Interpolation by Algebraic and Trigonometric Polynomials

By A. C. R. Newbery

The problem to be examined here is that of determining an algebraic or trigonometric polynomial of minimal degree exactly satisfying given constraints. The constraints to be considered are of the following forms: (A) Ordinates are specified at given distinct abscissas, (B) Slopes are specified at some or all of the points where the ordinates are constrained, (C) pth derivatives are specified at some points for which all lower-order derivatives, including the zeroth (i.e. the ordinate), are also specified. The polynomials which we will construct to satisfy such constraints will be of the form $P_n(x) = \sum_0^n c_r x^r$ or $S_n(x) = \sum_1^n b_r \sin rx$.

First we consider the standard Lagrangian problem with constraints of type (A) to be satisfied by an algebraic polynomial. Let $\prod_k (x) = \prod_0^k (x - x_i)$, and let $L_k(x)$ be a kth degree polynomial with specified ordinates f_i at the $k + 1$ distinct abscissas x_i, $i = 0, 1, \cdots, k$. A recursive procedure for generating the sequence of polynomials $L_k(x)$ is defined by writing $L_0 = f_0$, $\prod_0 = x - x_0$, and then, for $k = 0, 1, \cdots,$

$$
\begin{aligned}
b_k &= [f_{k+1} - L_k(x_{k+1})]/\textstyle\prod_k(x_{k+1}), \\
L_{k+1}(x) &= L_k(x) + b_k \textstyle\prod_k (x), \qquad (1)\\
\textstyle\prod_{k+1} (x) &= (x - x_{k+1}) \textstyle\prod_k (x).
\end{aligned}
$$

This is the same as Newton's formula [1] except that for each k we obtain $L_k(x)$ explicitly rather than as a sum in the form $A_0 + A_1(x - x_0) + \cdots + A_k \prod_0^k (x - x_i)$.

The algorithm (1) can be modified to accommodate constraints of type (B). Let $L_N(x)$ be a polynomial of Nth degree satisfying all the ordinate constraints and possibly some of the derivative constraints; $L_{N+1}(x)$ is to satisfy the same constraints with the additional requirement that $L_{N+1}'(x_p) = f'(x_p)$. Then we may write

$$
\begin{aligned}
h_N &= [f_p' - L_N'(x_p)]/\textstyle\prod_N' (x_p), \\
L_{N+1}(x) &= L_N(x) + h_N \textstyle\prod_N (x), \qquad (2)\\
\textstyle\prod_{N+1} (x) &= (x - x_p) \textstyle\prod_N (x).
\end{aligned}
$$

This is a convenient representation of the Hermite interpolation algorithm as presented by Natanson [2]. Note that this version is more general than the one commonly found in numerical analysis texts, because it is not presupposed that there must be a derivative constraint at *every* abscissa. In order to accommodate

Received June 25, 1965. Revised March 21, 1966.

constraints of type (C) one only has to replace the single primes in (2) by primes of appropriate multiplicity.

We now show how to construct a trigonometric polynomial $S_n(x) = \sum_1^n b_r \sin rx$ satisfying n constraints of the type considered above. We shall assume that the interval has been normalised to $[0, \pi]$ and that the nodes are strictly within this interval. As before, we consider constraints of type (A) first. Let $S_k(x)$ satisfy k ordinate constraints, and let $Q_k(x)$ be a sine polynomial of degree $k + 1$ with zeros at the k nodes. The algorithms will then closely resemble (1) and (2) with S and Q replacing L and $\prod$. The algorithm for type (A) constraints is:

$$\begin{aligned}
S_1(x) &= (f_1 \sin x)/\sin x_1, \qquad Q_1(x) = 2 \cos x_1 \sin x - \sin 2x, \\
b_k &= [f_{k+1} - S_k(x_{k+1})]/Q_k(x_{k+1}), \qquad k = 1, \cdots, \\
S_{k+1}(x) &= S_k(x) + b_k Q_k(x), \\
Q_{k+1}(x) &= 2(\cos x_{k+1} - \cos x)Q_k(x).
\end{aligned} \tag{3}$$

In order to prove the validity of this algorithm it must be shown that $\{Q_k(x)\}$ is a sequence of sine polynomials with exactly the required zeros and no others in $(0, \pi)$. From an inspection of Q_1 and of the recursion formula it is evident that the zeros of $Q_k(x)$ are correctly located. The fact that $Q_{k+1}(x)$ is a sine polynomial of degree $k + 2$ follows inductively from the recursion formula with the help of the identity

$$2 \cos x \sin rx = \sin (r + 1)x + \sin (r - 1)x. \tag{4}$$

When we apply (4) to the last of equations (3), the coefficient q_r of $\sin rx$ in $Q_k(x)$ is transformed into the coefficient q_r' of $\sin rx$ in $Q_{k+1}(x)$ in the following manner:

$$\begin{aligned}
q_1' &= 2q_1 \cos x_{k+1} - q_2, \qquad q_{k+1}' = -q_k, \\
q_r' &= 2q_r \cos x_{k+1} - (q_{r-1} + q_{r+1}), \qquad r = 2, \cdots, k.
\end{aligned}$$

