(i) = \varphi_1(j)$ and $\psi_1(i) < < \psi_1(j)$, then in calculating q_j we get q_i as an intermediate result.

Step 3. Calculate R_{ij}: From the properties of the H-transformation follows that always $\psi_1(i) > p_{\varphi_1(i)} - 1$, so R_{ij} must be calculated from the relation

$$R_{ij} = p_{\eta(j)}^{(\psi_1(i))}(a_{\varphi_1(i)}) - q_i.$$

In calculating the derivatives we use the same property of the algorithm for obtaining the derivatives as in Step 2.

Step 4. Solve the system $d = Ry + q$ with the right-hand sides $d_i = f_{\varphi_1(i)}^{(\psi_1(i))}$. The condition $\det(R) \neq 0$ is sufficient and necessary for the existence and uniqueness of the solution of the HB-interpolation problem.

Step 5. Replacing the unknown values by the solution of the system:

$$f_{\varphi_0(i)}^{(\psi_0(i))} = y_i,$$

we get a Hermite interpolation problem, which is equivalent to the original HB-problem. Now we can proceed by any suitable method for Hermite interpolation.

6. Example

The searched polynomial will be $p(x) = x^5 + 1$, $n = 3$, $a_1 = -1$, $a_2 = 0$, $a_3 = 1$.

$$E = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix},$$

H-transformation: $nr = 2$,

$$\varphi_1(1) = 2 \quad \psi_1(1) = 2 \quad \varphi_1(2) = 3 \quad \psi_1(2) = 2$$
$$\varphi_0(1) = 1 \quad \psi_0(1) = 2 \quad \varphi_0(2) = 1 \quad \psi_0(2) = 3$$

The unknown values are then $y_1 = f_1^{(2)}$, $y_2 = f_1^{(3)}$.
The prescribed values

$$f = \begin{pmatrix} 0 & 5 & 0 & 0 & -120 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 20 & 0 & 0 \end{pmatrix},$$

$p_1 = 5, p_2 = 1, p_3 = 0$.

1. $y = 0$. The Hermite problem:

-1	0 5 0 0 -120
0	1
1	*nothing*

$p_0(x) = x^5 - 10x^3 - 20x^2 - 10x + 1$.

2. $y = \eta^{(1)}$. The Hermite problem:

-1	0 5 1 0 -120
0	1
1	*nothing*

$p_{\eta^{(1)}}(x) = \frac{1}{2}(x^5 - 5x^4 - 30x^3 - 49x^2 - 23x + 2)$.

3. $y = \eta^{(2)}$. The Hermite problem:

-1	0 5 0 1 -120
0	1
1	*nothing*

$p_{\eta^{(2)}}(x) = \frac{1}{6}(5x^5 - 5x^4 - 69x^3 - 127x^2 - 62x + 6)$.

The system of equations:

$$\begin{aligned} q_1 &= p_0^{(2)}(a_2) = -40 & q_2 &= p_0^{(2)}(a_3) = -80 \\ R_{11} &= p_{\eta^{(1)}}^{(2)}(a_2) - q_1 = -9 & R_{12} &= p_{\eta^{(2)}}^{(2)}(a_2) - q_1 = -(2 + \tfrac{1}{3}) \\ R_{21} &= p_{\eta^{(1)}}^{(2)}(a_3) - q_2 = -79 & R_{22} &= p_{\eta^{(2)}}^{(2)}(a_3) - q_2 = -(24 + \tfrac{2}{3})\,. \end{aligned}$$

The system $d = Ry + q$, $d_1 = 0$, $d_2 = 20$ has the solution $y_1 = -20$, $y_2 = 60$ so that $f_1^{(2)} = -20$, $f_1^{(3)} = 60$.

7. Problem

Let T be an H-transformation of the matrix E of the degree nr. For all i, $1 \leqq 1 \leqq n$, we define

$$\Delta(i) = \max\{j + 1: (i, j) \in e_1\}.$$

Let $\sigma(l)$ be the number of operations (multiplications and divisions) of the method used for solving a system of linear equations with an $l \times l$ matrix (usually $\sigma(l) = cl^3$) and let $\pi(k)$ be the number of operations for calculating the derivatives up to the k-th order of the Hermite interpolation polynomial.

