JOURNAL OF APPROXIMATION THEORY 6, 50–62 (1972)

On Calculating with B-Splines

CARL DE BOOR

Division of Mathematical Sciences, Purdue University, Lafayette, Indiana, 47907

Communicated by Oved Shisha

Received September 28, 1970

DEDICATED TO PROFESSOR J. L. WALSH ON THE OCCASION OF HIS 75TH BIRTHDAY

INTRODUCTION

In computational dealings with splines, the question of representation is of primary importance. For splines of fixed order on a fixed partition, this is a question of *choice of basis*, since such splines form a linear space. Only three kinds of bases for spline spaces have actually been given serious attention; those consisting of truncated power functions, of cardinal splines, and of B-splines. Truncated power bases are known to be open to severe ill-conditioning, while cardinal splines are difficult to calculate. By contrast, bases consisting of B-splines are well-conditioned, at least for orders $\leqslant 20$. Such bases are also local in the sense that at every point only a fixed number (equal to the order) of B-splines is nonzero. B-splines are also evaluated quite easily, using their definition as a divided difference of the truncated power function. Unfortunately, such calculations are ill-conditioned, particularly for partitions of widely varying interval lengths, as is indicated by the fact that special measures have to be taken in case of coincident knots.

In this note, a different way of evaluating B-splines is discussed which is very well conditioned yet efficient, and which needs no special adjustments in case of coincident knots. It is also shown that the condition of the B-spline basis increases exponentially with the order.

1. DEFINITIONS AND BASIC PROPERTIES OF (NORMALIZED) B-SPLINES

B-splines were first introduced by Schoenberg in [5, 2]. A nice compendium of many of their algebraic properties can be found in [3]. These functions are

* This work was supported in part by NSF grant GP-07163 and by the Office of Naval Research.

also known as hump functions, patch functions or hill functions. In this section, we list a few facts about B-splines for later reference.

For simplicity, we deal with splines on a bi-infinite partition

$$\pi = \{t_i\}_{i=-\infty}^{\infty}; \qquad t_i \leqslant t_{i+1}, \qquad \text{all } i, \tag{1}$$

of the (open) subinterval $I = (\lim_{i\to-\infty} t_i, \lim_{i\to\infty} t_i)$ of the real line. Because of the localness of the B-splines, it is then a simple matter to specialize to the case of a finite partition of a finite interval (see, e.g., [3 or 1]). With k a positive integer, let

$$g_k(s; t) = (s-t)_+^{k-1} = \begin{cases} (s-t)^{k-1}, & s \geqslant t \\ 0, & s < t. \end{cases} \tag{2}$$

Then, the B-spline $M_{i,k}(t)$ is given as the k-th divided difference of $g_k(s; t)$ in s on $t_i, \ldots, t_{i+k}$ for fixed t, i.e.,

$$M_{i,k}(t) = g_k(t_i, \ldots, t_{i+k}; t), \tag{3}$$

while the *normalized B-spline* $N_{i,k}(t)$ is

$$\begin{aligned} N_{i,k}(t) &= (t_{i+k} - t_i)\, M_{i,k}(t) \\ &= g_k(t_{i+1}, \ldots, t_{i+k}; t) - g_k(t_i, \ldots, t_{i+k-1}; t). \end{aligned} \tag{4}$$

If $k > 1$ and if π is a k-extended partition [1], i.e., if at most $k-1$ consecutive t_j's coincide, then both $M_{i,k}(t)$ and $N_{i,k}(t)$, as given by (3) and (4), respectively, are well-defined continuous functions. Otherwise, (3) and (4) make, in general, sense only for $t \neq t_j$, all j, because of the jump discontinuity of

$$(\partial/\partial s)^{k-1}\, g_k(s; t)$$

at $s = t$. Whenever this situation arises, we assume the definitions (3) and (4) to be augmented by the (admittedly arbitrary) demand that $N_{i,k}(t)$ and $M_{i,k}(t)$ be *right-continuous* everywhere. For instance, we let

$$M_{i,1}(t) = \begin{cases} (t_{i+1} - t_i)^{-1}, & t_i \leqslant t < t_{i+1} \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

hence

$$N_{i,1}(t) = \begin{cases} 1, & t_i \leqslant t < t_{i+1} \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Note that these definitions imply

$$M_{i,1}(t) \equiv N_{i,1}(t) \equiv 0, \qquad \text{whenever } t_i = t_{i+1}. \tag{7}$$