In calculating b_k in the algorithm (3), we need to evaluate the polynomials S_k, Q_k at $x = x_{k+1}$. Rather than look up $\sin rx_{k+1}$, $r = 1, \cdots, k + 1$, it may be advantageous to look up $\sin x_{k+1}$ and $\cos x_{k+1}$ and then use the identity (4) to generate the sequence of sines at the cost of one multiplication and one addition each.

Extension of the algorithm to accommodate constraints of type (B) follows the same pattern as before. Let S_N satisfy all the ordinate constraints and possibly some of the derivative constraints; Q_N can be constructed to have zeros at all the nodes, with double zeros at each node where S_N satisfies a derivative constraint. The algorithm to construct S_{N+1} to satisfy the same constraints with the additional requirement $S_{N+1}'(x_p) = f_p'$ is:

$$\begin{aligned}
h_N &= [f_p' - S_N'(x_p)]/Q_N'(x_p), \\
S_{N+1}(x) &= S_N(x) + h_N Q_N(x), \\
Q_{N+1}(x) &= 2[\cos x_p - \cos x]Q_N(x).
\end{aligned}$$

The sequence of cosines required for the evaluation of h_N may be economically generated by the recursion $\cos (r + 1)x = 2 \cos x \cos rx - \cos (r - 1)x$. Extension to constraints of type (C) again involves no more than writing the appropriate number of primes in the above expression for h_N. It should be borne in mind that the approximant vanishes at the endpoints of the interval $[0, \pi]$; consequently if the approximant does not have this property, we should modify it accordingly; this may involve subtracting a linear trend as suggested in similar circumstances by Lanczos [3, p. 236].

University of Kentucky
Lexington, Kentucky 40506

1. P. J. Davis, *Interpolation and Approximation*, Blaisdell, New York, 1963. MR **28** #393.
2. I. P. Natanson, *Konstrucktive Funktionentheorie*, Akademie-Verlag, Berlin, 1955. MR **16**, 1100.
3. C. Lanczos, *Applied Analysis*, Prentice-Hall, Englewood Cliffs, N. J., 1956. MR **18**, 823.

A Note on Best Approximation in E^n†

By J. T. Day

Let D be a closed convex set with positive volume V in Euclidean n-dimensional space. Let f be a nonnegative function of class C^2 on D (see [2]), and Q be a linear polynomial on D, i.e.

$$Q(x) = a_0 + a_1x_1 + a_2x_2 + \cdots + a_nx_n, \qquad x \in D.$$

We consider the problem of "best" one sided approximation of f by Q in the sense that among all linear functions $Q(x)$ satisfying

$$(1) \qquad Q(x) \leqq f(x), \qquad x \in D,$$

we are looking for that one which maximizes $\int_D Q\,dx$.

Theorem 1. *The problem under consideration has a unique solution given by the tangent plane through the centroid p of D, provided that the eigenvalues of the Hessian matrix $(f_{ij}(x))$, $x \in D$, are nonnegative.*

The proof is by construction. Let the centroid p of D have cartesian coordinates $(p_1, p_2, \cdots, p_n)$. Then

$$(2) \qquad \int_D Q\,dx = V \cdot Q(p_1, p_2, \cdots, p_n)$$

for all linear polynomials Q (see [3]). Since $Q(p) \leqq f(p)$, we choose $Q^*(p) = f(p)$. Choose $Q_1^*(p) = f_1(p)$, $Q_2^*(p) = f_2(p), \cdots, Q_n^*(p) = f_n(p)$. Here $f_1(x) = (\partial f/\partial x_1)(x)$, etc. The above conditions determine $Q^*(x)$.

By Taylor's theorem we have $f(x) = Q^*(x) + R(x, p)$. The remainder $R(x, p)$ is nonnegative, since the eigenvalues of the Hessian matrix are nonnegative (see [2]). Thus $f(x) \geqq Q^*(x)$. We conclude that $Q^*(x)$ is a "best" approximate.

Suppose there were another "best" approximate $T(x)$. Then $T(p)$ must equal $f(p)$. Consider a point $x = (x_1, p_2, \cdots, p_n)$ where $x_1 > p_1$. By Taylor's theorem we have

Received October 11, 1965.
† This work was sponsored by the Mathematics Research Center under Contract No: DA-11-022-ORD-2059.