General formulation of the problem: Find the algorithm which for a given matrix E determines the H-transformation that minimizes

$$\sigma(nr) + \sum_{i=1}^{n} \pi(\Delta(i)).$$

Author does not know even the answers to the following simplifications of the problem:

a) Find the algorithm which for a given matrix E determines its H-transformation with the minimal degree.

b) Find the algorithm which determines that H-transformation of minimal degree which minimizes the sum $\sum_{i=1}^{n} \Delta(i)$.

APPENDIX

Iteration scheme for calculating the values and the derivatives of the Hermite interpolation problem

We are given numbers $a_1 < a_2 < \ldots < a_n$ and the corresponding multiplicities p_i. At the point a_i, the value $f_i^{(0)}$ and the derivatives $f_i^{(j)}$, for $1 \leqq j \leqq p_i - 1$, are given.

These conditions determine uniquely the Hermite interpolation polynomial $p(x)$. If for all i, $p_i = 1$, then we have the usual interpolation and we can use the Neville-Aitken iteration scheme:

If we need the value of p at x, we calculate for $k = 1, 2, \ldots, n$ the rows of the scheme

$$\varphi_1^k \varphi_2^k \cdots \varphi_{n+1-k}^k$$

according to the relations

$$\varphi_i^1 = f_i^{(0)},$$

$$\varphi_j^{k+1} = \varphi_j^k + \frac{\varphi_{j+1}^k - \varphi_j^k}{a_{j+k} - a_j}(x - a_j).$$

Then we have $p(x) = \varphi_1^n$. The essential property of this scheme is that φ_i^k is the value (at x) of the interpolation polynomial which is determined by the points and values:

$$\begin{matrix} a_i \, a_{i+1} & \ldots \, a_{i+k-1} \\ f_i^{(0)} f_{i+1}^{(0)} & \ldots f_{i+k-1}^{(0)} . \end{matrix}$$

A similar scheme, however, exists also for the Hermite interpolation: Let pp_i be the partial sums of the multiplicities: $pp_0 = 0$, $pp_i = p_1 + \ldots + p_i$. The numbers $\alpha_1, \alpha_2, \ldots, \alpha_m$ and values $\varphi_1^1, \varphi_2^1, \ldots, \varphi_m^1$ are defined in this way: $\alpha_j = a_i, \varphi_j^1 = f_i^{(0)}$, where i satisfies $pp_{i-1} < j \leqq pp_i$.

The rows of the scheme are then calculated according to the relations

$$\varphi_j^{k+1} = \varphi_j^k + \begin{cases} \dfrac{\varphi_{j+1}^k - \varphi_j^k}{\alpha_{j+k} - \alpha_j}(x - \alpha_j), & \text{if} \quad \alpha_{j+k} \neq \alpha_j . \\[2ex] \dfrac{f_i^{(k)}}{k!}(x - \alpha_j)^k, & \text{if} \quad \alpha_{j+k} = \alpha_j \, (i \text{ is determined by } \alpha_j = a_i) . \end{cases}$$

Then $p(x) = \varphi_1^m$ holds.

Proof. As in the proof of the Neville-Aitken scheme it is sufficient to notice that φ_i^k is the value of the (Hermite) interpolation polynomial, which is determined by the points and values uniquely obtainable from the numbers $\alpha_i, \alpha_{i+1}, \ldots, \alpha_{i+k-1}$. The Neville-Aitken procedure is interrupted only if two points are identical. Let the point a have the multiplicity r, i.e., $f^{(0)}, \ldots, f^{(r-1)}$ is given at a. The proposed scheme calculates the needed polynomial:

$$f^{(0)} + \frac{f^{(1)}}{1!}(x - a) + \frac{f^{(2)}}{2!}(x - a)^2 + \ldots + \frac{f^{(r-1)}}{(r-1)!}(x - a)^{r-1} .$$