Most of the known properties of B-splines can be derived from the simple identity

$$M_{i,k}(t) = \frac{t - t_i}{t_{i+k} - t_i} M_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_i} M_{i+1,k-1}(t), \tag{8}$$

which we now prove[1]. For the proof, recall Leibniz' formula

$$h(s_0, ..., s_k) = \sum_{r=0}^{k} f(s_0, ..., s_r)\, g(s_r, ..., s_k) \tag{9}$$

for the k-th divided difference of the function

$$h(s) = f(s)\, g(s)$$

in terms of the divided differences of $f(s)$ and $g(s)$. Apply (9) to the function

$$h(s) = g_k(s; t) = g_{k-1}(s; t)(s - t)$$

to get

$$g_k(t_i, ..., t_{i+k}; t) = g_{k-1}(t_i, ..., t_{i+k-1}; t) \cdot 1 + g_{k-1}(t_i, ..., t_{i+k}; t) \cdot (t_{i+k} - t),$$

since all divided differences of $(s - t)$ of order 2 or higher vanish. Hence, with (3),

$$M_{i,k}(t) = M_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_i} (M_{i+1,k-1}(t) - M_{i,k-1}(t)),$$

which is (8), slightly rewritten.

In terms of the $N_{i,k}$, (8) reads

$$N_{i,k}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} N_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(t). \tag{10}$$

The identity (8) states that, for any t, $M_{i,k}(t)$ is an average (or, a linear cross mean, as Aitken would call it) of the numbers $M_{i,k-1}(t)$ and $M_{i+1,k-1}(t)$. Further, for $t_i < t < t_{i+k}$, $M_{i,k}(t)$ is a strictly convex combination of these two numbers. Since $M_{i,1}(t)$ is positive for $t_i \leqslant t < t_{i+1}$ and zero otherwise, it therefore follows at once from (8) (by induction on k) that, for $k > 1$, $M_{i,k}(t)$ is positive for $t_i < t < t_{i+k}$ and zero otherwise. The normalized B-spline $N_{i,k}(t)$ satisfies, of course, the same condition.

It is this feature which makes the $N_{i,k}(t)$ so attractive for calculations. In order to evaluate the function

$$F(t) = \sum_i A_i N_{i,k}(t) \tag{11}$$

[1] This identity was also found by Lois Mansfield.

at a point $\hat{t} \in [t_j, t_{j+1})$, it is merely necessary to calculate the k numbers

$$N_{i,k}(\hat{t}), \qquad i = j - k + 1, \ldots, j;$$

$F(\hat{t})$ is then given by

$$F(\hat{t}) = \sum_{i=j-k+1}^{j} A_i N_{i,k}(\hat{t}).$$

Differentiation of $F(t)$ is equally simple. One has

$$\begin{aligned} N_{i,k}^{(1)}(t) &= (d/dt)[g_k(t_{i+1}, \ldots, t_{i+k}\,;t) - g_k(t_i, \ldots, t_{i+k-1}\,;t)] \\ &= -(k-1)[M_{i+1,k-1}(t) - M_{i,k-1}(t)]. \end{aligned}$$

Hence,

$$\begin{aligned} F^{(1)}(t) &= (k-1) \sum_i A_i[M_{i,k-1}(t) - M_{i+1,k-1}(t)] \\ &= (k-1) \sum_i A_i^{(1)} N_{i,k-1}(t), \end{aligned} \tag{12}$$

where

$$A_i^{(1)} = (A_i - A_{i-1})/(t_{i+k-1} - t_i). \tag{13}$$

More generally, with

$$\begin{aligned} A_i^{(0)} &= A_i, \\ A_i^{(j)} &= (A_i^{(j-1)} - A_{i-1}^{(j-1)})/(t_{i+k-j} - t_i), \qquad j > 0, \end{aligned} \tag{14}$$

one has

$$F^{(j)}(t) = (k-1) \cdots (k-j) \sum_i A_i^{(j)} N_{i,k-j}(t). \tag{15}$$

If π is uniform,

$$t_i = t_0 + ih, \qquad \text{all} \quad i,$$

then (15) reduces to

$$F^{(j)}(t) = h^{-j} \sum_i (\nabla^j A_i)\, N_{i,k-j}(t)$$

which is familiar from [5]. We return to the evaluation of the spline function $F(t)$ and its derivatives in Section 2.