To get $p^{(s)}(x)$ one can use this sequence of schemes: the k-th row of the s-th scheme is

$$_s\varphi_1^k \, _s\varphi_2^k \ldots {}_s\varphi_{m+1-k}^k .$$

The relations

$$_0\varphi_j^1 = \varphi_j^1$$

$$_s\varphi_j^1 = {}_s\varphi_j^2 = \ldots = {}_s\varphi_j^s = 0 \quad \text{for all corresponding } j \text{ (i.e. the first } s \text{ rows are zero)}$$

$$_s\varphi_j^{k+1} = {}_s\varphi_j^k + \begin{cases} \dfrac{({}_s\varphi_{j+1}^k - {}_s\varphi_j^k)(x - \alpha_j) + s\,({}_{s-1}\varphi_{j+1}^k - {}_{s-1}\varphi_j^k)}{\alpha_{j+k} - \alpha_j}, & \text{if} \quad \alpha_j \neq \alpha_{j+k} \\[2ex] \dfrac{f_i^{(k)}}{(k-s)!}(x - \alpha_j)^{k-s} & \text{if} \quad \alpha_j = \alpha_{j+k} \, (i \text{ is determined by } \alpha_j = a_i) . \end{cases}$$

We have $p^{(s)}(x) = {}_s\varphi_1^m$.

Proof. It is sufficient to notice that ${}_s\varphi_j^k$ are polynomials in x and to differentiate all the schemes.

When realizing this scheme it is sufficient to keep in the memory only one row from each scheme. A special case of this algorithm can be found in [2].

References

[1] *I. J. Schoenberg:* On Hermite-Birkhoff interpolation. J. Math. Anal. Appl. 16 (1966) 538—543.

[2] *J. Fiala:* Interpolation with prescribed values of derivatives instead of function values. Apl. mat. 16 (1971) 421—430.

[3] *J. N. Lyness, C. B. Moler:* Van der Monde systems and numerical differentiation. Num. Math. 8 (1966) 458—464.

[4] *G. Galimberti, V. Pereyra:* Solving confluent Vandermonde systems of Hermite type. Num. Math. 18 (1971) 44—60.

[5] *S. - Å. Gustafson:* Rapid computation of general interpolation formulas and mechanical quadrature rules. CACM 14 (1971) 797—801.

[6] *S. - Å. Gustafson:* A416 — Rapid computation of interpolation formulas. CACM 14 (1971) 806—807.

[7] *I. S. Berezin, N. P. Židkov:* Methods of Computations I (Russian). M 1959.

Souhrn

ALGORITMUS PRO HERMITE-BIRKHOFFOVU INTERPOLACI

JIŘÍ FIALA

Dána jsou čísla $a_1 < a_2 < \ldots < a_n$, incidenční matice $E = (\varepsilon_{ij})$, $i = 1, 2, \ldots, n$; $j = 0, 1, \ldots, v$ a hodnoty $f_i^{(j)}$ (j-tá derivace v bodě a_i) pro všechna i, j taková, že $\varepsilon_{ij} = 1$. Hermite-Birkhoffovou interpolační úlohou se rozumí nalezení polynomu p (existuje-li) stupně $m = \sum \varepsilon_{ij} - 1$, splňujícího podmínky $p^{(j)}(x_i) = f_i^{(j)}$, když $\varepsilon_{ij} = 1$.

V článku se předkládá algoritmus, který převádí tuto úlohu na Hermiteovu interpolaci. Algoritmus dopočítává vybrané chybějící hodnoty derivací: tyto hodnoty jsou vyjádřeny pomocí některých zadaných hodnot. Způsob výběru všech těchto hodnot se zadává pomocí transformace incidenční matice E. Chybějící hodnoty se získají řešením soustavy lineárních rovnic: matice soustavy a její pravé strany se počítají pomocí Hermiteovy interpolace. Potřebné derivace Hermiteových interpolačních polynomů lze při tom počítat algoritmem uvedeným v dodatku.

Author's address: Dr. *Jiří Fiala*, Ústředí výpočetní techniky dopravy, Olšanská 9, 130 66 Praha 3.