Using the identity (8), it is possible to rewrite $F(t)$ in terms of normalized B-splines of lower order, with certain polynomial coefficients. One has

$$\begin{aligned} F(t) &= \sum_i A_i N_{i,k}(t) \\ &= \sum_i A_i\{(t - t_i)\, M_{i,k-1}(t) + (t_{i+k} - t)\, M_{i+1,k-1}(t)\} \\ &= \sum_i \{A_i(t - t_i) + A_{i-1}(t_{i+k-1} - t)\}\, M_{i,k-1}(t) \\ &= \sum_i A_i^{[1]}(t)\, N_{i,k-1}(t), \end{aligned}$$

where

$$A_i^{[1]}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} A_i + \frac{t_{i+k-1} - t}{t_{i+k-1} - t_i} A_{i-1}\,.$$

More generally, setting

$$A_i^{[j]}(t) = \begin{cases} A_i\,, & j = 0 \\ \dfrac{t - t_i}{t_{i+k-j} - t_i} A_i^{[j-1]}(t) + \dfrac{t_{i+k-j} - t}{t_{i+k-j} - t_i} A_{i-1}^{[j-1]}(t), & j > 0, \end{cases} \tag{16}$$

one gets

$$F(t) = \sum_i A_i^{[j]}(t)\, N_{i,k-j}(t). \tag{17}$$

Since $N_{i,1}(t) = 1$ for $t_i \leqslant t < t_{i+1}$ and is zero otherwise, it follows that

$$F(t) = A_i^{[k-1]}(t), \qquad t_i \leqslant t < t_{i+1}\,. \tag{18}$$

Hence, if $t \in [t_i\,, t_{i+1})$, then $F(t)$ can also be found, from $A_{i-k+1}\,,\ldots, A_i$, by forming repeatedly certain convex combinations according to (16).

Finally, we mention the important identity

$$(s - t)^{k-1} = \sum_i \varphi_{i,k}(s)\, N_{i,k}(t), \quad \text{with} \quad \varphi_{i,k}(s) = \prod_{r=1}^{k-1} (s - t_{i+r}), \quad \text{all } i, \tag{19}$$

which was first proved by Marsden [4], and which simplifies many dealings with splines. Its proof is straightforward : Setting

$$A_i^{[0]}(t) = A_i = \varphi_{i,k}(s), \qquad \text{all } i,$$

one gets from (16) that

$$\begin{aligned} A_i^{[1]}(t) &= \{(t - t_i)\, \varphi_{i,k}(s) + (t_{i+k-1} - t)\, \varphi_{i-1,k}(s)\}/(t_{i+k-1} - t_i) \\ &= \varphi_{i,k-1}(s)\{(t - t_i)(s - t_{i+k-1}) + (t_{i+k-1} - t)(s - t_i)\}/(t_{i+k-1} - t_i) \\ &= \varphi_{i,k-1}(s)(s - t); \end{aligned}$$

hence

$$\sum_i \varphi_{i,k}(s)\, N_{i,k}(t) = (s - t) \sum_i \varphi_{i,k-1}(s)\, N_{i,k-1}(t).$$

Since

$$\sum_i \varphi_{i,1}(s)\, N_{i,1}(t) \equiv \sum_i N_{i,1}(t) \equiv 1,$$

induction on k now proves (19).

By expanding both sides of (19) in powers of s and comparing coefficients of like powers, it follows, e.g., that

$$\sum_i N_{i,k}(t) \equiv 1, \tag{20}$$

reaffirming the conclusion from (16) and (18) that, for $t \in [t_i, t_{i+1})$, the number $F(t)$ is a *convex* combination of the numbers $A_{i-k+1}, \ldots, A_i$.

It also follows from (19) that

$$(s - t)^{k-1} = \sum_i \psi_{i,k}(t)\, N_{i,k}(s), \qquad \psi_{i,k}(t) = (t_{i+1} - t) \cdots (t_{i+k-1} - t); \tag{21}$$

hence

$$(s - t_j)_+^{k-1} = \sum_i \psi_{i,k}^+(t_j)\, N_{i,k}(s), \qquad \psi_{i,k}^+(t) = (t_{i+1} - t)_+ \cdots (t_{i+k-1} - t)_+ . \tag{22}$$

2. Evaluation of B-splines

The material in the preceding section suggests (at least) two stable, yet efficient ways to evaluate the function

$$F(t) = \sum_i A_i N_{i,k}(t)$$

at any particular t, which we now discuss. The resulting algorithms can, of course, also be used to evaluate the single B-spline $N_{i,k}(t)$, merely by specializing to the situation

$$A_j = \delta_{ij}, \qquad \text{all} \quad j.$$

The more obvious of the two algorithms is based on (16)-(18). Having found i such that $t_i \leqslant t < t_{i+1}$, one generates all the entries in the following table, using (16):

$$\begin{array}{lllll}
A_{i-k+1}^{[0]}(t) & & & & \\
A_{i-k+2}^{[0]}(t) & A_{i-k+2}^{[1]} & & & \\
\vdots & \vdots & \ddots & & \\
A_{i-1}^{[0]}(t) & A_{i-1}^{[1]}(t) & \cdots & A_{i-1}^{[k-2]}(t) & \\
A_i^{[0]}(t) & A_i^{[1]}(t) & \cdots & A_i^{[k-2]}(t) & A_i^{[k-1]}(t)
\end{array}$$

The right-most entry is then the desired number $F(t)$.

Set

$$\begin{aligned}
&A(r, s) = A_{i-k+r}^{[s-1]}(t), \qquad r = s,\dots, k; \qquad s = 1,\dots, k,\\
&\left.\begin{aligned} DP(r) &= t_{i+r} - t,\\ DM(r) &= t - t_{i-k+r}, \end{aligned}\right\} r = 1,\dots, k,
\end{aligned} \tag{23}$$

to simplify notation. Then

$$A(r, 1) = A_{i-k+r}, \qquad r = 1,\dots, k,$$

$$A(r, s+1) = (DM(r) * A(r, s) + DP(r-s) * A(r-1, s))/(DM(r) + DP(r-s))$$
$$r = s+1,\dots, k; \qquad s = 1,\dots, k-1. \tag{24}$$

Note that

$$\begin{aligned}
DM(r) + DP(r-s) &= t - t_{i-k+r} + t_{i+r-s} - t\\
&= t_{i+r-s} - t_{i+r-k} \geqslant t_{i+1} - t_i > 0,
\end{aligned}$$

so that coincident points in the partition π cause no additional difficulty if, as we assume, i is chosen so that $t_i \leqslant t < t_{i+1}$.

The calculation of the $A(r, s)$ can either be carried out column by column, i.e., with

$$r = s,\dots, k; \qquad s = 2,\dots, k$$

or row by row, i.e., with

$$s = 2,\dots, r; \qquad r = 2,\dots, k$$

or downward diagonal by downward diagonal, i.e., with

$$s = 2,\dots, j, \qquad r = s + k - j; \qquad j = 2,\dots, k.$$

Each way requires only one one-dimensional array with k entries for the actual storage of the successively calculated numbers $A(r, s)$. In the first way, one would precalculate the array DP, in the last, one would precalculate the array DM, while the second would require initial calculation of both the DP and the DM array.

If the value of $F(t)$ *and* of some of its derivatives are required at the same time, then it is probably better to use an algorithm which generates at the same time all the numbers $N_{i,j}(t)$ which are not zero for the given t. With the assumption that

$$t_i \leqslant t < t_{i+1}\,,$$

this amounts to generating all the entries of the following triangular table:

$$\begin{array}{ccccc}
N_{i,1}(t) & N_{i-1,2}(t) & \cdots & N_{i-k+2,k-1}(t) & N_{i-k+1,k}(t) \\
 & N_{i,2}(t) & \cdots & N_{i-k+3,k-1}(t) & N_{i-k+2,k}(t) \\
 & & \ddots & \vdots & \vdots \\
 & & & N_{i,k-1}(t) & N_{i-1,k}(t) \\
 & & & & N_{i,k}(t)
\end{array}$$

The $(k - j)$th column of this table contains the numbers needed for the evaluation of $F^{(j)}(t)$ using (15), $j = 0,\dots, k - 1$. For this reason, we describe here only how to generate this table column by column.

Set

$$\begin{aligned}
N(r, s) &= N_{i+r-s,s}(t), \\
\left.\begin{aligned} DP(r) &= t_{i+r} - t, \\ DM(r) &= t - t_{i+1-r}\,, \end{aligned}\right\} &\; r = 1,\dots, k,
\end{aligned} \tag{25}$$

to simplify notation. The needed table entries are then

$$N(r, s), \qquad r = 1,\dots, s; \qquad s = 1,\dots, k,$$

while

$$N(r, s) = 0, \qquad \text{for} \quad r > s \quad \text{or} \quad r < 1. \tag{26}$$

With the abbreviations (25), we get from (10) that

$$N(r, s+1) = DM(s+1-r+1)\frac{N(r-1, s)}{DP(r-1) + DM(s+1-r+1)} + DP(r)\frac{N(r, s)}{DP(r) + DM(s+1-r)}. \tag{27}$$

We emphasize that, once again, this formula is unaffected by the possible presence of coincident t_j's, since

$$DP(r) + DM(s+1-r) = t_{i+r} - t_{i+r-s} \geqslant t_{i+1} - t_i > 0$$

for all values of r and s of interest.

Equations (25) and (27) lead to the following algorithm for the generation of the $N(r, s)$:

```
Set N(1, 1) = 1;
for s = 1,..., k - 1, do:
:       set DP(s) = t_{i+s} - t, DM(s) = t - t_{i+1-s},
:       set N(1, s + 1) = 0;
:       for r = 1,..., s, do:
:         :     set M = N(r, s)/(DP(r) + DM(s + 1 - r)),
:         :     set N(r, s + 1) = N(r, s + 1) + DP(r)* M,
:.......:.....set N(r + 1, s + 1) = DM(s + 1 - r) * M.
```

This algorithm can, of course, be modified so as to use only a one-dimensional array of k entries for the storage of the $N(r, s)$, by overwriting successive columns.

3. Condition of the B-spline basis

A limited number of numerical experiments have shown both algorithms presented in the preceding section to be extremely stable when used for the evaluation of $F(t)$. This is not suprising, since both algorithms arrive at $F(t)$ by repeatedly forming convex combinations. Even for $k = 80$, the absolute error in the computed value for $F(t)$ was only about the size of roundoff in the coefficients A_i.

These numerical experiments showed, incidentally, the unpleasant but important fact that the condition number of the normalized B-spline basis increases exponentially with the order k. This can be confirmed, in the case of a uniform partition, by the following calculations.

The condition number $\varkappa(k, \pi)$ of the normalized B-spline basis $\{N_{i,k}\}_i$ for the partition $\pi = \{t_i\}$ is defined by

$$\varkappa(k, \pi) = \sup_{\|A\|_\infty=1} \left\| \sum_i A_i N_{i,k} \right\|_\infty \Big/ \inf_{\|A\|_\infty=1} \left\| \sum_i A_i N_{i,k} \right\|_\infty, \tag{28}$$

where

$$\| A \|_\infty = \sup_i | A_i |$$

and

$$\|f\|_\infty = \sup_{t \in I} |f(t)|,$$

I being the interval for which π is a partition. It is assumed, of course, that $\{N_{i,k}\}$ is linearly independent, i.e., that no t_i agrees with more than $k - 1$ other t_j's. By (20),

$$\left\| \sum_i A_i N_{i,k} \right\|_\infty \leqslant \| A \|_\infty$$

with equality when $A_i = 1$, all i. Hence

$$\varkappa(k, \pi) = 1 \Big/ \inf_{\|A\|_\infty=1} \left\| \sum_i A_i N_{i,k} \right\|_\infty . \tag{29}$$

It was proved in [1] that, for each k,

$$D_k = \sup_\pi \varkappa(k, \pi) < \infty$$

Preliminary calculations based on the argument in [1] give upper bounds for D_k which increase about as fast as $k!$, as k increases. These bounds are probably not sharp. But it can be shown that D_k must increase at least exponentially with k.

THEOREM. *If the partition π is uniform,*

$$t_j = t_0 + jh, \qquad \textit{all} \quad j,$$

then

$$\varkappa(k, \pi) = 1/\varphi_k(\pi), \tag{30}$$

where

$$\varphi_k(u) = \sum_j \psi_k(u + 2\pi j),$$

$$\psi_k(u) = \left(\frac{\sin(u/2)}{u/2}\right)^k.$$

Hence,

$$\varkappa(k, \pi) \geqslant (\pi/2)^{k-2}, \qquad \text{for} \quad k > 1,$$
$$\lim_{k\to\infty} \varkappa(k, \pi)/(\pi/2)^k = 2. \tag{31}$$

Proof. We show how to obtain (30) from Schoenberg's recent paper [6] on Cardinal Interpolation. First, we note that $\varkappa(k, \pi)$ is invariant under a linear change of the independent variable; hence we may restrict attention to the particular uniform partition

$$\pi = \{j\}.$$

If $A = (A_i)$ is any sequence, then

$$\left\| \sum_i A_i N_{i,k} \right\|_\infty \geqslant \| C_A \|_\infty , \tag{32}$$

with

$$C_A(i) = \sum_j A_j N_{j,k}(i + k/2), \qquad \text{all } i.$$

Since we are dealing with the particular partition $\pi = \{j\}$, we have

$$N_{j,k}(t) = N_{0,k}(t - j) = M_k(t - j - k/2),$$

where, in the notation of [6 or 5],

$$M_k(t) = \frac{1}{(k-1)!}\, \delta^k t_+^{k-1}.$$

Therefore,

$$C_A(i) = \sum_j A_j M_k(i - j), \qquad \text{all } i.$$

In this form, the linear sequence-to-sequence transformation

$$A \to C_A$$

has been studied in detail in [6], where it is proved (see, in particular Section 6 of [6]) that

$$\| C_A \|_\infty \geqslant \varphi_k(\pi) \| A \|_\infty , \quad \text{with equality if} \quad A_i = -A_{i+1} , \quad \text{all} \quad i. \tag{33}$$

Combining (32) and (33), we get that

$$\inf_{\|A\|_\infty = 1} \Big\| \sum_j A_j N_{j,k} \Big\|_\infty \geqslant \varphi_k(\pi). \tag{34}$$

On the other hand, if

$$A_i = -A_{i+1} , \quad \text{all} \quad i,$$

then it easily follows (e.g., from Lemma 12 of [6]) that

$$\Big\| \sum_j A_j N_{j,k} \Big\|_\infty = \| C_A \|_\infty ,$$

hence, by (33), then

$$\Big\| \sum_j A_j N_{j,k} \Big\|_\infty = \varphi_k(\pi) \| A \|_\infty .$$

Combining this with (34) and (29) gives (30); Q.E.D.

We note that the inequality (31) alone can be derived directly by calculation of $\Sigma_j(-1)^j N_{j,k}(k/2)$. Further, the theorem implies that, for a uniform partition,

$$\varkappa(k, \pi) \approx 10^{k/5}. \tag{35}$$

This implies that, on a typical 7 decimal digit machine and with $k = 40$, the calculated value of $F(t) = \Sigma_j A_j N_{j,k}(t)$ at some point $\hat{t}$ may well be inaccurate in the first nonzero digit. Since, on the other hand, the normalized *B*-spline basis is, at present, the only suitable basis for dealing with splines in computations, this seems to limit the use of splines in solving functional equations on a computer to splines of relatively low order, say of order $k < 20$, unless one is willing to pay the price of multiple-precision arithmetic.

References

1. C. de Boor, On uniform approximation by splines, *J. Approximation Theory* **1** (1968), 219–235.
2. H. B. Curry and I. J. Schoenberg, On spline distributions and their limits: the Polya distributions, *Abstr. Bull. Amer. Math. Soc.* **53** (1947), 1114.

3. H. B. CURRY AND I. J. SCHOENBERG, On Polya frequency functions IV: The fundamental spline functions and their limits, *J. Anal. Math.* **17** (1966), 71–107.
4. M. MARSDEN, An identity for spline functions and its application to variation diminishing spline approximations, *J. Approximation Theory* **3** (1970), 7–49.
5. I. J. SCHOENBERG, Contributions to the problem of approximation of equidistant data by analytic functions, *Quart. Appl. Math.* **4** (1946), 45–99; 112–141.
6. I. J. SCHOENBERG, Cardinal interpolation and spline functions. *J. Approximation Theory* **2** (1969), 167–206.

Added in proof: The following additional references are of interest.

C. DE BOOR, "Subroutine package for calculating with *B*-splines," Los Alamos Scient. Lab. Report LA-4728-MS, Aug. 1971.

M. G. COX, "The numerical evaluation of *B*-splines," National Physical Laboratory Report DNAC 4, Aug. 1971.

T. SEGETHOVA, "Numerical construction of hill functions," Univ. Maryland Computer Science Center Technical Rep. 70-110, NGL-21-002-008, April 1970.